

9.3

*IBM MQ*

**IBM**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información en [“Avisos” en la página 319](#).

Esta edición se aplica a la versión 9 release 3 de IBM® MQ y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere adecuada, sin incurrir por ello en ninguna obligación con el remitente.

© **Copyright International Business Machines Corporation 2007, 2025.**

---

# Contenido

<b>Visión general técnica.....</b>	<b>5</b>
Introducción a la colocación de mensajes en colas.....	5
Características principales y ventajas de la colocación de mensajes en colas.....	7
Terminología sobre la colocación de mensajes en colas.....	9
Mensajes y colas.....	13
Objetos de IBM MQ.....	14
Tipos de objetos.....	16
Denominación de objetos de IBM MQ.....	37
Gestión de colas distribuidas y clústeres.....	43
Componentes de la gestión de colas distribuidas.....	47
Componentes de clúster.....	57
Mensajería de publicación/suscripción.....	63
Componentes de publicación/suscripción.....	64
Ejemplo de publicación/suscripción de gestor de colas individual.....	89
Redes de publicación/suscripción distribuidas.....	90
IBM MQ Multicast.....	108
Conceptos iniciales sobre la multidifusión.....	108
MQ TelemetryVisión general de.....	109
Introducción a MQ Telemetry.....	111
Casos de uso de telemetría.....	113
Conexión de dispositivos de telemetría a un gestor de colas.....	119
Protocolos de conexión de telemetría.....	120
Servicio de telemetría (MQXR).....	120
Canales de telemetría.....	121
Protocolo IBM MQ Telemetry Transport.....	121
Clientes MQTT.....	121
Envío de un mensaje a un cliente MQTT.....	122
Envío de un mensaje a una aplicación de IBM MQ desde un cliente MQTT.....	131
Aplicaciones de publicación/suscripción MQTT.....	132
Aplicaciones de telemetría.....	133
Integración de MQ Telemetry con gestores de colas.....	133
Sesiones de MQTT con estado y sin estado.....	136
Cuando un cliente MQTT no está conectado.....	136
Acoplamiento dinámico entre clientes MQTT y aplicaciones de IBM MQ.....	137
Seguridad de MQ Telemetry.....	137
Globalización de MQ Telemetry.....	138
Rendimiento y escalabilidad de MQ Telemetry.....	138
Dispositivos soportados por MQ Telemetry.....	141
Seguridad en IBM MQ.....	141
Soporte TLS de cliente gestionado de IBM MQ.NET.....	142
IBM MQ MQI clients.....	143
¿Por qué utilizar clientes IBM MQ.....	145
¿Qué es un cliente transaccional extendido?.....	147
Cómo se conecta el cliente al servidor.....	148
Gestión y soporte de transacciones.....	149
Extensión de recursos del gestor de colas.....	151
Interfaces de lenguaje de IBM MQ Java.....	152
IBM MQ classes for JMS/Jakarta Messaging.....	153
Proveedor de mensajería de IBM MQ.....	164
Conceptos de IBM MQ for z/OS.....	164
El gestor de colas en z/OS.....	166
El iniciador de canal en z/OS.....	167

Términos y tareas para gestionar IBM MQ for z/OS.....	168
Colas compartidas y grupos de compartición de colas.....	171
Transferencia a colas entre grupos.....	219
Gestión de almacenamiento en z/OS.....	233
Registro en IBM MQ for z/OS.....	238
Definición de sistema en z/OS.....	250
Recuperación y reinicio en z/OS.....	261
Conceptos de seguridad en IBM MQ for z/OS.....	279
Disponibilidad en z/OS.....	286
Supervisión y estadísticas en IBM MQ for z/OS.....	289
Disposición de unidad de recuperación en z/OS.....	291
IBM MQ y otros productos de z/OS.....	293
IBM MQ y CICS.....	294
IBM MQ y IMS.....	295
IBM MQ y los adaptadores por lotes, TSO y RRS de z/OS.....	299
IBM MQ for z/OS y WebSphere Application Server.....	300
Managed File Transfer.....	301
¿Cómo funciona MFT con IBM MQ?.....	304
Visión general de la topología de MFT.....	304
Descripción general de MFT REST API.....	306
IBM MQ Internet Pass-Thru.....	306
usos de MQIPT.....	307
Cómo funciona MQIPT.....	309
Configuraciones posibles de MQIPT.....	310
Configuraciones compatibles.....	312
Configuraciones de canales soportadas.....	314
Terminación del canal y condiciones de anomalía.....	315
Seguridad de mensajes.....	315
Gestores de colas de varias instancias y alta disponibilidad.....	315
IBM MQ Console y REST API.....	316
<b>Avisos.....</b>	<b>319</b>
Información acerca de las interfaces de programación.....	320
Marcas registradas.....	321

# Visión general técnica de IBM MQ

---

Utilice IBM MQ para conectar las aplicaciones y gestionar la distribución de información en toda la organización.

IBM MQ permite que los programas se comuniquen entre sí a través de una red de componentes diferentes (procesadores, sistemas operativos, subsistemas y protocolos de comunicación) utilizando una interfaz de programación de aplicaciones coherente. Las aplicaciones diseñadas y escritas mediante esta interfaz se conocen como aplicaciones de colocación de mensajes en colas.

Utilice los subtemas siguientes para obtener información sobre la gestión de colas de mensajes y otras características suministradas por IBM MQ.

## **Conceptos relacionados**

[Introducción a IBM MQ](#)

[Dónde encontrar información acerca de los requisitos del producto y el soporte](#)

## **Tareas relacionadas**

[Planificación de una arquitectura de IBM MQ](#)

## **Referencia relacionada**

[“Características principales y ventajas de la colocación de mensajes en colas” en la página 7](#)

Esta información destaca algunas de las características y las ventajas de la colocación de mensajes en colas. Describe características como la seguridad y la integridad de los datos en la colocación de mensajes en colas.

## Introducción a la colocación de mensajes en colas

---

Los productos de IBM MQ permiten que los programas se comuniquen entre sí a través de una red de componentes diferentes (procesadores, sistemas operativos, subsistemas y protocolos de comunicación) utilizando una interfaz de programación de aplicaciones coherente.

Las aplicaciones diseñadas y escritas con esta interfaz se conocen como aplicaciones de *colocación de mensajes en colas* porque utilizan el tipo de *mensajería* y el tipo *colocación en colas*:

- Mensajería significa que los programas se comunican entre sí enviando datos en mensajes, en lugar de invocándose entre sí.
- Encolamiento significa poner mensajes en colas a modo de almacenamiento, lo que permite a los programas ejecutar independientemente unos de otros, a velocidades y tiempos diferentes, en ubicaciones distintas y sin existir una conexión lógica entre ellos.

Durante muchos años se ha utilizado la colocación de mensajes en colas en el proceso de datos. Actualmente, es lo que se utiliza con mayor frecuencia en el correo electrónico. Sin la colocación en colas, enviar un mensaje electrónico a través de largas distancias requiere que cada nodo en la ruta esté disponible para reenviar mensajes, y que los destinatarios estén conectados y sean conscientes del hecho de que se les está intentando enviar un mensaje. En un sistema de colocación en colas, los mensajes se almacenan en nodos intermedios hasta que el sistema esté listo para reenviarlos. En el destino final se almacenan en un buzón hasta que el destinatario esté listo para leerlos.

Aun así, muchas transacciones empresariales complejas se procesan hoy en día sin la colocación en colas. En una red grande, el sistema podría mantener miles de conexiones en un estado preparado para utilizarse. Si una parte del sistema sufre un problema, muchas partes del sistema podrían quedar inservibles.

La colocación de mensajes en colas se puede considerar como un correo electrónico para los programas. En un entorno de colas de mensaje, cada programa que conforma una parte de una serie de aplicaciones realiza una función bien definida e independiente en respuesta a una solicitud específica. Para comunicarse con otro programa, un programa debe transferir un mensaje a una cola predefinida. El otro programa recupera el mensaje de la cola y procesa las solicitudes así como la información contenida

en el mensaje. Por lo tanto, la colocación de mensajes en colas es un tipo de comunicación programa a programa.

La colocación en colas es un mecanismo mediante el cual los mensajes se guardan hasta que la aplicación esté preparada para utilizarlos. La colocación en colas permite realizar lo siguiente:

- Comunicarse entre programas (que pueden estar ejecutándose en entornos diferentes) sin tener que escribir el código de comunicación.
- Seleccionar el orden en el que un programa procesa los mensajes.
- Equilibrar las cargas en un sistema organizando que más de un programa atienda una cola cuando el número de mensajes excede un umbral.
- Aumentar la disponibilidad de las aplicaciones organizando que un sistema alternativo atienda las colas si el sistema primario no está disponible.

## ¿Qué es una cola de mensajes?

Una cola de mensajes, conocida simplemente como cola es un destino específico al que se pueden enviar mensajes. Los mensajes se acumulan en las colas hasta que los recuperan programas que atienden esas colas.

Las colas residen en un gestor de colas y son gestionada por él; (consulte [“Terminología sobre la colocación de mensajes en colas”](#) en la página 9). La naturaleza física de una cola depende del sistema operativo en el que se ejecuta el gestor de colas. Una cola puede ser un área de almacenamiento intermedio volátil en la memoria de un sistema o un conjunto de datos en un dispositivo de almacenamiento permanente (por ejemplo un disco). La gestión física de las colas es responsabilidad del gestor de colas y no es visible para los programas de aplicación que participan.

Los programas sólo acceden a las colas a través de los servicios externos del gestor de colas. Pueden abrir una cola, transferirle mensajes, extraer mensajes de ella y cerrarla. También pueden establecer y realizar consultas sobre los atributos de las colas.

## Diferentes estilos de colas de mensaje

### Punto a punto

Un mensaje se coloca en la cola y una aplicación recibe ese mensaje.

En la mensajería punto a punto, una aplicación emisora debe tener información sobre la aplicación receptora antes de que pueda enviar un mensaje a esa aplicación. Por ejemplo, la aplicación emisora puede necesitar saber el nombre de la cola a la que enviar la información, y también puede especificar un nombre de gestor de colas.

### Publicación/suscripción

Una copia de cada mensaje publicado por una aplicación de publicación se entrega a cada aplicación interesada. Puede que haya muchas, una o ninguna aplicación interesada. En la publicación/suscripción, una aplicación interesada se conoce como un suscriptor y los mensajes se ponen en cola en una cola identificada mediante una suscripción.

La mensajería de publicación/suscripción permite separar el proveedor de información de los consumidores de dicha información. Para poder enviar y recibir información, no es necesario que la aplicación de envío y la de recepción sepan mucho la una de la otra. Para obtener más información, consulte [“Mensajería de publicación/suscripción”](#) en la página 63.

## Ventajas de la colocación de mensajes en colas para el diseñador y el desarrollador de aplicaciones

IBM MQ permite que los programas de aplicación utilicen la *colocación de mensajes en colas* para participar en el proceso controlado por mensajes. Los programas de aplicación pueden comunicarse entre distintas plataformas utilizando los productos de software colocación de mensajes en colas adecuados. Por ejemplo, las aplicaciones de z/OS pueden comunicarse a través de IBM MQ for z/OS.

Las aplicaciones son independientes del funcionamiento de las comunicaciones subyacentes. Algunas de las ventajas de la colocación de mensajes en colas son las siguientes:

- Puede diseñar aplicaciones utilizando pequeños programas que se pueden compartir entre muchas aplicaciones.
- Puede crear rápidamente nuevas aplicaciones reutilizando estos bloques de creación.
- Las aplicaciones escritas para utilizar técnicas de colocación de mensajes en colas no se ven afectadas por los cambios en el modo en que los gestores de colas funcionan.
- No es necesario utilizar ningún protocolo de comunicación. El gestor de colas se encarga automáticamente de todos los aspectos de la comunicación.
- Los programas que reciben mensajes no tienen que estar en ejecución en el momento en que se les envían mensajes. Los mensajes se retienen en las colas.

Los diseñadores pueden reducir el coste de sus aplicaciones porque el desarrollo es más rápido, se necesitan menos desarrolladores y los requisitos de aptitudes de programación son menores que los de las aplicaciones que no utilizan colas de mensajes.

IBM MQ implementa una interfaz de programación de aplicaciones común conocida como *interfaz de cola de mensajes* (o MQI) dondequiera que se ejecuten las aplicaciones. Esto facilita el traslado de programas de aplicación de una plataforma a otra.

Para obtener detalles sobre MQI, consulte [Visión general de la interfaz de cola de mensajes](#).

## Características principales y ventajas de la colocación de mensajes en colas

Esta información destaca algunas de las características y las ventajas de la colocación de mensajes en colas. Describe características como la seguridad y la integridad de los datos en la colocación de mensajes en colas.

Las características principales de las aplicaciones que utilizan las técnicas de colocación de mensajes en colas son:

- [“No hay conexiones directas entre programas” en la página 7](#)
- [“Comunicación independiente del tiempo” en la página 8](#)
- [“Programas pequeños” en la página 8](#)
- [“Procesamiento dirigido por mensajes” en la página 8](#)
- [“Procesamiento controlado por sucesos” en la página 9](#)
- [“Prioridad de mensaje” en la página 9](#)
- [“Seguridad” en la página 9](#)
- [“Integridad de datos” en la página 9](#)
- [“Soporte de recuperación” en la página 9](#)

**Nota:** A la hora de considerar clientes y servidores de IBM MQ, no tiene que cambiar una aplicación de servidor para dar soporte a IBM MQ MQI clients adicionales en plataformas nuevas. Asimismo, IBM MQ MQI client puede, sin cambios, funcionar con tipos adicionales de servidores.

### No hay conexiones directas entre programas

La colocación de mensajes en colas es una técnica para la comunicación indirecta de programa a programa. Se puede utilizar dentro de cualquier aplicación en la que los programas se comuniquen entre sí. La comunicación se produce cuando un programa transfiere mensajes a una cola (propiedad de un gestor de colas) y otro programa obtiene los mensajes de la cola.

Los programas pueden obtener mensajes que fueron colocados en una cola por otros programas. Los otros programas pueden estar conectados al mismo gestor de colas como programa receptor, o bien a otro gestor de colas. Este otro gestor de colas puede estar en otro sistema, en un sistema informático diferente, o incluso en una empresa distinta.

No hay conexiones físicas entre los programas que se comunican utilizando colas de mensajes. Un programa envía mensajes a una cola propiedad de un gestor de colas y otro programa recupera mensajes de la cola (consulte Figura 1 en la página 8).

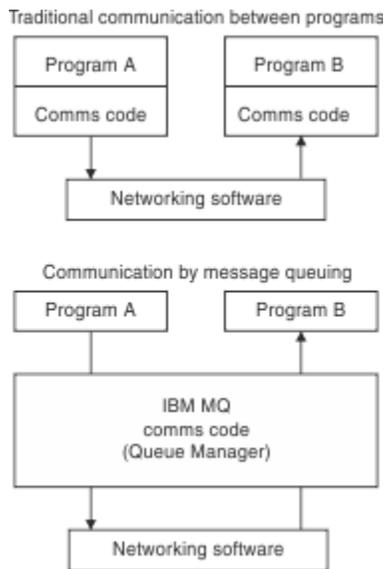


Figura 1. Colocación de mensajes en colas en comparación con la comunicación tradicional

Al igual que ocurre con el correo electrónico, los mensajes individuales que forman parte de una transacción viajan a través de una red de acuerdo con un sistema de almacén y reenvío. Si un enlace entre los nodos falla, el mensaje se conserva hasta que el enlace se ha restaurado o el operador o el programa redirige el mensaje.

El mecanismo por el que un mensaje se mueve de cola en cola queda oculto en los programas. Por lo tanto, los programas son más simples.

## Comunicación independiente del tiempo

Los programas que solicitan que otros realicen trabajos no tienen que esperar la respuesta a su solicitud. Pueden realizar otros trabajos y procesar la respuesta ya sea cuando llegue o en un momento posterior. Cuando se escribe una aplicación de mensajería, no es necesario saber (ni preguntarse) cuándo un programa envía un mensaje o cuando el destino puede recibir el mensaje. El mensaje no se pierde; el gestor de colas lo conserva hasta que el destinatario esté listo para procesarlo. El mensaje permanece en la cola hasta que un programa lo elimina. Esto significa que los programas de aplicación emisor y receptor están separados: el emisor pueda seguir procesando sin tener que esperar a que el receptor acuse recibo del mensaje. De hecho, no es necesario que la aplicación de destino esté ejecutándose cuando se envía el mensaje. Puede recuperar el mensaje una vez que se ha iniciado.

## Programas pequeños

La colocación de mensajes en colas permite utilizar la ventajas de los programas pequeños e independientes. En lugar de un único gran programa que realiza secuencialmente todas las partes de un trabajo, puede distribuir el trabajo en varios programas más pequeños e independientes. El programa solicitante envía mensajes a cada uno de los programas separados, solicitándoles que realicen su función; cuando cada programa está completo, los resultados se envían de nuevo en forma de uno o más mensajes.

## Procesamiento dirigido por mensajes

Al llegar a una cola, los mensajes pueden iniciar automáticamente una aplicación utilizando un mecanismo llamado *desencadenamiento*. Si es necesario, las aplicaciones se pueden detener después de haber procesado el mensaje o mensajes.

## Procesamiento controlado por sucesos

Los programas se pueden controlar de acuerdo con el estado de las colas. Por ejemplo, puede disponer que un programa se inicie tan pronto como un mensaje llegue a una cola, o puede especificar que el programa no se inicie hasta que haya, por ejemplo, 10 los mensajes por encima de una determinada prioridad en la cola, o 10 mensajes de cualquier prioridad en la cola.

## Prioridad de mensaje

Un programa puede asignar una prioridad a un mensaje cuando pone el mensaje en una cola. Esto determina la posición en la cola a la que se añade el nuevo mensaje.

Los programas pueden obtener mensajes de una cola en el orden en el que los mensajes están en la cola, o bien obteniendo un mensaje específico. (Un programa puede desear obtener un mensaje específico si está buscando la respuesta a una solicitud que envió anteriormente.)

## Seguridad

Se proporcionan funcionalidades de seguridad, incluyendo la autenticación de aplicaciones cuando utilizan un gestor de colas, comprobaciones de autorización cuando utilizan recursos como, por ejemplo, una cola en el gestor de colas y cifrado de datos de mensajes en la red y dentro de las colas. Para obtener más información sobre seguridad, consulte [Seguridad de canal](#).

## Integridad de datos

La integridad de los datos se proporciona mediante unidades de trabajo. La sincronización del inicio y el final de las unidades de trabajo está totalmente soportada como una opción en cada MQGET o MQPUT, permitiendo que el resultado de la unidad de trabajo se confirma o se restituya. El soporte de punto de sincronismo funciona de forma interna o externa a IBM MQ en función de la forma de coordinación de puntos de sincronismo seleccionada para la aplicación.

## Soporte de recuperación

Para que sea posible la recuperación, se registran todas las actualizaciones de IBM MQ persistentes. Si es necesaria la recuperación, todos los mensajes permanentes se restauran, todas las transacciones en curso se retrotraen y todas las confirmaciones y restituciones del punto de sincronismo se manejan de la manera habitual empleada por el gestor de puntos de sincronismo que tiene el control. Para obtener más información sobre mensajes permanentes, consulte [Persistencia de mensajes](#).

## Terminología sobre la colocación de mensajes en colas

Esta información ofrece una perspectiva de algunos términos que se emplean con relación a la colocación de mensajes en colas.

Estos incluyen:

- [Canales](#)
- [Clúster](#)
- [IBM MQ MQI client](#)
-  [Transferencia a colas dentro del grupo](#)
- [Mensaje](#)
- [Agente de canal de mensajes](#)
- [Descriptor de mensaje](#)
- [Punto a punto](#)
- [Publicar/Suscribir](#)
- [Cola](#)

- [Gestor de colas](#)
-  [Grupo de compartición de colas](#)
-  [Cola compartida](#)
- [Suscripción](#)
- [Tema](#)

## Canales

Los canales se utilizan para mover los mensajes desde un gestor de colas a otro y protegen las aplicaciones de los protocolos de comunicaciones subyacentes. Los gestores de colas pueden estar en sistemas diferentes de la misma plataforma o de plataformas distintas. Los mensajes que se envían pueden tener su origen en muchas ubicaciones:

- Programas de aplicación escritos por el usuario que transfieren datos de un nodo a otro.
- Aplicaciones de administración escritas por usuario que utilizan mandatos PCF o MQAI.
- IBM MQ Explorer.
- Gestores de colas que envían mensajes de sucesos de instrumentación a otro gestor de colas.
- Gestores de colas que envían mandatos de administración remotos a otro gestor de colas. Por ejemplo, utilizando los mandatos MQSC o la administrative REST API.

Para obtener más información sobre los canales, consulte [“Definiciones de canal” en la página 32](#).

## Clúster

Un *clúster* es una red de gestores de colas que están asociados lógicamente de alguna forma.

En una red IBM MQ que utiliza colas distribuidas sin agrupación en clúster, cada gestor de colas es independiente. Si un gestor de colas necesita enviar mensajes a otro, debe haber definido una cola de transmisión y un canal en el gestor de colas remoto.

Hay dos razones distintas para utilizar clústeres: reducir la administración del sistema y mejorar la disponibilidad y el equilibrio de la carga de trabajo.

En cuanto establezca incluso el clúster más pequeño, se beneficiará de una administración simplificada del sistema. Los gestores de colas que forman parte de un clúster necesitan menos definiciones y, por lo tanto, el riesgo de cometer un error en las definiciones es menor.

Puede obtener información adicional sobre agrupaciones en clúster consultando [Clústeres](#).

## IBM MQ MQI client

IBM MQ MQI *clientes* son componentes instalables de forma independiente de IBM MQ. Un cliente MQI permite ejecutar aplicaciones IBM MQ con un protocolo de comunicaciones, interactuar con uno o varios servidores de interfaz de cola de mensajes (MQI) en otras plataformas y conectarse a sus gestores de colas.

Para obtener detalles completos sobre cómo instalar y utilizar los componentes de IBM MQ MQI client, consulte los temas siguientes:

-  [Instalación de un cliente IBM MQ en AIX](#)
-  [Instalación de un cliente IBM MQ en Linux®](#)
-  [Instalación de un cliente IBM MQ en Windows](#)
-  [Instalación de un cliente IBM MQ en IBM i](#)

y [Configuración de las conexiones entre el servidor y el cliente](#).

## Transferencia a colas entre grupos



Los gestores de colas de un grupo de compartición de colas se pueden comunicar utilizando canales normales o bien pueden utilizar una técnica que se denomina *transferencia a colas dentro del grupo* (IGQ), que cual permite realizar una transferencia rápida de mensajes sin definir canales. Esto solo se aplica a IBM MQ for z/OS.

Para obtener más información sobre la transferencia a colas dentro del grupo, consulte [“Transferencia a colas entre grupos”](#) en la página 219.

## Mensaje

En la colocación de mensajes en colas, un mensaje es una colección de datos enviados por un mensaje y dirigidos a otro programa. Consulte [Mensajes de IBM MQ](#).

Para obtener información sobre los tipos de mensajes, consulte [Tipos de mensajes](#).

## Agente de canal de mensajes

Un agente de canal de mensajes es un extremo de un canal. Un par de agentes de canal de mensajes, uno emisor y otro receptor, configuran un canal y mueven mensajes de un gestor de colas a otro.

Para obtener información sobre cómo se utilizan los agentes de canal de mensajes, consulte [Introducción a la gestión de colas distribuidas](#).

## Descriptor de mensaje

Un mensaje de IBM MQ consta de información de control y datos de aplicación.

La información de control está definida en una estructura de descriptor de mensaje (MQMD) y contiene elementos como:

- El tipo de mensaje
- Un identificador del mensaje
- La prioridad de entrega del mensaje

La estructura y el contenido de los datos de la aplicación están determinados por los programas participantes, no por IBM MQ.

Para obtener más información, consulte [MQMD](#).

## Mensajería punto a punto

En la mensajería punto a punto, cada mensaje se traslada de una aplicación de producción a una aplicación de consumo. Los mensajes se transfieren mediante la aplicación de producción que coloca los mensajes en una cola, y la aplicación de consumo los obtiene de dicha cola.

## Mensajería de publicación/suscripción

En la mensajería de publicación/suscripción, una copia de cada mensaje publicado por una aplicación de publicación se entrega a todas las aplicaciones interesadas. Puede haber muchas, una o ninguna aplicación interesada. En la publicación/suscripción, una aplicación interesada se conoce como un suscriptor y los mensajes se ponen en cola en una cola identificada mediante una suscripción.

Para obtener más información, consulte [“Mensajería de publicación/suscripción”](#) en la página 63.

## Cola

Un destino designado al que se pueden enviar mensajes. Los mensajes se acumulan en las colas hasta que los recuperan programas que atienden esas colas.

Para obtener más información, consulte [“Colas” en la página 20](#).

## Gestor de colas

Un *gestor de colas* es un programa del sistema que proporciona servicios de colocación de mensajes en colas a las aplicaciones.

Proporciona una interfaz de programación de aplicaciones para que los programas puedan transferir y obtener mensajes de las colas. Un gestor de colas proporciona funciones adicionales para que los administradores puedan crear colas nuevas, alterar las propiedades de las colas existentes y controlar el funcionamiento del gestor de colas.

Para que los servicios de colas de mensajes de IBM MQ estén disponibles en un sistema, debe haber un gestor de colas en ejecución. Puede haber más de un gestor de colas en ejecución en un único sistema (por ejemplo, para separar un sistema de prueba de un sistema *activo*). En una aplicación, cada gestor de colas está identificado con un *manejador de conexión (Hconn)*.

Muchas aplicaciones diferentes pueden utilizar los servicios del gestor de colas al mismo tiempo y estas aplicaciones pueden no estar relacionadas en absoluto. Para que un programa utilice los servicios de un gestor de colas, debe establecer una conexión con ese gestor de colas.

Para que las aplicaciones envíen mensajes a las aplicaciones que están conectadas a otros gestores de colas, los gestores de colas deben ser capaces de comunicarse entre sí. IBM MQ implementa un protocolo de *almacén y reenvío* para garantizar los mensajes se entregan de un modo seguro entre dichas aplicaciones.

Para obtener más información, consulte [“Gestores de colas” en la página 29](#).

## Grupo de compartición de colas



Los gestores de colas que pueden acceder al mismo conjunto de colas compartidas forman un grupo denominado *grupo de compartición de colas (QSG)*. Se comunican entre sí con un recurso de acoplamiento (CF) que almacena las colas compartidas. Esto solo se aplica a IBM MQ for z/OS.

Para obtener más información, consulte [“Colas compartidas y grupos de compartición de colas” en la página 171](#).

## Cola compartida



Una *cola compartida* es un tipo de cola local con mensajes a los que pueden acceder uno o varios gestores de colas que están en un sysplex. No es lo mismo que una cola compartida por más de una aplicación, que utiliza el mismo gestor de colas. Esto solo se aplica a IBM MQ for z/OS.

## Suscripción

Una aplicación de publicación/suscripción puede registrar un interés en mensajes sobre temas específicos. Cuando una aplicación hace esto, se la denomina suscriptor y el término suscripción define cómo se ponen en cola los mensajes coincidentes para su proceso.

Una suscripción contiene información sobre la identidad del suscriptor y la identidad de la cola de destino en la que se van a colocar publicaciones. También contiene información sobre cómo se colocará una publicación en la cola de destino.

Para obtener más información, consulte [“Suscriptores y suscripciones” en la página 66](#).

## Tema

Un tema es una serie de caracteres que describe el asunto de la información que se publica en un mensaje de publicación/suscripción.

Los temas son fundamentales para la entrega satisfactoria de mensajes en un sistema de publicación/ suscripción. En lugar de incluir una dirección de destino específica en cada mensaje, un publicador asigna un tema a cada mensaje. El gestor de colas correlaciona el tema con una lista de suscriptores que se han suscrito a ese tema, y entrega el mensaje a cada uno de esos suscriptores.

Para obtener más información, consulte [“Temas” en la página 69](#).

## Mensajes y colas

Los mensajes y las colas son los componentes básicos de un sistema de colocación de mensajes en colas.

### ¿Qué es un mensaje?

Un *mensaje* es una serie de bytes que tiene un significado para las aplicaciones que lo utilizan. Los mensajes se utilizan para transferir información de un programa de aplicación a otro (o a distintas partes de la misma aplicación). Las aplicaciones pueden estar ejecutándose en la misma o en distintas plataformas.

Un mensaje IBM MQ se compone de:

- *Los datos de la aplicación.* El contenido y la estructura de los datos de la aplicación los definen los programas de aplicación que los utilizan.
- *Un descriptor de mensaje.* El descriptor del mensaje identifica el mensaje y contiene información de control adicional, como el tipo de mensaje y la prioridad que ha asignado al mensaje la aplicación emisora.

El formato del descriptor de mensaje se define en IBM MQ. Si desea una descripción completa del descriptor de mensajes, consulte [MQMD - descriptor de mensajes](#).

- *Propiedades de mensaje.* Metadatos del mensaje. El contenido de las propiedades del mensaje lo definen los programas de aplicación de las utilizan. Puede obtener información adicional consultando [Propiedades de un mensaje](#).

### Longitudes de los mensajes

La longitud máxima predeterminada de un mensaje es de 4 MB, aunque puede aumentarla a una longitud máxima de 100 MB (donde 1 MB equivale a 1.048.576 bytes). En la práctica, la longitud del mensaje puede estar limitada por:

- La longitud máxima de mensajes definida para la cola receptora
- La longitud máxima de mensajes definida para el gestor de colas
- La longitud máxima de mensajes definida por la cola
- La longitud máxima de mensajes definida por la aplicación emisora o receptora
- La cantidad de almacenamiento disponible para el mensaje.

Pueden ser necesarios varios mensajes para enviar toda la información que una aplicación necesita.

### ¿Cómo envían y reciben mensajes las aplicaciones?

Los programas de aplicación envían y reciben mensajes utilizando las **llamadas MQI**.

Por ejemplo, para transferir un mensaje a una cola, una aplicación:

1. Abre la cola necesaria emitiendo una llamada MQOPEN de MQI
2. Emite una llamada MQPUT de MQI para transferir el mensaje a la cola

Otra aplicación puede recuperar el mensaje de la misma cola emitiendo una llamada MQGET de MQI.

Para obtener más información sobre las llamadas MQI, consulte [Llamadas MQI](#).

## ¿Qué es una cola?

Una *cola* es una estructura de datos que se utiliza para almacenar mensajes.

Cada cola es propiedad de un *gestor de colas*. El gestor de colas es responsable del mantenimiento de las colas de su propiedad y del almacenamiento de todos los mensajes que recibe en las colas adecuadas. Los mensajes se pueden transferir a la cola mediante programas de aplicación o mediante un gestor de colas como parte de sus operaciones normales.

## Colas predefinidas y colas dinámicas

Las colas pueden distinguirse por la forma en que se han creado:

- Las **colas predefinidas** las crea un administrador mediante los mandatos MQSC o PCF adecuados. Las colas predefinidas son permanentes; existen independientemente de las aplicaciones que las utilizan y sobreviven a los reinicios de IBM MQ.
- Las **colas dinámicas** se crean cuando una aplicación emite una petición MQOPEN especificando el nombre de una *cola modelo*. La cola creada se basa en una *plantilla de definición de cola* que es la cola modelo. Puede crear una cola modelo utilizando el mandato MQSC DEFINE QMODEL. Los atributos de una cola modelo, por ejemplo, el número máximo de mensajes que se pueden almacenar en la cola, son heredados por cualquier cola dinámica que se cree a partir de la cola modelo.

Las colas modelo tienen un atributo que especifica si la cola dinámica va a ser persistente o temporal. Las colas persistentes perduran después de los reinicios del gestor de colas y de la aplicación; las colas temporales se pierden al reiniciar.

## Recuperación de mensajes de colas

Las aplicaciones debidamente autorizadas pueden recuperar mensajes de una cola en función de los siguientes algoritmos de recuperación:

- Primero en entrar, primero en salir (FIFO).
- Prioridad del mensaje, según lo definido en el descriptor del mensaje. Los mensajes que tienen la misma prioridad se recuperan según el método FIFO.
- Una petición de programa para un mensaje específico.

La petición MQGET de la aplicación determina el método utilizado.

## Objetos de IBM MQ

---

Los gestores de colas definen las propiedades de los objetos de IBM MQ. Los valores de estas propiedades afectan al modo en que IBM MQ procesa estos objetos. Puede crear y gestionar objetos utilizando mandatos e interfaces de IBM MQ. Desde sus aplicaciones, utilice la interfaz de cola de mensajes (MQI) para controlar los objetos. Los objetos se identifican mediante un *descriptor de objeto* (MQOD) de IBM MQ cuando se direccionan desde un programa.

## Administración de objetos

La administración de objetos incluye las tareas siguientes:

- Iniciar y detener gestores de colas.
- Crear objetos, especialmente colas, para aplicaciones.
- Mostrar o alterar los atributos de objetos.
- Suprimiendo objetos.
- Trabajar con canales para crear vías de comunicación con gestores de colas existentes en otros sistemas (remotos).
- Crear *clústeres* de gestores de colas para simplificar el proceso global de administración o para equilibrar la carga de trabajo.

Con la excepción de las colas dinámicas, los objetos deben definirse ante el gestor de colas para poder trabajar con ellos.

Cuando utiliza un mandato de IBM MQ para llevar a cabo una operación de administración de objetos, el gestor de colas comprueba si dispone del nivel de autorización necesario para realizar la operación. Del mismo modo, cuando una aplicación utiliza la llamada MQOPEN para abrir un objeto, el gestor de colas comprueba si la aplicación dispone del nivel de autorización necesario antes de permitir el acceso a dicho objeto. Las comprobaciones se realizan en el nombre del objeto que se abre.

Puede definir y gestionar objetos mediante los métodos siguientes:

- Los mandatos PCF que se describen en [Guía de consulta sobre formatos de mandatos programables y Automatización de tareas de administración](#)
- Los mandatos MQSC que se describen en [Mandatos MQSC](#)
-  Los paneles de operaciones y de control de IBM MQ for z/OS , descritos en [Administración IBM MQ for z/OS](#)
-   IBM MQ Explorer (solamente para sistemas Windows y Linux for Intel). Para obtener más información, consulte [Introducción a MQ Explorer](#).

También puede gestionar objetos mediante los métodos siguientes:

- Con los mandatos de control, que se escriben desde un teclado. Consulte [Administración de IBM MQ for Multiplatforms utilizando mandatos de control](#).
- Las llamadas de la interfaz de administración de IBM MQ (MQAI) en un programa. Consulte [Interfaz de administración de IBM MQ \(MQAI\)](#).

 En el caso de las secuencias de mandatos de IBM MQ en AIX, Linux, and Windows, puede utilizar el recurso MQSC para ejecutar una serie de mandatos que se guardan en un archivo. Para obtener más información, consulte [Administración de IBM MQ utilizando mandatos MQSC](#).

 En el caso de las secuencias de mandatos de IBM MQ para IBM i que utiliza regularmente, puede escribir programas CL. Para obtener más información, consulte [Gestión de IBM MQ for IBM i utilizando mandatos CL](#).

 En el caso de las secuencias de mandatos de IBM MQ for z/OS que utiliza regularmente, puede escribir programas de administración que creen mensajes que contienen mandatos y que colocan estos mensajes en la cola de entrada del mandato del sistema. El gestor de colas procesa los mensajes en esta cola del mismo modo que procesa los mandatos entrados desde la línea de mandatos o bien desde las operaciones y los paneles de control. Esta técnica se describe en [Escribir programas para administrar IBM MQ](#) y se muestra en la aplicación de ejemplo Mail Manager que se suministra con IBM MQ for z/OS. Para obtener una descripción de este ejemplo, consulte [Programas de ejemplo para IBM MQ for z/OS](#).

## Atributos de objetos

Las propiedades de un objeto se definen mediante sus atributos. Algunas pueden especificarse, mientras que otras sólo pueden verse.

Por ejemplo, la longitud máxima de mensaje que una cola puede admitir está definida por su atributo **MaxMsgLength**; este atributo se puede especificar cuando se crea una cola. El atributo **DefinitionType** especifica cómo se ha creado la cola; este atributo sólo se puede visualizar.

En IBM MQ, hay dos formas de referirse a un atributo:

- Utilizando su nombre PCF, por ejemplo, **MaxMsgLength**.
- Utilizando su nombre de mandato MQSC, por ejemplo, MAXMSGL.

## Grupos de compartición de colas



Los gestores de colas que pueden acceder al mismo conjunto de colas compartidas forman un grupo llamado *grupo de compartición de colas* (QSG) y se comunican entre sí utilizando un recurso de acoplamiento (CF) que almacena las colas compartidas. Tenga en cuenta que un QSG no es estrictamente un objeto.

Una cola compartida es un tipo de cola local con mensajes a los que pueden acceder uno o varios gestores de colas que están en un grupo de compartición de colas. No es lo mismo que una cola compartida por más de una aplicación, que utiliza el mismo gestor de colas.

Los grupos de compartición de colas tienen un nombre de hasta cuatro caracteres. El nombre debe ser exclusivo en la red y debe ser diferente de los nombres de gestor de colas.

**Importante:** Las colas compartidas y los grupos de compartición de colas solo están soportados en IBM MQ for z/OS.

Consulte [“Colas compartidas y grupos de compartición de colas”](#) en la página 171 para obtener más información.

## Objetos predeterminados del sistema

Los *objetos predeterminados del sistema* son un conjunto de definiciones de objetos que se crean automáticamente siempre que se crea un gestor de colas. Puede copiar y modificar cualquiera de estas definiciones de objetos para utilizarlas en aplicaciones de su instalación.

Los nombres de los objetos predeterminados tienen la raíz SYSTEM; por ejemplo, la cola local predeterminada es SYSTEM.DEFAULT.LOCAL.QUEUE y el canal receptor predeterminado es SYSTEM.DEF.RECEIVER. No se puede cambiar el nombre de estos objetos; es obligatorio que los objetos predeterminados tengan estos nombres.

Cuando se define un objeto, todos los atributos que no se especifican explícitamente se copian del objeto predeterminado adecuado. Por ejemplo, si define una cola local, los atributos que no especifique se tomarán de la cola predeterminada SYSTEM.DEFAULT.LOCAL.QUEUE.

Consulte [Objetos del sistema y predeterminados](#) para obtener más información.

## Tipos de objetos

Muchas de las tareas de administración consisten en manipular varios tipos de IBM MQ *objetos*.

Para obtener información sobre la denominación de objetos IBM MQ, consulte [“Denominación de objetos de IBM MQ”](#) en la página 37.

Para obtener información acerca de los objetos predeterminados creados en un gestor de colas, consulte [“Objetos predeterminados del sistema”](#) en la página 16.

Para obtener información sobre los distintos tipos de objetos de IBM MQ, consulte lo siguiente:

### Objetos de información de autenticación

Un objeto de información de autenticación proporciona las definiciones necesarias para realizar la comprobación de revocación de certificados.

El objeto de información de autenticación del gestor de colas forma parte del soporte de IBM MQ para la seguridad de la capa de transporte (TLS). Proporciona las definiciones necesarias para comprobar los certificados revocados. Las autoridades de certificación revocan los certificados que ya no son fiables.

Puede utilizar el mandato MQSC **DEFINE AUTHINFO** para definir un objeto de información de autenticación. Para obtener más información sobre los atributos de los objetos de información de autenticación, consulte [DEFINE AUTHINFO](#).

Puede utilizar los siguientes mandatos de control de IBM MQ con un objeto de información de autenticación:

- **setmqaut** (otorgar o revocar autorización)
- **dspmqa** (visualizar autorización de objeto)
- **dmpmqaut** (volcar autorizaciones)
- **rcrmqobj** (volver a crear objeto)
- **rcdmqing** (registrar imagen de soporte)
- **dspmqfls** (visualizar nombres de archivos)

Para obtener una visión general de TLS y el uso de los objetos de información de autenticación, consulte [Conceptos de seguridad de la capa de transporte \(TLS\)](#) y [Protocolos de seguridad TLS en IBM MQ](#).

## Canales

Los canales son objetos que proporcionan una vía de comunicación de un gestor de colas a otro.

Consulte [“Canales”](#) en la página 30 para obtener más información.

## Objetos de información de comunicación

IBM MQ Multicast ofrece baja latencia, alta diseminación y mensajería de multidifusión fiable. Se precisa un objeto de información de comunicación (COMMINFO) para utilizar la transmisión Multidifusión.

Consulte [“IBM MQ Multicast”](#) en la página 108 para obtener más información.

Un objeto COMMINFO es un objeto de IBM MQ que contiene los atributos asociados a la transmisión de multidifusión. Para obtener más información sobre estos atributos, consulte [DEFINE COMMINFO](#). Para obtener más información sobre cómo crear un objeto COMMINFO, consulte [Iniciación con la multidifusión](#).

## Escuchas

Los *escuchas* son procesos que aceptan peticiones de red de otros gestores de colas o aplicaciones cliente e inician los canales asociados.

Los *procesos de escucha* se pueden iniciar utilizando el mandato de control **runmqtsr**.

Los *objetos de escucha* son objetos de IBM MQ que le permiten gestionar el inicio y la detención de procesos de escucha desde el ámbito de un gestor de colas. Al definir los atributos de un objeto de escucha, hace lo siguiente:

- Configurar el proceso de escucha.
- Especificar si el proceso de escucha se inicia y se detiene automáticamente cuando se inicia y se detiene el gestor de colas.

**Importante:**  Los objetos de escucha no están soportados en IBM MQ for z/OS. Para obtener más información sobre cómo IBM MQ for z/OS implementa la escucha, utilizando el iniciador de canal, consulte [“El iniciador de canal en z/OS”](#) en la página 167.

## Listas de nombres

Una *lista de nombres* es un objeto de IBM MQ que contiene una lista de nombres de clúster, nombres de colas o nombres de objetos de información de autenticación. En un clúster, se puede utilizar para identificar una lista de clústeres para los que el gestor de colas contiene los repositorios.

Una lista de nombres es un objeto de IBM MQ que contiene una lista de otros objetos de IBM MQ. Normalmente, las listas de nombres las utilizan aplicaciones como los supervisores desencadenantes y se utilizan para identificar un grupo de colas. La ventaja de utilizar una lista de nombres es que se mantiene independientemente de las aplicaciones, es decir, se puede actualizar sin detener ninguna de

las aplicaciones que la utilizan. Además, si una aplicación no se ejecuta correctamente, ello no afecta a la lista de nombres y otras aplicaciones podrán seguir utilizándola.

Las listas de nombres también se utilizan con los clústeres de gestores de colas para mantener una lista de clústeres a los que hace referencia más de un objeto IBM MQ .

Puede definir y modificar listas de nombres utilizando los mandatos MQSC [DEFINE NAMELIST](#) y [ALTER NAMELIST](#) .

**Nota:**  De forma alternativa, en z/OS, puede utilizar los paneles de operaciones y control de IBM MQ for z/OS

Los programas pueden utilizar MQI para averiguar qué colas están incluidas en estas listas de nombres. La organización de las listas de nombres es responsabilidad del diseñador de aplicaciones y administrador del sistema.

Para obtener una lista de los atributos de lista de nombres disponibles para utilizar, consulte [Atributos para listas de nombres](#),

## Definiciones de procesos

Los objetos de definición de proceso permiten que las aplicaciones se inicien sin necesidad de que intervenga el operador definiendo los atributos de la aplicación para que los utilice el gestor de colas.

El objeto de definición de proceso define una aplicación que se inicia como respuesta a un suceso desencadenante en un gestor de colas de IBM MQ. Los atributos de la definición de proceso incluyen el ID de aplicación, el tipo de aplicación y los datos específicos de la aplicación. Para obtener más información, consulte *Colas de inicio* en [“Colas utilizadas para fines específicos por IBM MQ”](#) en la página 27.

Para que una aplicación se pueda iniciar sin la necesidad de la intervención del operador, tal como se describe en [Inicio de aplicaciones de IBM MQ utilizando desencadenantes](#), los atributos de la aplicación deben ser conocidos para el gestor de colas. Estos atributos se definen en un *objeto de definición de proceso*.

El atributo **ProcessName** es fijo cuando se crea el objeto. Sin embargo, puede cambiar otros atributos utilizando los mandatos de IBM MQ.

**Nota:**  De forma alternativa, en z/OS, puede utilizar los paneles de operaciones y de control de IBM MQ for z/OS .

Puede averiguar los valores de todos los atributos utilizando [MQINQ - Consultar atributos de objetos](#).

Para obtener una lista de los atributos de definición de proceso disponibles para utilizar, consulte [Atributos para definiciones de proceso](#).

## Colas

Un IBM MQ *cola* es un objeto con nombre en el que las aplicaciones pueden transferir mensajes y del que las aplicaciones pueden obtener mensajes.

Consulte [“Colas”](#) en la página 20 para obtener más información.

## Gestores de colas

IBM MQ Los gestores de colas proporcionan servicios de colocación en colas a las aplicaciones y gestionan las colas que les pertenecen.

Consulte [“Gestores de colas”](#) en la página 29 para obtener más información.

## Servicios

Los objetos de *servicio* son una forma de definir programas para que se ejecuten cuando se inicie o se detenga un gestor de colas.

Los programas pueden ser de uno de los tipos siguientes:

### Servidores

Un *servidor* es un objeto de servicio que tiene el parámetro SERVTYPE establecido en SERVER. Un objeto de servicio de servidor es la definición de un programa que se ejecutará cuando se inicie un gestor de colas especificado. Sólo se puede ejecutar una instancia de un proceso de servidor al mismo tiempo. Mientras se está ejecutando, el estado de un proceso de servidor puede supervisarse con el mandato MQSC, DISPLAY SVSTATUS. Normalmente, los objetos de servicio de servidor son definiciones de programas como, por ejemplo, manejadores de mensajes no entregados o supervisores de desencadenantes, sin embargo, los programas que se pueden ejecutar no se limitan a los que se suministran con IBM MQ. Además, un objeto de servicio de servidor se puede definir para que incluya un mandato que se ejecutará cuando el gestor de colas especificado se apague para finalizar el programa.

### Mandatos

Un *mandato* es un objeto de servicio que tiene el parámetro SERVTYPE establecido en COMMAND. Un objeto de servicio de mandato es la definición de un programa que se ejecutará cuando se inicie o se detenga un gestor de colas especificado. Se pueden ejecutar varias instancias de un proceso de mandato al mismo tiempo. Los objetos de servicio de mandato se diferencian de los objetos de servicio de servidor en que, una vez que se haya ejecutado el programa, el gestor de colas no lo supervisará. Normalmente, los objetos de servicio de mandato son definiciones de programas de corta duración que realizan una tarea específica, como por ejemplo iniciar una o más tareas diferentes.

**Importante:**  Los objetos de servicio no están soportados en IBM MQ for z/OS.

Para obtener más información, consulte [Trabajar con servicios](#).

### Clases de almacenamiento



Una clase de almacenamiento correlaciona una o más colas con un conjunto de páginas.

Esto significa que los mensajes de dicha cola se almacenan (están sujetos al almacenamiento intermedio) en ese conjunto de páginas.

Las clases de almacenamiento solamente están soportadas en IBM MQ for z/OS.

Para obtener más información sobre las clases de almacenamiento, consulte [Planificación del entorno de IBM MQ en z/OS](#).

### Objetos de tema

Un *objeto de tema* es un objeto de IBM MQ que permite asignar atributos específicos, no predeterminados a temas.

Un *tema* se define por una aplicación que se publica en, o se suscribe a, una *serie de temas* determinada. Una serie de tema puede especificar una jerarquía de temas separándolos con un carácter de barra inclinada (/). Esto se puede visualizar con un *árbol de temas*. Por ejemplo, si una aplicación se publica en las series de temas /Sport/American Football y /Sport/Soccer, se crea un árbol de temas que tiene un nodo padre Sport y dos hijos, American Football y Soccer.

Los temas heredan sus atributos del primer nodo administrativo padre encontrado en su árbol de temas. Si no hay nodos de tema administrativo en un árbol de temas específico, todos los temas heredan los atributos del objeto de tema base, SYSTEM.BASE.TOPIC.

Puede crear un objeto de tema en cualquier nodo de un árbol de temas especificando la serie de temas de dicho nodo en el atributo TOPICSTR del objeto del tema. También puede definir otros atributos para el nodo de tema administrativo. Para obtener más información sobre estos atributos, consulte [Los mandatos MQSC o Automatización de la administración utilizando mandatos PCF](#). Cada objeto de tema hereda, de forma predeterminada, sus atributos del nodo de tema administrativo padre más cercano.

Los objetos del tema también se pueden utilizar para ocultar todo el árbol de temas de los desarrolladores de aplicación. Si se crea un objeto de tema denominado FOOTBALL . US para el tema / Sport/American Football, una aplicación puede publicar o suscribirse al objeto denominado FOOTBALL . US en lugar de a la serie /Sport/American Football con el mismo resultado.

Si indica un carácter #, +, / ó \* en una serie de tema de un objeto de tema, el carácter se tratará como un carácter normal en la serie y se considerará que forma parte de la serie de temas asociada a un objeto de tema.

Para obtener más información sobre los objetos de temas, consulte [“Mensajería de publicación/suscripción”](#) en la [página 63](#).

### **Conceptos relacionados**

[“Introducción a la colocación de mensajes en colas”](#) en la [página 5](#)

Los productos de IBM MQ permiten que los programas se comuniquen entre sí a través de una red de componentes diferentes (procesadores, sistemas operativos, subsistemas y protocolos de comunicación) utilizando una interfaz de programación de aplicaciones coherente.

### **Referencia relacionada**

[Mandatos MQSC](#)

## **Colas**

Introducción a las colas y los atributos de cola de IBM MQ.

Los mensajes se almacenan en una cola, de modo que si la aplicación que realiza la transferencia está esperando una respuesta a su mensaje, es libre de realizar otros trabajos mientras espera esa respuesta. Las aplicaciones acceden a una cola utilizando la Interfaz de colas de mensajes (MQI), que se describe en [Visión general de la interfaz de colas de mensajes](#).

Para que un mensaje pueda transferirse a una cola, la cola ya se tiene que haber creado. Una cola es propiedad de un gestor de colas, y ese gestor de colas puede poseer muchas colas. Sin embargo, cada cola debe tener un nombre que sea exclusivo dentro de ese gestor de colas.

Una cola se mantiene a través de un gestor de colas. En la mayoría de los casos, cada cola la gestiona físicamente el gestor de colas pero esto no es evidente para un programa de aplicación. Las colas compartidas de IBM MQ for z/OS las puede gestionar cualquier gestor de colas del grupo de compartición de colas.

Para crear una cola, puede utilizar mandatos de IBM MQ (MQSC), mandatos PCF o interfaces específicas de la plataforma. Por ejemplo, las operaciones y los paneles de control de IBM MQ for z/OS son específicos de la plataforma.

Puede crear colas locales para trabajos temporales *dinámicamente* desde la aplicación. Por ejemplo, puede crear colas de *respuestas* (que no son necesarias después de que finalice una aplicación). Para obtener más información, consulte [“Colas dinámicas y de modelo”](#) en la [página 25](#).

Antes de utilizar una cola, debe abrirla, especificando qué desea hacer con ella. Por ejemplo, puede abrir una cola para:

- Examinar mensajes únicamente (no recuperarlos)
- Recuperar mensajes (y compartir el acceso con otros programas o con acceso exclusivo)
- Transferir mensajes a la cola
- Realizar consultas sobre los atributos de la cola
- Establecer los atributos de la cola

Para obtener una lista completa de las opciones que puede especificar cuando se abre una cola, consulte [MQOPEN – Abrir objeto](#).

## **Atributos de colas**

Algunos de los atributos de una cola se especifican cuando la cola está definida, y no se puede cambiar después (por ejemplo, el tipo de la cola). Otros atributos de colas se pueden agrupar en aquellos que se pueden cambiar:

- Mediante el gestor de colas durante el proceso de la cola (por ejemplo, la profundidad actual de una cola)
- Con mandatos únicamente (por ejemplo, el texto descriptivo de la cola)
- Mediante aplicaciones, utilizando la llamada MQSET (por ejemplo, si se permiten operaciones de transferencia en la cola)

Puede averiguar los valores de todos los atributos utilizando la llamada MQINQ.

Los atributos que son comunes a más de un tipo de cola son los siguientes:

**QName**

Nombre de la cola.

**QType**

Tipo de la cola.

**QDesc**

Texto descriptivo de la cola.

**InhibitGet**

Si se permite que los programas obtengan mensajes de la cola. Sin embargo, nunca puede obtener mensajes de colas remotas.

**InhibitPut**

Si se permite que los programas transfieran mensajes a la cola.

**DefPriority**

Prioridad predeterminada de los mensajes transferidos a la cola.

**DefPersistence**

Persistencia predeterminada de los mensajes transferidos a la cola

**Ámbito**

Controla si una entrada para esta cola también existe en un servicio de nombres.

 El atributo **Scope** no está soportado en z/OS.

Para obtener una descripción completa de estos atributos, consulte [Atributos para colas](#).

## Formas de definir las colas

Puede definir colas en IBM MQ utilizando el mandato MQSC DEFINE o el mandato PCF Crear cola. Los mandatos especifican el tipo de cola y sus atributos. Por ejemplo, un objeto de cola local tiene atributos que especifican lo que sucede cuando las aplicaciones hacen referencia a dicha cola en llamadas MQI. Son ejemplos de atributos:

- Si las aplicaciones pueden recuperar mensajes de la cola (GET habilitado).
- Si las aplicaciones pueden transferir mensajes a la cola (PUT habilitado).
- Si el acceso a la cola es exclusivo de una aplicación o lo comparten varias aplicaciones.
- El número máximo de mensajes que pueden almacenarse en la cola al mismo tiempo (profundidad máxima de cola).
- La longitud máxima de los mensajes que se pueden transferir a la cola.

También hay varias interfaces específicas de la plataforma que puede utilizar para definir colas.

### Conceptos relacionados

[“Colas de clúster” en la página 59](#)

Una cola de clúster es una cola que se aloja en un gestor de colas de clúster y que está disponible para otros gestores de colas del clúster.

[“Colas de mensajes no entregados” en la página 50](#)

La cola de mensajes no entregados es la cola a la que se envían los mensajes que no pueden enviarse a su destino correcto. Cada gestor de colas suele tener una cola de mensajes no entregados.

[Automatización de administración utilizando mandatos PCF](#)

[Trabajar con colas en IBM MQ Console](#)

### **Tareas relacionadas**

[Administración de IBM MQ utilizando mandatos MQSC](#)

[Creación y configuración de gestores de colas y objetos con MQ Explorer](#)

 [Gestión de IBM MQ for IBM i utilizando mandatos CL](#)

 [Orígenes desde los que puede emitir mandatos MQSC y PCF en IBM MQ for z/OS](#)

### **Referencia relacionada**

“Comparación entre las colas compartidas y las colas de clúster” en la página 59

Esta información está pensada para ayudarle a comparar las colas compartidas y las colas de clúster y a decidir cuáles pueden ser más adecuadas para su sistema.

### **Información relacionada**

“¿Qué es una cola compartida?” en la página 171

## **Colas locales**

Las colas de transmisión, de iniciación, de mensajes no entregados, predeterminada, de canal y de sucesos son tipos de cola local.

Una cola se conoce en un programa como *local* si es propiedad del gestor de colas al que está conectado el programa. Puede obtener mensajes y transferirlos en las colas locales.

El objeto de definición de cola contiene la información de definición de la cola, así como los mensajes físicos colocados en cola.

Cada gestor de colas puede tener algunas colas locales que utiliza para fines especiales:

### **Colas de transmisión**

Cuando una aplicación envía un mensaje a una cola remota, el gestor de colas local almacena el mensaje en una cola local especial, llamada *cola de transmisión*. Las aplicaciones pueden transferir mensajes directamente a una cola de transmisión aunque también pueden transferirse indirectamente a través de una definición de cola remota.

Cuando un gestor de colas envía mensajes a un gestor de colas remoto, identifica la cola de transmisión mediante la secuencia siguiente:

1. La cola de transmisión nombrada en el atributo XMITQ de la definición local de una cola remota.
2. Una cola de transmisión con el mismo nombre que el gestor de colas remoto. Este es el valor predeterminado en XMITQ de la definición local de una cola remota.
3. La cola de transmisión nombrada en el atributo DEFXMITQ del gestor de colas local.

Un *agente de canal de mensajes* es un programa de canal asociado a la cola de transmisión, y que entrega el mensaje a su siguiente destino. El siguiente destino es el gestor de colas al que está conectado el canal de mensajes. No es, necesariamente, el mismo gestor de colas que el destino final del mensaje. Cuando el mensaje se entregue en su siguiente destino, se suprimirá de la cola de transmisión. Es posible que el mensaje tenga que pasar por muchos gestores de colas en su ruta hacia el destino final. Debe definir una cola de transmisión en cada gestor de colas, junto con la ruta, que cada una contiene mensajes que están esperando a transmitirse en el siguiente destino. Una cola de transmisión normal contiene mensajes para el destino siguiente, aunque los mensajes podrían tener destinos eventuales distintos. Una cola de transmisión de clúster contiene mensajes para varios destinos. El valor de `correlID` de cada mensaje identifica el canal en el que se coloca el mensaje, para transferirlo a su siguiente destino.

En un gestor de colas puede definir varias colas de transmisión. Puede definir varias colas de transmisión para el mismo destino; cada uno se utiliza para una clase diferente de servicio. Por

ejemplo, es posible que desee crear colas de transmisión diferentes para los mensajes pequeños y grandes que se dirijan al mismo destino. De este modo, puede transferir los mensajes utilizando canales de mensajes diferentes, de forma que los mensajes grandes no retrasarán a los más pequeños. Todos los mensajes dirigidos a colas de clúster o a temas de clúster, se colocan en una sola cola de transmisión de clúster `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, de forma predeterminada. Como opción, puede cambiar el valor predeterminado, y separar el tráfico de los mensajes que van a gestores de colas de clúster diferentes en colas de transmisión de clúster diferentes. Si establece el atributo de gestor de colas `DEFCLXQ` en `CHANNEL`, cada canal de clúster emisor crea una cola de transmisión de clúster distinta. Como alternativa, puede definir manualmente las colas de transmisión de clúster que los canales de clúster emisor deben utilizar.

Las colas de transmisión pueden desencadenar un agente de canal de mensajes para enviar mensajes hacia adelante; consulte [Inicio de aplicaciones IBM MQ utilizando desencadenantes](#).

**z/OS** En IBM MQ for z/OS, si se utiliza la transferencia a colas entre grupos, la cola de transmisión recibe servicio de un *agente de transferencia a colas entre grupos*. Se utiliza una cola de transmisión compartida cuando se utiliza la transferencia a colas entre grupos IBM MQ for z/OS.

## Colas de inicio

Una *cola de inicio* es una cola local en la que el gestor de colas coloca un mensaje desencadenante cuando se produce un suceso desencadenante en una cola de aplicación.

Un suceso desencadenante es un suceso diseñado para provocar que un programa empiece a procesar una cola. Por ejemplo, un suceso puede ser que lleguen más de 10 mensajes. Para obtener más información sobre cómo funciona el desencadenante, consulte [Inicio de aplicaciones IBM MQ utilizando desencadenantes](#).

## Cola de mensajes no entregados (mensaje no entregado)

Una *cola de mensajes no entregados (mensaje no entregado)* es una cola local en la que el gestor de colas coloca mensajes que no puede entregar.

Cuando el gestor de colas coloca un mensaje en la cola de mensajes no entregados, añade una cabecera al mensaje. La información de cabecera incluye el motivo por el cual el gestor de colas ha colocado el mensaje en la cola de mensajes no entregados. También contiene el destino del mensaje original, la fecha y la hora en que el gestor de colas haya colocado el mensaje en la cola de mensajes no entregados.

Las aplicaciones también pueden utilizar la cola para mensajes que no pueden entregar. Para obtener más información, consulte [Utilización de la cola de mensajes no entregados](#).

## Cola de mandatos del sistema

La *cola de mandatos del sistema* es una cola a la que aplicaciones debidamente autorizadas pueden enviar mandatos de IBM MQ. Estas colas reciben los mandatos `PCF`, `MQSC` y `CL`, como soporte en su plataforma, en preparación para que el gestor de colas los ejecute.

**z/OS** En IBM MQ for z/OS, la cola se denomina `SYSTEM.COMMAND.INPUT`; en otras plataformas se denomina `SYSTEM.ADMIN.COMMAND.QUEUE`. Los mandatos aceptados varían según la plataforma. Para obtener más detalles, consulte [Referencia de formatos de comandos programables](#).

## Colas predeterminadas del sistema

Las *colas predeterminadas del sistema* contienen las definiciones iniciales de las colas para el sistema. Cuando se crea una definición de cola, el gestor de colas copia la definición de la cola predeterminada del sistema apropiado. El proceso de crear una definición de cola es diferente del de crear una cola dinámica. La definición de la cola dinámica se basa en la cola de modelo que elija como plantilla para la cola dinámica.

## Colas de sucesos

*Colas de sucesos* contienen mensajes de sucesos. Estos mensajes los notifica el gestor de colas o un canal.

## Colas remotas

Para un programa, una cola es *remota* si pertenece a un gestor de colas diferentes de aquel al que está conectado el programa.

Cuando se ha establecido un enlace de comunicaciones, un programa puede enviar un mensaje a una cola remota. Un programa nunca podrá obtener un mensaje de una cola remota.

El objeto de definición de cola, creado al definir una cola remota sólo guarda la información necesaria para que el gestor de colas localice la cola al que desea que vaya el mensaje. Este objeto se conoce como la *definición local de una cola remota*. Todos los atributos de la cola remota se guardan en el gestor de colas que lo posee, porque es una cola local para ese gestor de colas.

Cuando se abre una cola remota, para identificar la cola debe especificar uno de los siguientes:

- El nombre de la definición local que define la cola remota. Desde el punto de vista de una aplicación, es lo mismo que abrir una cola local. Una aplicación no necesita saber si una cola es local o remota.

Para crear una definición local de una cola remota en todas las plataformas excepto IBM i, utilice el mandato `DEFINE QREMOTE`.

 En IBM i, utilice el mandato `CRTMQMQ`.

- El nombre del gestor de colas remoto y el nombre de la cola tal como se le conoce para dicho gestor de colas remoto.

Las definiciones locales de colas remotas tienen tres atributos además de los atributos comunes que se describen en “Atributos de colas” en la [página 20](#). Estos tres atributos son:

### RemoteQName

El nombre por el que se conoce en el gestor de colas propietario de la cola.

### RemoteQMgrName

El nombre del gestor de colas propietario.

### XmitQName

El nombre de la cola de transmisión local que se utiliza cuando se reenvían mensajes a otros gestores de colas.

Para obtener más información sobre estos atributos, consulte [Atributos para colas](#).

Si utiliza la llamada `MQINQ` para la definición local de una cola remota, el gestor de colas devolverá los atributos de la definición local únicamente, que es el nombre de cola remota, el nombre del gestor de colas remoto y el nombre de la cola de transmisión, no los atributos de la cola local que coinciden en el sistema remoto.

Véase también [Colas de transmisión](#).

## Colas alias

Un *alias de cola* es un objeto de IBM MQ que se puede utilizar para acceder a otra cola o a un tema. Esto significa que puede haber más de un programa que trabaje con la misma cola, accediendo a ella con nombres diferentes.

La cola resultante de la resolución de un nombre de alias, que se conoce como la cola base, puede ser cualquiera de los tipos de colas siguientes, soportados por la plataforma:

- Una cola local
- La definición local de una cola remota.
-  Una cola compartida, que es un tipo de cola local solo disponible en IBM MQ for z/OS.
- Una cola predefinida
- Una cola dinámica

Un nombre de alias también se puede resolver en un tema. Si actualmente una aplicación transfiere mensajes a una cola, se puede conseguir que se publique en un tema convirtiendo el nombre de la cola en un alias para el tema. No es necesario ningún cambio en el código de aplicación.

**Nota:** Un alias no se puede resolver directamente en otro alias en el mismo gestor de colas.

Un ejemplo del uso de las colas alias es que un administrador del sistema otorgue diferentes autorizaciones de acceso al nombre de cola base (es decir, la cola en la que se resuelve el alias) y al nombre de cola alias. Esto significa que un programa o usuario pueden estar autorizados a utilizar la cola alias, pero no la cola base.

Como alternativa, se puede establecer una autorización para inhibir las operaciones de transferir (put) para el nombre de alias, pero permitir las para la cola base.

En algunas aplicaciones, el uso de las colas alias significa que los administradores del sistema pueden cambiar fácilmente la definición de un objeto de cola alias sin tener que modificar la aplicación.

IBM MQ realiza comprobaciones de autorización con el nombre de alias cuando los programas intentan utilizar ese nombre. No comprueba si el programa está autorizado a acceder al nombre en el que se resuelve el alias. Por consiguiente, un programa puede estar autorizado a acceder a un nombre de cola alias, pero no el nombre de cola resuelto.

Además de los atributos de cola generales que se describen en [“Colas” en la página 20](#), las colas de alias tienen un atributo **BaseQName**. Es el nombre de la cola base en el que se resuelve el nombre de alias. Para obtener una descripción más completa de este atributo, consulte [BaseQName \(MQCHAR48\)](#).

Los atributos *InhibitGet* y **InhibitPut** (consulte [“Colas” en la página 20](#)) de las colas alias pertenecen al nombre de alias. Por ejemplo, si el nombre de colas alias ALIAS1 se resuelve en el nombre de la cola base BASE, las inhibiciones ALIAS1 sólo afectan a ALIAS1 y no se inhibe BASE. Sin embargo, las inhibiciones en BASE también afectan a ALIAS1.

Los atributos *DefPriority* y **DefPersistence** también pertenecen al nombre de alias. Por lo tanto, puede por ejemplo asignar diferentes propiedades a diferentes alias de la misma cola base. Además, puede cambiar estas propiedades sin tener que cambiar las aplicaciones que utilizan los alias.

### **Colas dinámicas y de modelo**

Esta información ofrece una perspectiva de las colas dinámicas, las propiedades de las colas dinámicas temporales y permanentes, los usos de las colas dinámicas y algunas consideraciones que hay que tomar en cuenta cuando se utilizan colas dinámicas, y colas modelo.

Cuando un programa de aplicación emite una llamada MQOPEN para abrir una cola modelo, el gestor de colas crea dinámicamente una instancia de una cola local con los mismos atributos que la cola modelo. Según el valor del campo *DefinitionType* de la cola modelo, el gestor de colas crea una cola dinámica temporal o permanente (consulte [Creación de colas dinámicas](#)).

#### **Propiedades de las colas dinámicas temporales**

Las *colas dinámicas temporales* tienen las propiedades siguientes:

-  No pueden ser colas compartidas, accesibles desde los gestores de colas de un grupo de compartición de colas.

Tenga en cuenta que los grupos de compartición de colas solo están disponibles en IBM MQ for z/OS.

- Sólo contienen mensajes persistentes
- Son irrecuperables.
- Se suprimen cuando se inicia el gestor de colas.
- Se suprimen cuando la aplicación que ha emitido la llamada MQOPEN que creó la cola cierra la cola o termina.
  - Si hay mensajes confirmados en la cola, se suprimen.

- Si hay alguna llamada no confirmada MQGET, MQPUT o MQPUT1 para la cola en este momento, la cola se marca como suprimida lógicamente, y sólo se suprimirá físicamente (después de que estas llamadas se hayan confirmado) como parte del proceso de cierre, o cuando la aplicación finalice.
- Si la cola está en uso en este momento (por la aplicación que la crea o cualquier otra), la cola se marca como suprimida lógicamente, y sólo se suprimirá físicamente cuando se cierre la última aplicación que utiliza la cola.
- Los intentos de acceder a una cola suprimida lógicamente (que no se para cerrarla) fallarán con el código de razón MQRC\_Q\_DELETED.
- MQCO\_NONE, MQCO\_DELETE y MQCO\_DELETE\_PURGE se tratan todos como MQCO\_NONE cuando se han especificado en una llamada QCLOSE para la llamada MQOPEN correspondiente que ha creado la cola.

### Propiedades de las colas dinámicas permanentes

Las *colas dinámicas permanentes* tienen las propiedades siguientes:

- Contienen mensajes persistentes o no persistentes.
- Son recuperables en caso de anomalías del sistema.
- Se suprimen cuando una aplicación (no necesariamente la que ha emitido la llamada MQOPEN que ha creado la cola) cierre satisfactoriamente la cola utilizando la opción MQCO\_DELETE o MQCO\_DELETE\_PURGE.
  - Una petición de cierre con la opción MQCO\_DELETE falla si todavía hay algún mensaje (confirmado o sin confirmar) en la cola. Una petición de cierre con la opción MQCO\_DELETE\_PURGE será satisfactoria aunque haya mensajes confirmados en la cola (los mensajes que están suprimiéndose como parte del cierre), pero no será satisfactoria si hay alguna llamada no confirmada MQGET, MQPUT o MQPUT1 pendiente para la cola.
  - Si la petición de supresión es satisfactoria, pero la cola está en uso (por la aplicación que la crea o cualquier otra), la cola se marca como suprimida lógicamente y sólo se suprimirá físicamente cuando se cierre la última aplicación que utiliza la cola.
- No se suprimen si las cierra una aplicación que no está autorizada a suprimir la cola, a menos que la aplicación de cierre emitiera la llamada MQOPEN que creó la cola. Se llevan a cabo comprobaciones de autorización con respecto al identificador de usuario (o un identificador de usuario alternativo si se especificó MQOO\_ALTERNATE\_USER\_AUTHORITY) que se ha utilizado para validar la llamada MQOPEN correspondiente.
- Se pueden suprimir de la misma manera que una cola normal.

### Usos de las colas dinámicas

Puede utilizar colas dinámicas para:

- Aplicaciones que no requieren que se guarden colas después de que la aplicación haya finalizado.
- Aplicaciones que requieren que las respuestas a los mensajes para ser procesados por otra aplicación. Estas aplicaciones pueden crear dinámicamente una cola de respuestas abriendo una cola modelo. Por ejemplo, una aplicación de cliente puede:
  1. Crear una cola dinámica.
  2. Suministrar el nombre en el campo **ReplyToQ** de la estructura de descriptor de mensaje del mensaje de solicitud.
  3. Colocar la petición en una cola que un servidor está procesando.

A continuación, el servidor puede colocar el mensaje de respuesta en la cola de respuestas. Finalmente, el cliente puede procesar la respuesta, y cerrar la cola de respuestas con la opción de supresión.

### Consideraciones al utilizar colas dinámicas

Tenga en cuenta las cuestiones siguientes al utilizar colas dinámicas:

- En un modelo cliente-servidor, cada cliente debe crear y utilizar su propia dinámica cola de respuestas. Si una cola dinámica de respuestas se comparte entre más de un cliente, la supresión de la cola de respuestas podría demorarse debido a que hay actividad no confirmada en la cola o porque otro cliente está utilizando la cola. Además, la cola puede marcarse como suprimida lógicamente e inaccesible para las peticiones posteriores de la API (distintas de MQCLOSE).
- Si el entorno de aplicación requiere que se compartan colas dinámicas entre aplicaciones, asegúrese de que la cola sólo se cierra (con la opción de supresión) cuando toda la actividad que hay en ella se haya confirmado. Esta acción la debe realizar el último usuario. De este modo se asegura que la supresión de la cola no se retrasa, y se minimiza el período en el que la cola es inaccesible porque se ha marcado como suprimida lógicamente.

## Colas modelo

Una *cola modelo* es una plantilla de definición de cola que se utiliza al crear una cola dinámica.

Puede crear una cola local dinámicamente desde un programa IBM MQ, nombrando la cola modelo que desea utilizar como plantilla para los atributos de cola. En ese momento puede cambiar algunos atributos de la nueva cola. Pero no puede cambiar **DefinitionType**. Si, por ejemplo, necesita una cola persistente, seleccione una cola de modelo con el tipo de definición establecido en permanente. Algunas aplicaciones conversacionales pueden utilizar colas dinámicas para albergar las respuestas a sus consultas porque probablemente no necesitan mantener estas colas después de que hayan procesado las respuestas.

Debe especificar el nombre de una cola modelo en el *descriptor de objeto* (MQOD) de la llamada MQOPEN. Utilizando los atributos de la cola modelo, el gestor de colas crea de forma automática y dinámica una cola local.

El usuario puede especificar un nombre (completo) para la cola dinámica o la raíz de un nombre (por ejemplo, ABC) y dejar que el gestor de colas le añada una parte exclusiva o bien dejar que el gestor de colas asigne automáticamente un nombre exclusivo completo. Si el gestor de colas asigna el nombre, lo coloca en la estructura MQOD.

No puede emitir una llamada MQPUT1 directamente a una cola modelo, pero puede emitir una llamada MQPUT1 a la cola dinámica que se ha creado abriendo una cola modelo.

MQSET y MQINQ no se pueden emitir contra una cola de modelo. La apertura de una cola de modelo con MQOO\_INQUIRE o MQOO\_SET, resulta en posteriores llamadas de MQINQ y MQSET realizadas contra la cola creada dinámicamente.

Los atributos de una cola modelo son un subconjunto de los de una cola local. Para obtener una descripción más completa, consulte [Atributos para colas](#).

## Colas utilizadas para fines específicos por IBM MQ

IBM MQ utiliza algunas colas locales para fines específicos relacionados con su funcionamiento.

Debe definir estas colas para que IBM MQ las pueda utilizar.

### Colas de inicio

Las colas de inicio son colas que se utilizan para la activación. Un gestor de colas coloca un mensaje desencadenante en una cola de inicio cuando se produce un suceso desencadenante. Un suceso desencadenante es una combinación lógica de condiciones detectada por un gestor de colas. Por ejemplo, puede generarse un suceso desencadenante cuando el número de mensajes de una cola alcanza una profundidad predefinida. Este suceso hace que el gestor de colas transfiera un mensaje desencadenante a una cola de inicio especificada. Este mensaje desencadenante lo recupera un *supervisor desencadenante*, una aplicación especial que supervisa una cola de inicio. El supervisor desencadenante inicia entonces el programa de aplicación que se haya especificado en el mensaje desencadenante.

Si un gestor de colas ha de utilizar el mecanismo de activación, debe definirse al menos una cola de inicio para ese gestor de colas. Consulte [Gestión de objetos para el mecanismo de activación](#), [runmqtrm](#) e [Inicio de aplicaciones IBM MQ utilizando desencadenantes](#)

## Colas de transmisión

Las colas de transmisión son colas que almacenan temporalmente mensajes destinados a un gestor de colas remoto. Debe definir al menos una cola de transmisión para cada gestor de colas remoto al que el gestor de colas local va a enviar mensajes directamente. Estas colas también se utilizan en la administración remota; consulte [Administración remota desde un gestor de colas local](#). Para obtener información sobre el uso de colas de transmisión en colas distribuidas, consulte [Técnicas de gestión de colas distribuidas de IBM MQ](#).

Cada gestor de colas puede tener una cola de transmisión predeterminada.

- Si un gestor de colas que no forme parte de un clúster transfiere un mensaje a una cola remota, la acción predeterminada es utilizar la cola de transmisión predeterminada.
- Si existe una cola de transmisión que tenga el mismo nombre que el gestor de colas de destino, el mensaje se coloca en dicha cola de transmisión .
- Si existe una definición de alias de gestor de colas, en la que el parámetro **RQMNAME** coincida con el gestor de colas de destino, y se especifica el parámetro **XMITQ**, el mensaje se coloca en la cola de transmisión a la que **XMITQ** ponga nombre.
- Si no existe el parámetro **XMITQ**, el mensaje se coloca en la cola local que se indique en el mensaje.
- Si no hay ninguna, y el mensaje está destinado a un gestor de colas remoto, IBM MQ utiliza la cola de transmisión por defecto, que es **SYSTEM.DEFAULT.XMIT.QUEUE**.

## Colas de transmisión de clúster

Cada gestor de colas de un clúster tiene una cola de transmisión de clúster denominada **SYSTEM.CLUSTER.TRANSMIT.QUEUE**, y una cola de transmisión de clúster modelo, **SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE**. Las definiciones de estas colas se crean, de forma predeterminada, cuando se define un gestor de colas. Si el atributo **DEFCLXQ** del gestor de colas se establece en **CHANNEL**, se crea, de forma automática, una cola de transmisión de clúster dinámico permanente para cada canal de clúster emisor que se cree. Las colas se denominan **SYSTEM.CLUSTER.TRANSMIT.ChannelName**. También puede definir las colas de transmisión de clúster de clúster manualmente.

Un gestor de colas que forme parte del clúster envía los mensajes de una de estas colas a otros gestores de colas que pertenezcan al mismo clúster.

Durante la resolución de nombres, una cola de transmisión de clúster tiene prioridad sobre la cola de transmisión predeterminada, y una cola de transmisión de clúster específica tiene prioridad sobre **SYSTEM.CLUSTER.TRANSMIT.QUEUE**.

## Colas de mensajes no entregados

Una cola de mensajes no entregados es una cola que almacena mensajes que no pueden direccionarse a sus destinos correctos. Un mensaje no se puede direccionar cuando, por ejemplo, la cola de destino está llena. La cola de mensajes no entregados que se suministra con MQSeries se llama **SYSTEM.DEAD.LETTER.QUEUE**.

Para las colas distribuidas, debe definir una cola de mensajes no entregados en cada gestor de colas implicado.

## Colas de mandatos

La cola de mandatos, **SYSTEM.ADMIN.COMMAND.QUEUE**, es una cola local a la que las aplicaciones debidamente autorizadas pueden enviar mandatos MQSC para procesarlos. A continuación, estos mandatos los recupera un componente IBM MQ denominado servidor de mandatos. El servidor de mandatos valida los mandatos, pasa los que son válidos al gestor de colas para que los procese y devuelve las respuestas a la cola de respuestas apropiada.

Al crear un gestor de colas, se crea automáticamente una cola de mandatos para el mismo.

## Colas de respuestas

Cuando una aplicación envía un mensaje de solicitud, la aplicación que lo recibe puede devolver un mensaje de respuesta a la aplicación emisora. Este mensaje se transfiere a una cola, llamada cola de respuestas, que suele ser una cola local de la aplicación emisora. El nombre de la cola de respuestas lo especifica la aplicación emisora como parte del descriptor del mensaje.

## Colas de sucesos

Los sucesos de instrumentación, que pueden utilizarse para supervisar gestores de colas, independientemente de las aplicaciones MQI.

Cuando se produce un suceso de instrumentación, el gestor de colas coloca un mensaje de suceso en una cola de sucesos. Este mensaje puede leerlo entonces una aplicación de supervisión, que puede informar a un administrador o iniciar alguna acción correctora si el suceso indica un problema.

**Nota:** Los sucesos desencadenantes son diferentes de los sucesos de instrumentación. Los sucesos desencadenantes no los causan las mismas condiciones, y no generan mensajes de sucesos.

Para obtener más información sobre los sucesos de instrumentación, consulte [Sucesos de instrumentación](#).

## Gestores de colas

Una introducción a los *gestores de colas* y a los servicios en colas que proporcionan a las aplicaciones.

Un programa debe tener una conexión con un gestor de colas para poder utilizar los servicios de ese gestor de colas. Un programa puede realizar esta conexión explícitamente (utilizando la llamada MQCONN o MQCONNX), o la conexión se puede realizar implícitamente (este depende de la plataforma y el entorno en el que se ejecuta el programa).

Un gestor de colas de IBM MQ garantiza las acciones siguientes:

- Los atributos de objeto se modifican conforme a los mandatos recibidos.
- Los sucesos especiales, tales como sucesos desencadenantes o sucesos de instrumentación, se generan cuando se cumplen las condiciones adecuadas.
- Los mensajes se colocan en la cola correcta, según lo solicitado por la aplicación que ha efectuado la llamada MQPUT. En caso de que esto no sea posible, se notifica a la aplicación y se emite un código de razón apropiado.

Cada cola pertenece a un solo gestor de colas y se dice que es una *cola local* respecto a ese gestor de colas. El gestor de colas al que está conectada una aplicación es el *gestor de colas local* respecto a esa aplicación. Para la aplicación, las colas que pertenecen a su gestor de colas local son colas locales.

Una *cola remota* es una cola que pertenece a otro gestor de colas. Un *gestor de colas remoto* es cualquier gestor de colas que no sea el gestor de colas local. Un gestor de colas remoto puede estar en una máquina remota de la red o puede estar en la misma máquina que el gestor de colas local. IBM MQ ofrece soporte a varios gestores de colas en la misma máquina.

En algunas llamadas MQI, se puede utilizar un objeto gestor de colas. Por ejemplo, puede efectuar consultas sobre los atributos del objeto gestor de colas utilizando la llamada MQI MQINQ.

## Atributos de los gestores de colas

Asociados a cada gestor de colas hay un conjunto de atributos (o propiedades) que definen sus características. Algunos de los atributos de un gestor de colas se arreglan cuando éste se crea; puede cambiar otros utilizando los mandatos de IBM MQ. Puede realizar consultas sobre los valores de todos los atributos, excepto los que se utiliza para el cifrado de TLS (Transport Layer Security), mediante la llamada MQINQ.

Los atributos fijos incluyen los siguientes:

- El nombre del gestor de colas
- La plataforma en la que se ejecuta el gestor de colas (por ejemplo, Windows)
- El nivel de los mandatos de control del sistema al que el gestor de colas da soporte
- La prioridad máxima que se puede asignar a los mensajes procesados por el gestor de colas
- El nombre de la cola a la que los programas pueden enviar mandatos de IBM MQ
- La longitud máxima de mensajes que el gestor de colas puede procesar  (solo arreglado en IBM MQ for z/OS)

- Si el gestor de colas ofrece soporte al recurso de puntos de sincronismo cuando los programas transfieren y obtienen mensajes

Los atributos *modificables* incluyen los siguientes:

- Una descripción de texto del gestor de colas
- El identificador del juego de caracteres que el gestor de colas utiliza para las series de caracteres cuando procesa llamadas MQI
- El intervalo de tiempo que el gestor de colas utiliza para restringir el número de mensajes desencadenantes
-  El intervalo de tiempo que el gestor de colas utiliza para determinar la frecuencia con la que se deben explorar en las colas los mensajes que han caducado (solo IBM MQ for z/OS)
- El nombre de la cola de mensajes no entregados del gestor de colas (mensaje no entregado)
- El nombre de la cola de transmisión predeterminada del gestor de colas
- El número máximo de manejadores abiertos para cualquier conexión individual
- La habilitación e inhabilitación de diversas categorías de informes de sucesos
- El número máximo de mensajes sin confirmar dentro de una unidad de trabajo

## Gestores de colas y gestión de carga de trabajo

Puede configurar un clúster de gestores de colas que tenga más de una definición para la misma cola (por ejemplo, los gestores de colas en el clúster podrían ser clones de entre sí). Los mensajes de una cola determinada los puede manejar cualquier gestor de colas que aloja una instancia de la cola. Un algoritmo de gestión de carga de trabajo decide qué gestor de colas maneja el mensaje y reparte la carga de trabajo entre los gestores de colas; consulte [El algoritmo de gestión de carga de trabajo de clúster](#) para obtener más información.

## Canales

Un *canal* es un enlace de comunicación lógico, utilizado por gestores de colas distribuidos, entre un servidor de IBM MQ MQI client y un servidor de IBM MQ o entre dos servidores de IBM MQ.

Los canales se utilizan para mover los mensajes desde un gestor de colas a otro y protegen las aplicaciones de los protocolos de comunicaciones subyacentes. Los gestores de colas pueden estar en sistemas diferentes de la misma plataforma o de plataformas distintas. Los mensajes que se envían pueden tener su origen en muchas ubicaciones:

- Programas de aplicación escritos por el usuario que transfieren datos de un nodo a otro.
- Aplicaciones de administración escritas por usuario que utilizan mandatos PCF o MQAI.
- IBM MQ Explorer.
- Gestores de colas que envían mensajes de sucesos de instrumentación a otro gestor de colas.
- Gestores de colas que envían mandatos de administración remotos a otro gestor de colas. Por ejemplo, utilizando los mandatos MQSC o la administrative REST API.

Un canal tiene dos definiciones: una en cada extremo de la conexión. Para que los gestores de colas se comuniquen entre sí, debe definir un objeto canal en el gestor de colas que va a enviar los mensajes y otro complementario en el gestor de colas que va a recibirlos. Debe utilizarse el mismo *nombre de canal* en cada extremo de la conexión y el *tipo de canal* utilizado debe ser compatible.

Hay tres categorías de canal en IBM MQ, con diferentes tipos de canal dentro de estas categorías:

- Canales de mensaje, que son unidireccionales y transfieren mensajes de un gestor de colas a otro.
- Canales MQI, que son bidireccionales y transfieren llamadas MQI de un IBM MQ MQI client a un gestor de colas y respuestas de un gestor de colas a un cliente de IBM MQ.

- Canales AMQP, que son bidireccionales y conectan un cliente AMQP a un gestor de colas en una máquina servidor. IBM MQ utiliza canales AMQP para transferir llamadas y respuestas AMQP entre aplicaciones AMQP y gestores de colas

## Canales de mensajes

El objetivo de un canal de mensajes es transferir mensajes de un gestor de colas a otro. Los canales de mensajes no son necesarios para el entorno de cliente-servidor.

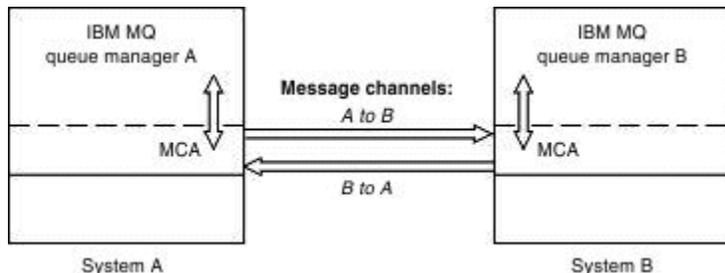


Figura 2. Canales de mensajes entre dos gestores de colas

Un canal de mensajes es un enlace unidireccional. Si desea que un gestor de colas remoto responda a los mensajes enviados por un gestor local, debe configurar un segundo canal para que devuelva las respuestas al gestor de colas local.

Un canal de mensajes conecta dos gestores de colas utilizando los *agentes de canal de mensajes* (MCA). Hay un agente de canal de mensajes en cada extremo de un canal. Puede permitir que MCA transfiera los mensajes utilizando varias hebras. Este proceso se conoce como *canalización*. El proceso de canalización permite que MCA transfiera los mensajes de forma más eficaz y mejora el rendimiento de los canales. Para obtener más información sobre la interconexión, consulte [Atributos de los canales](#).

Para obtener más información sobre los canales, consulte la sección [Llamadas de salida de canal y estructuras de datos](#) y [“Componentes de la gestión de colas distribuidas”](#) en la página 47.

## Canales MQI

Un canal de interfaz de cola de mensajes (MQI) conecta un IBM MQ MQI client a un gestor de colas en una máquina servidor y se establece cuando se emite una llamada MQCONN o MQCONNX desde una aplicación de IBM MQ MQI client.

Es un enlace bidireccional y se utiliza para la transferencia de llamadas MQI y respuestas únicamente, incluyendo llamadas MQPUT que contienen datos de mensajes y llamadas MQGET que hacen que se devuelvan los datos de mensajes. Hay diferentes formas de crear y utilizar definiciones de canal (consulte [Definición de canales MQI](#)).

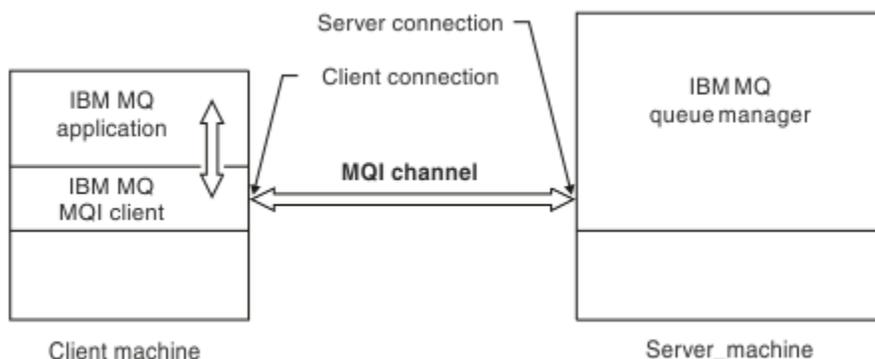


Figura 3. Conexión con el cliente y con el servidor en un canal MQI

 Un canal MQI se puede utilizar para conectar un cliente con un único gestor de colas o con un gestor de colas que forma parte de un grupo de compartición de colas (consulte [Conexión de un cliente a un grupo de compartición de colas](#)).

Hay dos tipos de canal para definiciones de canal MQI. Definen el canal MQI bidireccional.

### **Canal de conexión con el cliente**

Este tipo es para IBM MQ MQI client.

### **Canal de conexión con el servidor**

Este tipo es para el servidor que ejecuta el gestor de colas, con el que la aplicación de IBM MQ, que se ejecuta en un entorno de IBM MQ MQI client, se va a comunicar.

## **Canales AMQP**



Sólo hay un tipo de canal AMQP.

Utilice el canal para conectar una aplicación de mensajería AMQP con un gestor de colas, lo cual permite que la aplicación intercambie mensajes con aplicaciones IBM MQ. Un canal AMQP le permite desarrollar una aplicación utilizando MQ Light y desplegarlo después como una aplicación de empresa, beneficiándose de los recursos a nivel de empresa que proporciona IBM MQ.

## **Canales de conexión de cliente**

Los *canales de conexión de cliente* son objetos que proporcionan una vía de comunicación de un IBM MQ MQI client a un gestor de colas.

Los canales de conexión de cliente se utilizan en la gestión de colas distribuidas para trasladar mensajes entre un gestor de colas y un cliente. Permiten que las aplicaciones no tengan que preocuparse por los protocolos de comunicaciones subyacentes. El cliente puede existir en la misma plataforma que el gestor de colas o en una diferente.

## **Definiciones de canal**

Consulte [“Definiciones de canal”](#) en la página 32 si desea obtener descripciones de cada tipo de canal.

### **Conceptos relacionados**

[“Gestión de colas distribuidas y clústeres”](#) en la página 43

La gestión de colas distribuidas hace referencia al envío de mensajes desde un gestor de colas a otro. El gestor de colas receptor puede estar en la misma máquina o en otra; puede estar próximo o en el otro lado del mundo. Puede ejecutarse en la misma plataforma que el gestor de colas local o puede estar en cualquiera de las plataformas soportadas por IBM MQ. Puede definir manualmente todas las conexiones en un entorno de colas distribuidas o puede crear un clúster y dejar que IBM MQ defina automáticamente gran parte del detalle de conexión.

[Visión general de la Interfaz de cola de mensajes](#)

### **Tareas relacionadas**

[Administración de objetos de IBM MQ remotos](#)

[Detención de canales MQI](#)

[Configurar conexiones entre el servidor y el cliente](#)

### **Referencia relacionada**

[Llamadas de salida de canal y estructuras de datos](#)

[“Comunicaciones”](#) en la página 36

IBM MQ MQI clients utiliza canales MQI para comunicarse con el servidor.

## **Definiciones de canal**

Tablas que describen los distintos tipos de canales de mensajes y canales MQI que IBM MQ utiliza.

Cuando se hace referencia a canales de mensajes, la palabra canal se utiliza a menudo como sinónimo de definición de canal. Normalmente, el contexto permite discernir si se habla de un canal completo, con dos extremos, o de una definición de canal, con un único extremo.

## Canales de mensajes

El canal de mensajes puede ser de los siguientes tipos:

Tipo de definición de canal de mensajes	Descripción
Emisor	Un canal emisor es un canal de mensajes que utiliza el gestor de colas para enviar mensajes a otros gestores. Para enviar mensajes utilizando un canal emisor, debe crear también un canal receptor en el otro gestor de colas con el mismo nombre que el canal emisor. También puede utilizar canales emisores con canales peticionarios si implementa un mecanismo de "devolución de llamada".
Servidor	Un canal servidor es un canal de mensajes que el gestor de colas utiliza para enviar mensajes a los otros gestores. Para enviar mensajes utilizando un canal emisor, debe crear también un canal receptor en el otro gestor de colas con el mismo nombre que el canal servidor. También puede utilizar los canales servidores con canales peticionarios. En ese caso, la definición de canal peticionario en el otro extremo del canal solicita que se inicie la definición de canal servidor. El servidor envía mensajes al peticionario. El servidor también puede iniciar la comunicación, siempre y cuando sepa el nombre de conexión del canal asociado.
Receptor	Un canal receptor es un canal de mensajes que utiliza el gestor de colas para recibir mensajes de otros gestores de colas. Para recibir mensajes utilizando un canal receptor, debe crear también un canal emisor o un canal servidor en el otro gestor de colas con el mismo nombre que el canal receptor.
Peticionario	Un canal peticionario es un canal de mensajes que utiliza el gestor de colas para recibir mensajes de otros gestores de colas. Un canal peticionario puede solicitar que se inicie el canal de socio definido en el extremo remoto. Si el canal de socio es un canal de servidor, el canal de servidor acepta la solicitud de inicio y empieza a enviar mensajes, desde la cola de transmisión identificada en la definición del canal de servidor, al canal de solicitante. Si el canal de socio es un canal emisor, el canal emisor acepta la solicitud de inicio pero, a continuación, cierra la conexión con el solicitante. A continuación, el canal emisor se inicia, negocia una sesión con el canal solicitante de socio y comienza a enviar mensajes desde la cola de transmisión identificada en la definición del canal emisor. Este último caso proporciona esencialmente un mecanismo de devolución de llamada en el que el canal solicitante solicita al canal emisor que devuelva la llamada.

Tipo de definición de canal de mensajes	Descripción
Clúster emisor	La definición de canal de clúster emisor (CLUSDR) define el extremo emisor de un canal en el que un gestor de colas de clúster puede enviar información de clúster a uno de los depósitos completos. El canal de clúster emisor se utiliza para notificar el depósito de cualquier cambio que se realice en el estado del gestor de colas, por ejemplo la adición o supresión de una cola. Se utiliza también para transmitir mensajes. Los gestores de colas de depósito completo propiamente tienen canales de clúster emisor que se apuntan uno al otro. Se utilizan para comunicarse entre sí los cambios de estado del clúster. No tiene mucha importancia saber qué depósito completo apunta la definición de canal CLUSSDR del gestor de colas. Tras el contacto inicial, se definirán automáticamente más objetos de gestor de colas de clúster, según sea necesario, para que el gestor de colas pueda enviar la información de clúster a cada depósito completo, y los mensajes a cada gestor de colas.
Clúster receptor	La definición de canal de clúster receptor (CLUSRCVR) define el extremo receptor de un canal en el que un gestor de colas de clúster puede recibir mensajes de otros gestores de colas del clúster. Un canal de clúster receptor también puede transportar información sobre el clúster destinada para el repositorio. Al definir el canal de clúster receptor, el gestor de colas indica a los otros gestores de colas de clúster que se encuentra disponible para recibir mensajes. Necesita como mínimo un canal de clúster receptor para gestor de colas de clúster.

Debe definir los dos extremos de todos los canales para tener una definición de canal para cada extremo del canal. Los dos extremos del canal deben ser compatibles.

Puede tener las siguientes combinaciones de definiciones de canal:

- Emisor-Receptor
- Servidor-Receptor
- Peticionario-Servidor
- Peticionario-Servidor (devolución de llamada)
- Emisor del clúster-Receptor del clúster

### Agentes de canal de mensajes

Todas las definiciones de canal que cree pertenecen a un gestor de colas determinado. Un gestor de colas puede tener varios canales del mismo o de diferentes tipos. En cada extremo del canal hay un programa, el agente de canal de mensajes (MCA). En cada extremo del canal, el MCA de llamada coge los mensajes de la cola de transmisión y los envía a través del canal. En el otro extremo del canal, el MCA de respuesta recibe los mensajes y los entrega al gestor de colas remoto.

Un MCA de llamada puede estar asociado con un canal emisor, servidor o petionario. Un MCA de respuesta puede estar asociado con cualquier tipo de canal de mensajes.

IBM MQ admite las siguientes combinaciones de tipos de canal en los dos extremos de una conexión:

De llamada		Dirección del flujo de mensajes	De respuesta	
Tipo de canal	¿Es necesario un escucha?		¿Es necesario un escucha?	Tipo de canal
Emisor	No	De canal de llamada a canal de respuesta	Sí	Receptor
Servidor	No	De canal de llamada a canal de respuesta	Sí	Receptor
Servidor	No	De canal de llamada a canal de respuesta	Sí	Peticionario
Peticionario	No	De canal de respuesta a canal de llamada	Sí	Servidor
Peticionario	Sí	De canal de respuesta a canal de llamada	Sí	Emisor

## Canales MQI

Los canales MQI pueden ser de los siguientes tipos:

Tipo de canal MQI	Descripción
Conexión de servidor	Un canal de conexión con el servidor es un canal MQI bidireccional que se utiliza para conectar un cliente de IBM MQ con un servidor de IBM MQ. El canal de conexión con el servidor es el extremo del servidor del canal.
Conexión de cliente	Un canal de conexión con el cliente es un canal MQI bidireccional que se utiliza para conectar un cliente de IBM MQ con un servidor de IBM MQ. IBM MQ Explorer también utiliza conexiones de cliente para conectarse a gestores de colas remotos. El canal de conexión de cliente es el extremo del cliente del canal. Cuando cree el canal de conexión con el cliente, se creará un archivo en el sistema que alojará el gestor de colas. A continuación, debe copiar el archivo de conexión de cliente en el sistema cliente de IBM MQ.

### Soporte de varias hebras-interconexiones

Opcionalmente, puede permitir que un agente de canal de mensajes (MCA) transfiera mensajes utilizando varias hebras. Este proceso, denominado *canalización*, permite al MCA transferir mensajes de forma más eficiente, con menos estados de espera, lo que mejora el rendimiento del canal. Cada MCA está limitado a un máximo de dos hebras.

Puede controlar la interconexión con el parámetro *PipeLineLength* en el archivo *qm.ini*. Este parámetro se añade a la stanza Channels.

**Nota:** El canalizado solo es efectivo para los canales TCP/IP.

Cuando se utiliza la interconexión, los gestores de colas en ambos extremos del canal deben estar configurados para tener una longitud de *PipeLine* mayor que 1.

## Consideraciones sobre la salida de canal

La interconexión puede hacer que algunos programas de salida fallen, porque:

- Es posible que las salidas no se llamen en serie.
- Las salidas se pueden llamar alternativamente desde diferentes hebras.

Compruebe el diseño de los programas de salida antes de utilizar la interconexión:

- Las salidas deben ser reentrantes en todas las etapas de su ejecución.
- Cuando utilice llamadas MQI, recuerde que no puede utilizar el mismo descriptor de contexto MQI cuando se invoca la salida desde hebras diferentes.

Considere una salida de mensajes que abra una cola y utilice su manejador para llamadas MQPUT en todas las invocaciones posteriores de la salida. Esto falla en modalidad de interconexión porque se llama a la salida desde distintas hebras. Para evitar esta anomalía, mantenga un descriptor de contexto de cola para cada hebra y compruebe el identificador de hebra cada vez que se invoque la salida.

## Comunicaciones

IBM MQ MQI clients utiliza canales MQI para comunicarse con el servidor.

Se debe crear una definición de canal en los extremos de IBM MQ MQI client y del servidor de la conexión. Cómo crear definiciones de canal se explica en [Definición de canales MQI](#).

Los protocolos de transmisión posibles se muestran en la tabla siguiente:

Tabla 1. Protocolos de transmisión para canales MQI				
Plataforma de cliente	LU6.2	TCP/IP	NetBIOS	SPX
 IBM i		Sí		
 Linux  AIX Sistemas AIX and Linux	Sí <sup>1</sup>	Sí		
 Windows	Sí	Sí	Sí	Sí

### Nota:

1.  LU6.2 no está soportado en las plataformas siguientes:

- Linux (plataforma POWER)
- Linux (plataforma x86-64)
- Linux (plataforma zSeries s390x)

En [Protocolos de transmisión - combinación de plataformas IBM MQ MQI client y de servidor](#) se muestran las posibles combinaciones de plataformas IBM MQ MQI client y de servidor, utilizando estos protocolos de transmisión.

Una aplicación de IBM MQ en un IBM MQ MQI client puede utilizar todas las llamadas MQI del mismo modo que cuando el gestor de colas es local. **MQCONN** o **MQCONNX** asocia la aplicación de IBM MQ con el gestor de colas seleccionado, creando un *descriptor de conexión*. A continuación, el gestor de colas conectado procesa otras llamadas que utilizan ese manejador de conexión. La comunicación de IBM MQ MQI client requiere una conexión activa entre el cliente y el servidor, a diferencia de la comunicación entre gestores de colas, que es independiente de la conexión e independiente del tiempo.

El protocolo de transmisión se especifica utilizando la definición de canal y no afecta a la aplicación. Por ejemplo, una aplicación de Windows se puede conectar a un gestor de colas a través de TCP/IP y a otro gestor de colas a través de NetBIOS.

## Consideraciones sobre el rendimiento

El protocolo de transmisión que utilice puede afectar al rendimiento del sistema cliente y servidor de IBM MQ. En determinadas situaciones en las que la transmisión es lenta, puede utilizar la compresión de canal de IBM MQ.

## Denominación de objetos de IBM MQ

El convenio de denominación adoptado para los objetos de IBM MQ depende del objeto. El nombre de las máquinas y los ID de usuario que utiliza con IBM MQ también están sujetos a algunas restricciones de denominación.

Cada instancia de un gestor de colas se reconoce por su nombre. Este nombre debe ser exclusivo dentro de la red de gestores de colas interconectados, de modo que un gestor de colas pueda identificar inequívocamente el gestor de colas de destino al que se envía un determinado mensaje.

En los demás tipos de objetos, cada objeto tiene un nombre asociado con el que puede hacerse referencia al mismo. Estos nombres deben ser exclusivos dentro de un gestor de colas y de un tipo de objeto. Por ejemplo, puede haber una cola y un proceso con el mismo nombre, pero no puede haber dos colas con el mismo nombre.

En IBM MQ, los nombres pueden tener un máximo de 48 caracteres, con la excepción de los *canales* que tienen un máximo de 20 caracteres. Para obtener más información sobre la denominación de objetos IBM MQ, consulte [“Reglas de denominación de objetos de IBM MQ”](#) en la página 37.

El nombre de las máquinas y los ID de usuario que utiliza con IBM MQ también están sujetos a algunas restricciones de denominación:

- Asegúrese de que el nombre de la máquina no contiene espacios. IBM MQ no da soporte a nombres de máquina que incluyan espacios. Si instala IBM MQ en una máquina de este tipo, no puede crear ningún gestor de colas.
- En el caso de las autorizaciones de IBM MQ, los nombres de los ID de usuario y los grupos no deben tener más de 20 caracteres (no se permiten espacios).
-  Un servidor de IBM MQ for Windows no da soporte a la conexión de un IBM MQ MQI client si el cliente se ejecuta con un ID de usuario que contiene el carácter @, por ejemplo, abc@d.

### Conceptos relacionados

[“Nombres de archivo de IBM MQ”](#) en la página 40

Cada gestor de colas de IBM MQ, cola, definición de proceso, lista de nombres, canal, canal de conexión cliente, escucha, servicio y objeto de información de autenticación se representa mediante un archivo. Puesto que los nombres de objetos no son necesariamente nombres de archivo válidos, el gestor de colas convierte el nombre de objeto en un nombre de archivo válido cuando es necesario.

### Referencia relacionada

[“Reglas de denominación de objetos de IBM MQ”](#) en la página 37

Los nombres de objeto de IBM MQ tienen longitudes máximas y distinguen entre mayúsculas y minúsculas. No todos los caracteres están soportados para cada tipo de objeto y muchos objetos tienen reglas relativas a la exclusividad de nombres.

## Reglas de denominación de objetos de IBM MQ

Los nombres de objeto de IBM MQ tienen longitudes máximas y distinguen entre mayúsculas y minúsculas. No todos los caracteres están soportados para cada tipo de objeto y muchos objetos tienen reglas relativas a la exclusividad de nombres.

Hay muchos tipos diferentes de objeto IBM MQ y los objetos de cada tipo pueden tener todos el mismo nombre porque existen en espacios de nombres de objetos separados: por ejemplo, una cola local y un canal emisor pueden tener ambos el mismo nombre. Sin embargo, un objeto no puede tener el mismo nombre que otro objeto en el mismo espacio de nombres: por ejemplo, una cola local no puede tener el mismo nombre que una cola de modelo y un canal emisor no puede tener el mismo nombre que un canal receptor.

Los siguientes objetos de IBM MQ existen en espacios de nombres de objetos separados:

- Información de autenticación
- Canal
- Canal de cliente
- Escucha
- Lista de nombres
- Proceso
- Cola
- Servicio
- Clase de almacenamiento
- Suscripción
- Tema

### Longitud de caracteres de nombres de objeto

En general, los nombres de objeto de IBM MQ pueden tener hasta 48 caracteres de longitud. Esta regla se aplica a los siguientes objetos:

- Información de autenticación
- Clúster
- Escucha
- Lista de nombres
- Definición de proceso
- Cola
- Gestor de colas
- Servicio
- Suscripción
- Tema

Existen restricciones:

1.  En los sistemas z/OS, los gestores de colas deben tener un máximo de 4 caracteres y deben estar en mayúsculas y sólo caracteres numéricos.
2. La longitud máxima de los nombres de objeto de canal y los nombres de canal de conexión con el cliente es 20 caracteres. Consulte [Definición de los canales](#) para obtener más información sobre los canales.
3. Las series de tema pueden tener un máximo de 10240 bytes. Todos los nombres de objeto de IBM MQ distinguen entre mayúsculas y minúsculas.
4. Los nombres de suscripción pueden tener un máximo de 10240 bytes y pueden contener espacios.
5. La longitud máxima de los nombres de clase de almacenamiento es 8 caracteres.
6. La longitud máxima de los nombres de estructura de CF es 12 caracteres .

### Caracteres en nombres de objeto

Los caracteres válidos para los nombres de objeto de IBM MQ son:

Caracteres	Restricciones
De la A a la Z en mayúsculas	• Ninguna

Caracteres	Restricciones
De la a a la z en minúsculas	<ul style="list-style-type: none"> <li>• En los scripts MQSC, los nombres con caracteres en minúsculas deben ir entre comillas simples. Esto impide que los caracteres en minúsculas se convertirán en mayúsculas.</li> <li>• Los sistemas que utilizan EBCDIC Katakana no pueden utilizar los caracteres de a- z en minúsculas en los nombres de objeto.</li> <li>•  Podrían haber restricciones al utilizar caracteres en minúsculas en los sistemas z/OS; por ejemplo, los nombres de gestor de colas no pueden contener caracteres en minúsculas.</li> <li>•  En los sistemas IBM i, cuando se utilizan mandatos CL, los nombres con caracteres en minúsculas deben ir entre comillas simples. Esto impide que los caracteres en minúsculas se convertirán en mayúsculas.</li> </ul>
Numéricos de 0 a 9	<ul style="list-style-type: none"> <li>• Ninguna</li> </ul>
Punto (.)	<ul style="list-style-type: none"> <li>• Ninguna</li> </ul>
Subrayado (_)	<ul style="list-style-type: none"> <li>•  Ninguna</li> <li>•  Evite utilizar nombres con caracteres de subrayado iniciales o finales porque los paneles de operaciones y de control de IBM MQ for z/OS no pueden manejarlos.</li> </ul>
Barra inclinada (/)	<ul style="list-style-type: none"> <li>•  En los sistemas Windows, el primer carácter de un nombre de gestor de colas no puede ser una barra inclinada.</li> <li>•  En los sistemas IBM i, cuando se utilizan mandatos CL, los nombres que contienen una barra inclinada deben ir entre comillas simples.</li> <li>•  Ninguna</li> </ul>
Signo de porcentaje (%)	<ul style="list-style-type: none"> <li>•  Ninguna</li> <li>•  Si utiliza RACF como gestor de seguridad externa para IBM MQ for z/OS, no utilice % en nombres de objeto porque los nombres no están incluidos en las comprobaciones de seguridad cuando se utilizan perfiles genéricos de RACF.</li> <li>•  En los sistemas IBM i, cuando se utilizan mandatos CL, los nombre que contienen un símbolo de porcentaje deben ir entre comillas simples.</li> </ul>

Existen algunas reglas generales en relación con los caracteres en los nombres de objeto:

1. No se permiten los espacios en blanco iniciales o intercalados.
2. No se permiten caracteres de idiomas nacionales.
3. Cualquier nombre que sea menor que la longitud total del campo puede rellenarse por la derecha con espacios en blanco. Todos los nombres abreviados devueltos por el gestor de colas se rellenan siempre con espacios en blanco a la derecha.

## Nombres de cola

El nombre de una cola tiene dos partes:

- El nombre de un gestor de colas
- El nombre local de la cola como es conocido para dicho gestor de colas

Cada parte del nombre de cola tiene 48 caracteres de longitud.

Para hacer referencia a una cola local, puede omitir el nombre del gestor de colas (sustituyéndolo por caracteres en blanco o utilizando un carácter nulo inicial). Sin embargo, todos los nombres de cola devueltos a un programa por IBM MQ contienen el nombre del gestor de colas.

 Una cola compartida, accesible a cualquier gestor de colas en su grupo de compartición de colas, no puede tener el mismo nombre que cualquier cola local no compartida en el mismo grupo de compartición de colas. Esta restricción evita la posibilidad de que una aplicación abra por error una cola compartida cuando se pretendía abrir una cola local, o viceversa. Las colas compartidas y los grupos de compartición de colas solo están disponibles en IBM MQ for z/OS.

Para hacer referencia a una cola remota, un programa debe incluir el nombre del gestor de colas en el nombre de cola completo, o debe haber una definición local de la cola remota.

Cuando una aplicación utiliza un nombre de cola, dicho nombre puede ser el nombre de una cola local (o un alias de una) o el nombre de una definición local de una cola remota, pero la aplicación no necesita saber cual, a menos que deba obtener un mensaje de la cola (cuando la cola debe ser local). Cuando la aplicación abre el objeto de cola, la llamada de MQOPEN realiza una función de resolución de nombre para determinar en qué cola realizar operaciones posteriores. La importancia de esto es que la aplicación no tiene ninguna dependencia incorporada en las colas particulares que se están definiendo en ubicaciones determinadas en una red de gestores de colas. Por lo tanto, si un administrador del sistema reubica las colas en la red y cambia sus definiciones, no es necesario cambiar las aplicaciones que utilizan estas colas.

## Nombres de objeto reservados

Los nombres de objeto que empiezan por SYSTEM. están reservados para los objetos definidos por el gestor de colas. Puede utilizar los mandatos **Alter**, **Define** y **Replace** para cambiar estas definiciones de objeto para adaptarse a la instalación. Los nombres definidos para IBM MQ se listan de forma completa en [Nombres de cola](#).

 En IBM MQ for z/OS, el nombre de estructura de aplicación de recurso de acoplamiento CSQSYSAPPL está reservado.

### Conceptos relacionados

[Nombre de instalación en AIX, Linux, and Windows](#)

## Nombres de archivo de IBM MQ

Cada gestor de colas de IBM MQ, cola, definición de proceso, lista de nombres, canal, canal de conexión cliente, escucha, servicio y objeto de información de autenticación se representa mediante un archivo. Puesto que los nombres de objetos no son necesariamente nombres de archivo válidos, el gestor de colas convierte el nombre de objeto en un nombre de archivo válido cuando es necesario.

La vía de acceso predeterminada a un directorio de gestor de colas es la siguiente:

- Un prefijo, que se define en la información de configuración de IBM MQ:

-  En AIX and Linux, el prefijo predeterminado es /var/mqm. El prefijo se configura en la stanza DefaultPrefix del archivo de configuración mqs.ini.
-  En los sistemas Windows de 32 bits, el prefijo predeterminado es C:\Archivos de programa (x86)\IBM\WebSphere MQ. En los sistemas Windows de 64 bits, el prefijo predeterminado es C:\Archivos de programa\IBM\MQ. Tanto en las instalaciones de 32 bits como de 64 bits, los directorios de datos se instalan en C:\ProgramData\ IBM \ MQ. El prefijo se configura en la stanza DefaultPrefix del archivo de configuración mqs.ini.

Cuando está disponible, el prefijo se puede cambiar utilizando la página de propiedades de IBM MQ en IBM MQ Explorer; de lo contrario, edite el archivo de configuración de mqs . ini manualmente.

- El nombre del gestor de colas se transforma en un nombre de directorio válido. Por ejemplo, el gestor de colas:

```
queue.manager
```

se representaría como:

```
queue!manager
```

Este proceso se denomina *transformación de nombres*.

En IBM MQ, puede asignar a un gestor de colas un nombre que contenga hasta 48 caracteres.

Por ejemplo, podría asignar el siguiente nombre a un gestor de colas:

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

Sin embargo, cada gestor de colas se representa mediante un archivo y existen limitaciones con respecto a la longitud máxima que pueden tener los nombres de archivo y a los caracteres que se pueden utilizar en los nombres. Como resultado, los nombres de archivo que representan objetos se transforman automáticamente para cumplir los requisitos del sistema de archivos.

Las reglas que controlan la transformación de un nombre de gestor de colas son las siguientes:

1. Transformar caracteres individuales:
  - De . a !
  - De / a &
2. Si el nombre sigue sin ser válido:
  - a. Se trunca a ocho caracteres
  - b. Se añade un sufijo numérico de tres caracteres

Por ejemplo, suponiendo el prefijo predeterminado y un gestor de colas con el nombre queue . manager:

-  En Windows con NTFS o FAT32, el nombre del gestor de colas se convierte en:

```
C:\Archivos de programa\IBM\MQ\mqgrs\queue!manager
```

-  En Windows con FAT, el nombre de gestor de colas se convierte en:

```
C:\Archivos de programa\IBM\MQ\mqgrs\queue!ma
```

-  En AIX and Linux, el nombre del gestor de colas pasa a ser:

```
/var/mqm/mqgrs/queue!manager
```

El algoritmo de transformación también permite distinguir entre nombres que sólo difieren en el tipo de letra (mayúsculas/minúsculas) en sistemas de archivos que no son sensibles a mayúsculas y minúsculas.

## Transformación de nombres de objetos

Los nombres de objetos no son necesariamente nombres válidos para los sistemas de archivos. Es posible que tenga que transformar los nombres de objetos. El método utilizado es distinto del empleado para los nombres de gestores de colas porque, aunque sólo hay unos pocos nombres de gestores de colas por cada máquina, puede haber un gran número de otros objetos para cada gestor de colas. En el sistema de archivos se representan colas, definiciones de proceso, listas de nombres, canales, canales de conexión con el cliente, escuchas, servicios y objetos de información de autenticación.

Cuando el proceso de transformación genera un nuevo nombre, no hay ninguna relación simple con el nombre de objeto original. Puede utilizar el mandato **dspmqls** para efectuar conversiones entre nombres de objetos reales y transformados.

### Referencia relacionada

**dspmqls** ([visualizar nombres de archivos](#))

### Información relacionada

Stanza [AllQueueManagers](#) del archivo `mqs.ini`

## IBM i Nombres de objetos en IBM i

Un gestor de colas tiene una biblioteca de gestor de colas asociada que tiene un nombre exclusivo. Es posible que los nombres del gestor de colas y los nombres de objetos deban transformarse para cumplir los requisitos del sistema de archivos integrado (IFS) de IBM i.

Cuando se crea un gestor de colas, IBM MQ asocia una biblioteca del gestor de colas con él. A esta biblioteca del gestor de colas se le da un nombre exclusivo, de no más de 10 caracteres, que se basa principalmente en el nombre del gestor de colas definido por el usuario. Tanto el gestor de colas como la biblioteca del gestor de colas se colocan en un directorio que también se basa en el nombre del gestor de colas con el prefijo `/QIBM/UserData/mqm`. A continuación, se ofrece un ejemplo de un gestor de colas, una biblioteca del gestor de colas y un directorio:

Nombre del gestor de colas	ORANGE
Nombre de la biblioteca del gestor de colas	QMORANGE
Directorio	<code>/QIBM/UserData/mqm/ORANGE</code>

Todos los nombres de gestor de colas y nombres de biblioteca del gestor de colas se graban en stanzas en el archivo `/QIBM/UserData/mqm/mqs.ini`.

## Directorios y archivos IFS de IBM MQ

Integrated File System (IFS) de IBM i utiliza de forma extensiva IBM MQ para almacenar datos. Para obtener más información acerca del IFS consulte la publicación *Integrated File System Introduction*.

Cada objeto de IBM MQ, por ejemplo, un canal o un gestor de colas, se representa mediante un archivo. Puesto que los nombres de objetos no son necesariamente nombres de archivo válidos, el gestor de colas convierte el nombre de objeto en un nombre de archivo válido cuando es necesario.

La vía de acceso a un directorio de gestor de colas se forma a partir de los siguientes elementos:

- Un prefijo, que se define en el archivo de configuración del gestor de colas, `qm.ini`. El prefijo predeterminado es `/QIBM/UserData/mqm`.
- Un literal, `qmgrs`.
- Un nombre de gestor de colas codificado, que es el nombre del gestor de colas transformado en un nombre de directorio válido. Por ejemplo, el gestor de colas `queue/manager` está representado por `queue&manager`.

Este proceso se denomina transformación de nombres.

## Transformación de nombres de gestores de colas del IFS

En IBM MQ, puede asignar a un gestor de colas un nombre que contenga hasta 48 caracteres.

Por ejemplo, puede denominar a un gestor de colas `QUEUE/MANAGER/ACCOUNTING/SERVICES`. Del mismo modo que cada gestor de colas crea una biblioteca, cada gestor de colas se representa mediante un archivo. Debido a puntos de código variantes en EBCDIC, existen limitaciones a los caracteres que se pueden utilizar en el nombre. Como resultado, los nombres de los archivos del IFS que representan objetos se transforman automáticamente para cumplir los requisitos del sistema de archivos.

Utilizando el ejemplo de un gestor de colas con el nombre `queue/manager`, transformando el carácter `/` en `&` y asumiendo el prefijo predeterminado, el nombre del gestor de colas en IBM MQ for IBM i se convierte en `/QIBM/UserData/mqm/qmgrs/queue&manager`.

## Transformación de nombres de objetos

Los nombres de objetos no son necesariamente nombres válidos del sistema de archivos, por lo que es posible que los nombres de objetos deban transformarse. El método utilizado es distinto del empleado para los nombres de los gestores de colas porque, aunque sólo hay unos pocos nombres de gestores de colas para cada máquina, puede haber un gran número de otros objetos para cada gestor de colas. Sólo las definiciones de procesos, colas y listas de nombres están representadas en el sistema de archivos; los canales no se ven afectados por estas consideraciones.

Cuando el proceso de transformación genera un nuevo nombre, no hay ninguna relación simple con el nombre de objeto original. Puede utilizar el mandato `DSPMQMOBJN` para ver los nombres transformados para objetos IBM MQ.

## Gestión de colas distribuidas y clústeres

---

La gestión de colas distribuidas hace referencia al envío de mensajes desde un gestor de colas a otro. El gestor de colas receptor puede estar en la misma máquina o en otra; puede estar próximo o en el otro lado del mundo. Puede ejecutarse en la misma plataforma que el gestor de colas local o puede estar en cualquiera de las plataformas soportadas por IBM MQ. Puede definir manualmente todas las conexiones en un entorno de colas distribuidas o puede crear un clúster y dejar que IBM MQ defina automáticamente gran parte del detalle de conexión.

## Gestión de colas distribuidas

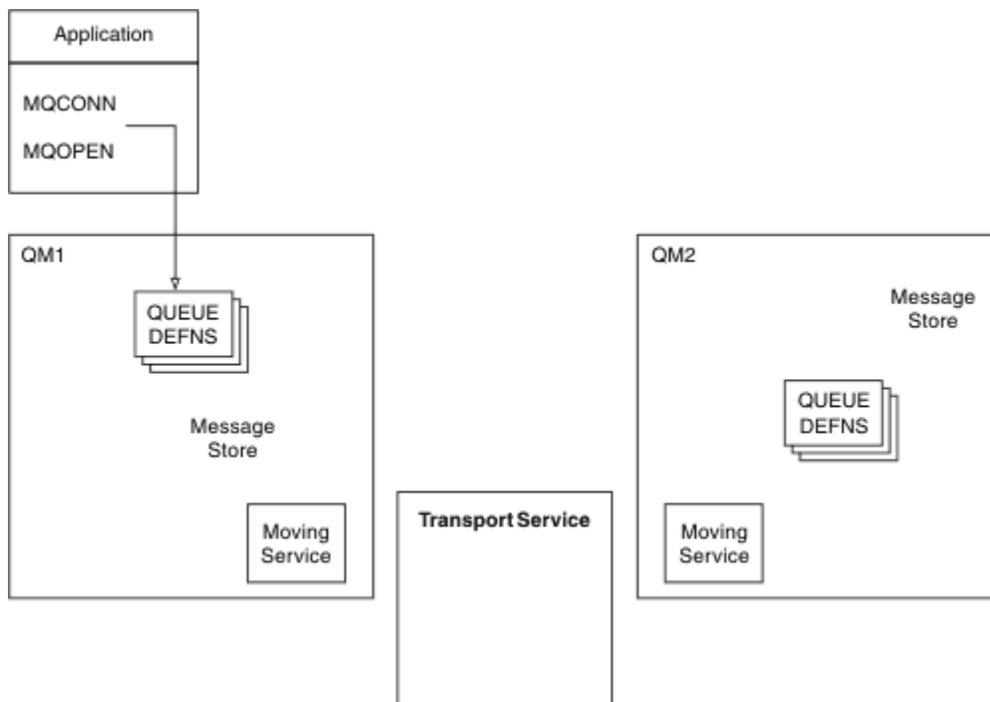


Figura 4. Visión general de los componentes de la gestión de colas distribuidas

En la figura anterior:

- Una aplicación utiliza la llamada MQCONN para conectarse a un gestor de colas. A continuación, la aplicación utiliza la llamada MQOPEN para abrir una cola de modo que pueda poner mensajes en dicha cola.
- Cada gestor de colas tiene una definición para cada una de sus colas. Puede tener definiciones de *colas locales* (es decir, alojadas en este gestor de colas) y definiciones de *colas remotas* (es decir, alojadas en otros gestores de colas).
- Si los mensajes van destinados a una cola remota, el gestor de colas local los retiene en una *cola de transmisión*, que los persiste en un almacén de mensajes hasta que puedan reenviarse al gestor de colas remoto.
- Cada gestor de colas contiene un software de comunicaciones, conocido como *servicio de mudanzas*, que utiliza para comunicarse con otros gestores de colas.
- El *servicio de transporte* es independiente del gestor de colas y puede ser uno de los siguientes (dependiendo de la plataforma):
  - Systems Network Architecture Advanced Program-to-Program Communication (SNA APPC)
  - Protocolo de control de la transmisión/Protocolo de Internet (TCP/IP)
  - Network Basic Input/Output System (NetBIOS)
  - Sequenced Packet Exchange (SPX)

### Componentes necesarios para enviar un mensaje

Si se va a enviar un mensaje a un gestor de colas remoto, el gestor de colas local necesita las definiciones de una *cola de transmisión* y de un *canal*. Un canal es un enlace de comunicaciones unidireccional entre dos gestores de colas. Puede transportar mensajes destinados a cualquier número de colas del gestor de colas remoto.

Cada extremo de un canal tiene una definición independiente, definiéndose, por ejemplo, como el extremo emisor y el extremo receptor. Un canal sencillo consta de una definición de canal *emisor* en el gestor de colas local y una definición de canal *receptor* en el gestor de colas remoto. Estas dos definiciones deben tener el mismo nombre y juntas conforman un canal.

El software que maneja el envío y recepción de mensajes es el *Agente de canal de mensajes* (MCA). Hay un *agente de canal de mensajes* (MCA) en cada extremo de un canal.

Cada gestor de colas debe tener un *cola de mensajes no entregados* (también conocido como *cola de mensajes no entregados*). Los mensajes se colocan en esta cola si no se pueden entregar en su destino.

La figura siguiente muestra la relación entre gestores de colas, colas de transmisión, canales y los MCA:

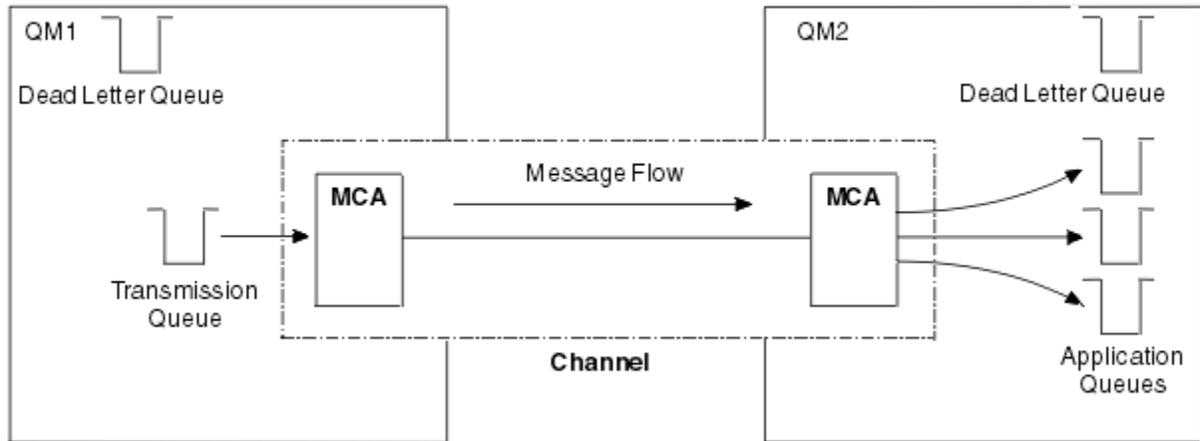


Figura 5. Envío de mensajes

### Componentes necesarios para devolver un mensaje

Si la aplicación necesita que el gestor de colas remoto devuelva mensajes, debe definir otro canal, que se ejecutará en el sentido contrario entre los gestores de colas, tal como se muestra en la figura siguiente:

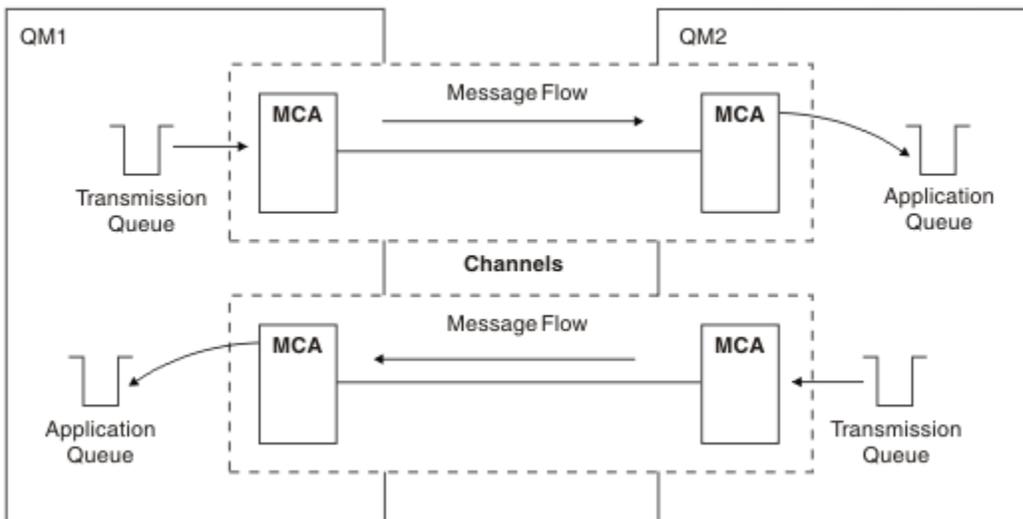


Figura 6. Envío de mensajes en ambos sentidos

### Clústeres

En lugar de definir manualmente todas las conexiones en un entorno de gestión de colas distribuidas, puede agrupar un conjunto de gestores de colas en un clúster. Cuando se hace esto, los gestores de colas pueden poner sus colas a disposición de otros gestores de colas en el clúster sin necesidad de crear explícitamente definiciones de canal, definiciones de cola remota ni colas de transmisión por cada destino. Cada gestor de colas de un clúster tiene una cola de transmisión única que transmite mensajes a cualquier otro gestor de colas del clúster. Por cada gestor de colas sólo debe definir un canal de clúster receptor y un canal de clúster emisor, los canales adicionales se gestionan automáticamente por el clúster.

Un cliente IBM MQ se puede conectar a un gestor de colas que forma parte de un clúster, del mismo modo que se puede conectar a cualquier otro gestor de colas. Al igual que ocurre con una gestión de colas distribuidas configurada manualmente, la llamada MQPUT se utiliza para poner un mensaje en una cola de cualquier gestor de colas. Puede utilizar la llamada MQGET para recuperar los mensajes de una cola local.

Los gestores de colas en las plataformas que soportan los clústeres no deben formar parte de un clúster. Puede seguir utilizando la configuración manual de gestión de colas distribuidas, así como (o en lugar de) utilizar clústeres.

### **Ventajas de utilizar clústeres**

La agrupación en clúster proporciona dos beneficios clave:

- Los clústeres simplifican la administración de las redes IBM MQ, que normalmente requieren la configuración de muchas definiciones de objeto para canales, colas de transmisión y colas remotas. Esta situación es especialmente útil en las redes de gran tamaño con posibilidad de cambios, donde deben interconectarse muchos gestores de colas. Se trata de una arquitectura particularmente difícil de configurar y mantener activa.
- Los clústeres se pueden utilizar para distribuir la carga de trabajo del tráfico de mensajes entre colas y gestores de colas en el clúster. Este tipo de distribución permite distribuir la carga de trabajo de mensajes de una sola cola entre varias instancias equivalentes de esa cola ubicadas en varios gestores de colas. Esta distribución de la carga de trabajo se puede utilizar para lograr una mayor resistencia a las anomalías del sistema y mejorar el rendimiento de escalado de flujos de mensajes especialmente activos en un sistema. En este tipo de entorno, cada una de las instancias de las colas distribuidas tienen aplicaciones consumidoras que procesan los mensajes. Para obtener más información, consulte [Utilización de clústeres para la gestión de carga de trabajo](#).

### **Cómo se direccionan los mensajes en un clúster**

Puede pensarse en un clúster como una red de gestores de colas mantenida por un administrador de sistemas meticuloso. Siempre que defina una cola de clúster, el administrador del sistema crea automáticamente las definiciones de cola remota correspondientes, según sea necesario en los otros gestores de colas.

No es necesario crear definiciones de cola de transmisión porque IBM MQ proporciona una cola de transmisión en cada gestor de colas del clúster. Esta única cola de transmisión se puede utilizar para llevar mensajes a cualquier otro gestor de colas del clúster. No está limitado a utilizar una cola de transmisión único. Un gestor de colas puede utilizar varias colas de transmisión para separar los mensajes que van a cada gestor de colas en un clúster. Por lo general, un gestor de colas utiliza una sola cola de transmisión de clúster. Puede cambiar el atributo de gestor de colas DEFCLXQ, para que el gestor de colas utilice una cola de transmisión de clúster diferente para cada gestor de colas en un clúster. También puede definir las colas de transmisión de clúster de clúster manualmente.

Todos los gestores de colas que se unen a un clúster acuerdan trabajar de esta forma. Envían información sobre ellos mismos y sobre las colas que contienen, y reciben información sobre los otros miembros del clúster.

Para asegurar que no se pierda información cuando un gestor de colas deja de estar disponible, se especifican dos gestores de colas en el clúster para que actúen como *repositorios completos*. Dichos gestores de colas almacenan una información completa sobre todos los gestores de colas y colas en el clúster. Todos los demás gestores de colas del clúster solo almacenan información sobre los gestores de colas y colas con los que intercambian mensajes. Estos gestores de colas se conocen como *repositorios parciales*. Para obtener más información, consulte [“Repositorio de clúster”](#) en la página 57.

Para poder formar parte de un clúster, un gestor de colas debe tener dos canales; un canal de clúster emisor y un canal de clúster receptor:

- Un canal de clúster emisor es un canal de comunicación como un canal emisor. Debe crear manualmente un canal de clúster emisor en un gestor de colas para conectarse a un repositorio completo que ya es miembro del clúster.

- Un canal de clúster receptor es un canal de comunicación como un canal receptor. Debe crear manualmente un canal de clúster receptor. El canal actúa como el mecanismo para que el gestor de colas reciba las comunicaciones de clúster.

Todos los demás canales necesarios para la comunicación entre este gestor de colas y otros miembros del clúster se crean automáticamente.

La figura siguiente muestra los componentes de un clúster llamado CLUSTER:

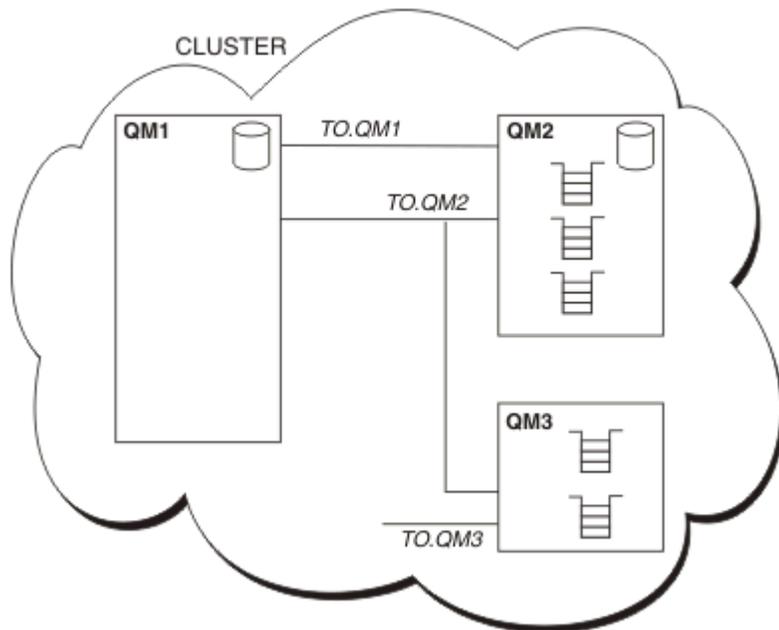


Figura 7. Un clúster de gestores de colas

- CLUSTER contiene tres gestores de colas, QM1, QM2 y QM3.
- QM1 y QM2 alojan repositorios completos de información sobre los gestores de colas y las colas del clúster.
- QM2 y QM3 alojan algunas colas de clúster, es decir, colas a las que puede acceder cualquier otro gestor de colas del clúster.
- Cada gestor de colas tiene un canal de clúster receptor denominado TO.qmgr en el que puede recibir mensajes.
- Cada gestor de colas también tiene un canal de clúster emisor mediante el que puede enviar información a uno de los gestores de colas de repositorio.
- QM1 y QM3 envían al depósito de QM2 y QM2 envía al depósito de QM1.

## Componentes de la gestión de colas distribuidas

Los componentes de la gestión de colas distribuidas son canales de mensaje, agentes de canal de mensaje, colas de transmisión, escuchas e iniciadores de canal y programas de salida de canal. La definición de cada extremo de un canal de mensajes puede ser de varios tipos.

Los canales de mensajes son los canales que transportan mensajes de un gestor de colas a otro. No hay que confundir los canales de mensajes con los canales MQI. Hay dos tipos de canales MQI, de conexión con el servidor (SVRCONN) y de conexión con el cliente (CLNTCONN). Para más información, consulte [Canales](#).

La definición de cada extremo de un canal de mensajes puede ser de uno de los tipos siguientes:

- Emisor (SDR)

- Receptor (RCVR)
- Servidor (SVR)
- Peticionario (RQSTR)
- Emisor de clúster (CLUSSDR)
- Receptor de clúster (CLUSRCVR)

Un canal de mensajes se define utilizando uno de estos tipos definidos en un extremo y un tipo compatible en el otro extremo. Las posibles combinaciones son:

- Emisor-receptor
- Peticionario-servidor
- Peticionario-emisor (devolución de llamada)
- Servidor-receptor
- Emisor de clúster-receptor de clúster

Las instrucciones detalladas para crear un canal emisor-receptor se incluyen en [Definición de los canales](#). Para obtener ejemplos de los parámetros necesarios para configurar los canales emisor-receptor, consulte la [Información de configuración de ejemplo aplicable a su plataforma](#). Para conocer los parámetros necesarios para definir un canal de cualquier tipo, consulte [DEFINE CHANNEL](#).

## Canales emisor-receptor

Un emisor en un sistema inicia el canal para que pueda enviar mensajes al otro sistema. El emisor solicita al receptor en el otro extremo del canal que se inicie. El emisor envía mensajes desde su cola de transmisión al receptor. El receptor pone los mensajes en la cola de destino. La [Figura 8 en la página 48](#) ilustra este proceso.

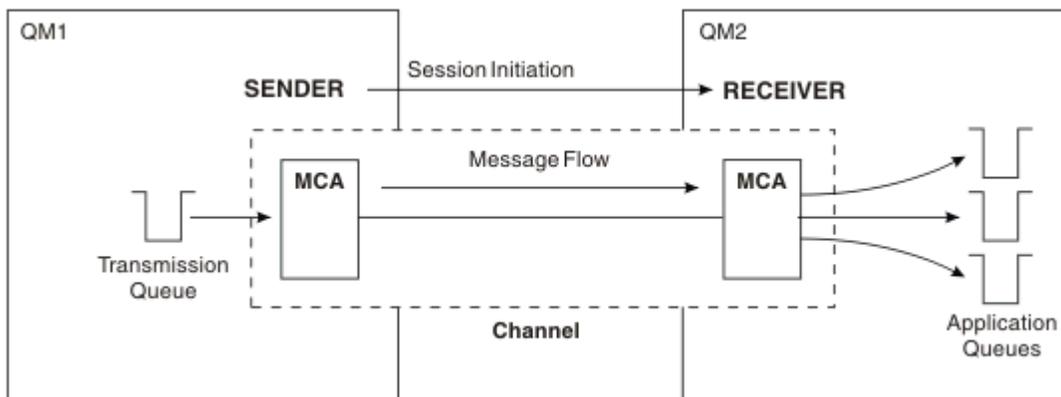


Figura 8. Un canal emisor-receptor

## Canales peticionario-servidor

Un peticionario en un sistema inicia el canal para que pueda recibir mensajes del otro sistema. El peticionario solicita al servidor en el otro extremo del canal que se inicie. El servidor envía mensajes al peticionario desde la cola de transmisión definida en su definición de canal.

Un canal servidor también puede iniciar la comunicación y enviar mensajes a un peticionario. Esto sólo se aplica a servidores *totalmente calificados*, es decir, canales servidor que tienen el nombre de conexión del socio especificado en la definición de canal. Un peticionario puede iniciar un servidor totalmente calificado o puede ser éste el que inicie la comunicación con un peticionario.

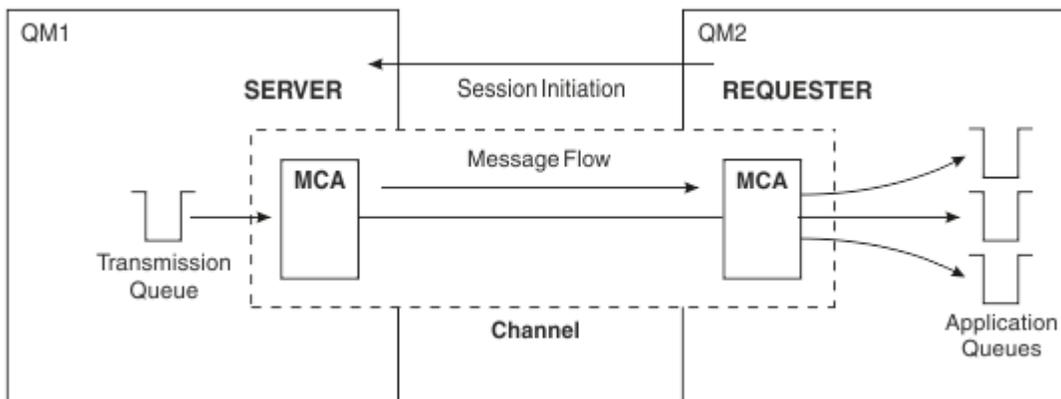


Figura 9. Un canal peticionario-servidor

### Canales peticionario-emisor

El peticionario inicia el canal y el emisor finaliza la llamada. Seguidamente, el emisor reinicia la comunicación en función de la información de su definición de canal (conocida como *devolución de llamada*). Envía mensajes desde la cola de transmisión al peticionario.

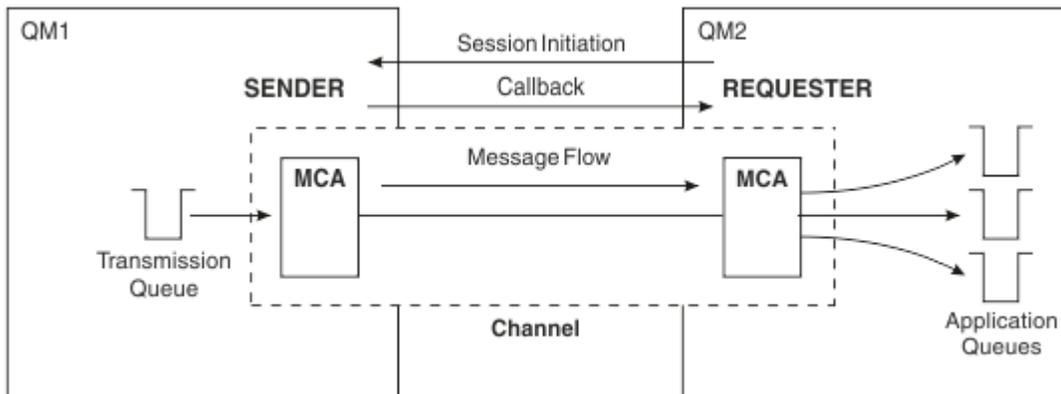


Figura 10. Un canal peticionario-emisor

### Canales servidor-receptor

Se trata del mismo caso que el de los canales emisor-receptor, pero sólo se aplica a servidores *totalmente calificados*, es decir, canales servidor que tienen el nombre de conexión del socio especificado en la definición de canal. El inicio del canal debe llevarse a cabo en el extremo servidor del enlace. La ilustración de este proceso es idéntica a la ilustración de la [Figura 8 en la página 48](#).

### Canales de clúster emisor

En un clúster, cada gestor de colas tiene un canal de clúster emisor mediante el que puede enviar información del clúster a uno de los gestores de cola de repositorio completo. Los gestores de colas también pueden enviar mensajes a otros gestores de colas en los canales de clúster emisor.

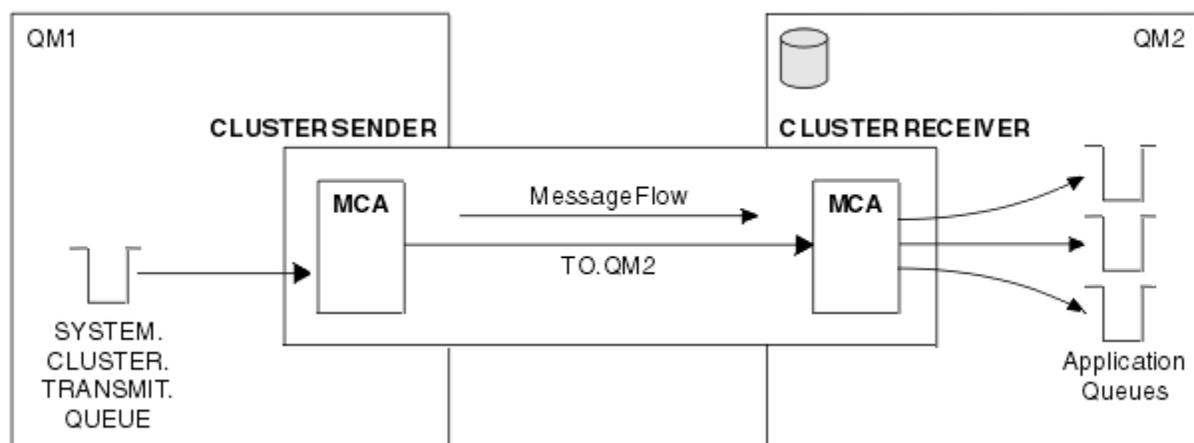


Figura 11. Un canal de clúster emisor

## Canales de clúster receptor

En un clúster, cada gestor de colas tiene un canal de clúster receptor en el que puede recibir mensajes e información sobre el clúster. La ilustración de este proceso es idéntica a la ilustración de la [Figura 11](#) en la [página 50](#).

## Colas de mensajes no entregados

La cola de mensajes no entregados es la cola a la que se envían los mensajes que no pueden enviarse a su destino correcto. Cada gestor de colas suele tener una cola de mensajes no entregados.

Una *cola de mensajes no entregados* (DLQ), denominada a veces *cola de mensajes sin entregar* es una cola que retiene los mensajes que no se pueden entregar a sus colas de destino, por ejemplo, porque la cola no exista o porque esté llena. Las colas de mensajes no entregados también se utilizan en el extremo emisor de un canal para los errores de conversión de datos. Cada gestor de colas en una red suele tener una cola local que se utiliza como cola de mensajes no entregados para que los mensajes que no pueden entregarse en su destino puedan almacenarse para recuperarlos posteriormente.

Los gestores de colas, los agentes de canal de mensajes (MCA) y las aplicaciones pueden transferir mensajes a las DLQ. Todos los mensajes de la DLQ deben tener como prefijo una estructura de *cabecera de mensaje no entregado*, MQDLH. El campo *Reason* de la estructura MQDLH contiene un código de razón que indica el motivo por el cual el mensaje está en la DLQ.

Normalmente debe definirse una cola de mensajes no entregados por cada gestor de colas. Si no lo hace y el MCA no puede poner un mensaje, se deja en la cola de transmisión y el canal se detiene. Además, si no se pueden entregar mensajes no permanentes de forma rápida (consulte [Mensajes rápidos y no permanentes](#)) y no existe ninguna cola de mensajes no entregados en el sistema de destino, estos mensajes se descartan.

No obstante, la utilización de colas de mensajes no entregados puede afectar al orden en que se entregan los mensajes, por lo que puede optar por no utilizarlas.

### Tareas relacionadas

[Trabajar con colas de mensajes no entregados](#)

[Resolución de problemas de mensajes no entregados](#)

### Referencia relacionada

[runmqdlq \(ejecutar manejador de cola de mensajes no entregados\)](#)

## Definiciones de colas remotas

Las definiciones de colas remotas son definiciones de las colas que pertenecen a otro gestor de colas.

Mientras que las aplicaciones pueden recuperar mensajes sólo de colas locales, pueden colocar mensajes en colas locales o colas remotas. Por lo tanto, además de una definición para cada una de sus colas locales, un gestor de colas puede tener *definiciones de colas remotas*. La ventaja de las definiciones de colas remotas es que permiten que una aplicación transfiera un mensaje a una cola remota sin tener que especificar el nombre de la cola remota o el gestor de colas remoto o el nombre de la cola de transmisión. Las definiciones de colas remotas le ofrecen independencia de la ubicación.

Existen otros usos para las definiciones de colas remotas, que se describen más adelante.

## Cómo acceder al gestor de colas remoto

Puede que no siempre tenga un canal entre cada uno de los gestores de colas de origen y de destino. Hay otras formas de conectarlos, incluyendo los saltos múltiples y el compartición de canales, utilizando canales diferentes y agrupaciones.

### Salto múltiple

Si no hay ningún enlace de comunicación directo entre el gestor de colas de origen y el gestor de colas de destino, es posible pasar por un *gestor de colas intermedio* (o más de uno) de camino al gestor de colas de destino. Esto se conoce como *salto múltiple*.

Debe definir canales entre todos los gestores de colas y colas de transmisión en los gestores de colas intermedios. Esto se muestra en la [Figura 12 en la página 51](#).

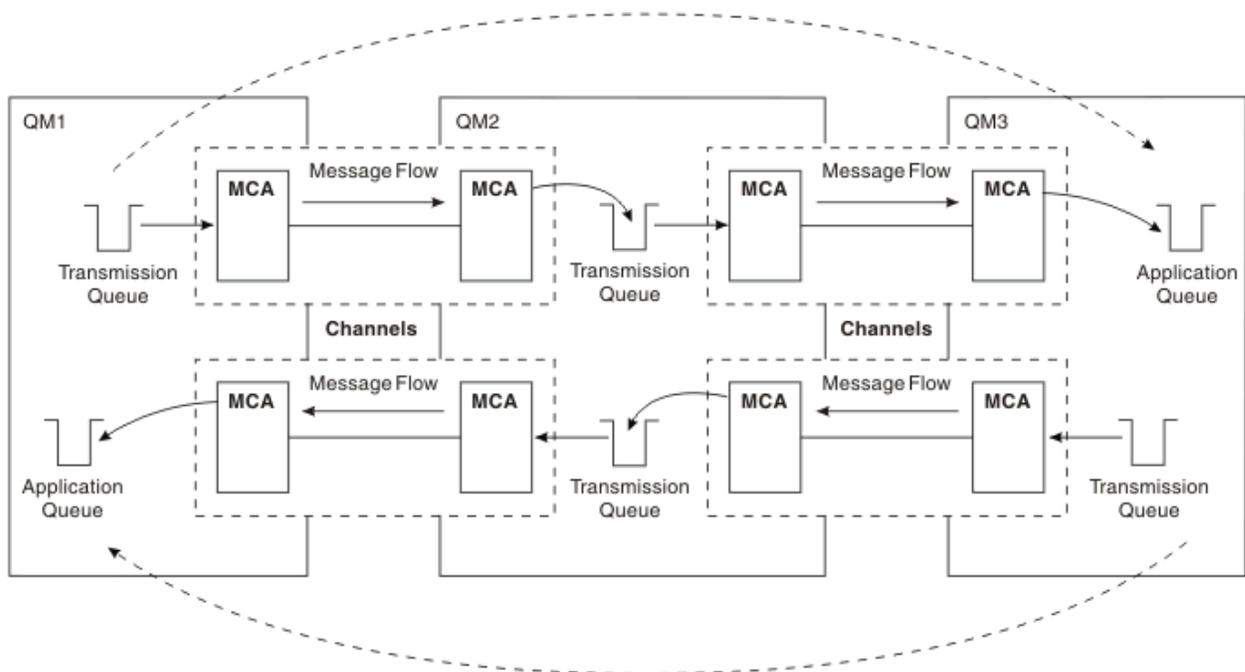


Figura 12. Paso a través de gestores de colas intermedios

### Compartición de canales

Como diseñador de aplicaciones, tiene la opción de obligar a las aplicaciones a especificar el nombre del gestor de colas remoto junto con el nombre de la cola, o de crear una *definición de cola remota* para cada cola remota. Esta definición contiene el nombre del gestor de colas remoto, el nombre de la cola y el nombre de la cola de transmisión. De cualquier modo, los mensajes de todas las aplicaciones que utilizan colas en la misma ubicación remota se envían a través de la misma cola de transmisión. Esto se muestra en la [Figura 13 en la página 52](#).

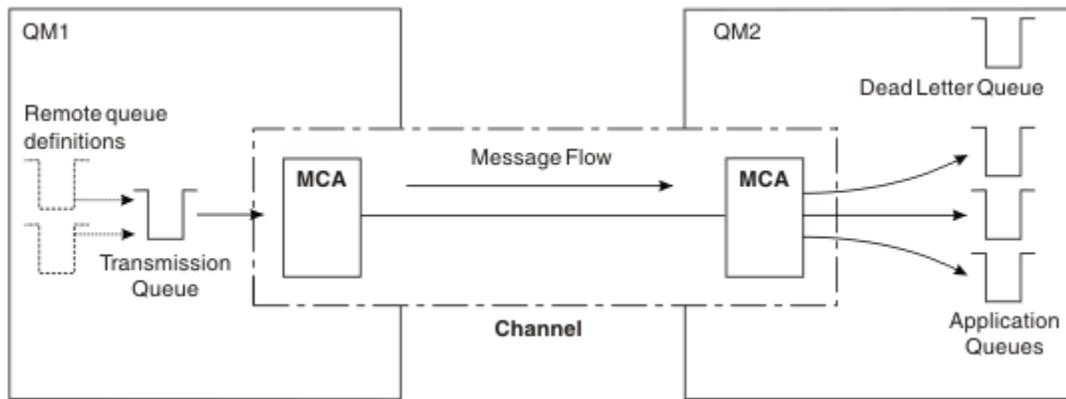


Figura 13. Compartición de una cola de transmisión

En la Figura 13 en la página 52 se muestra que los mensajes de varias aplicaciones a varias colas remotas pueden utilizar el mismo canal.

### Utilización de canales diferentes

Si tiene que enviar mensajes de tipos distintos entre dos gestores de colas, puede definir más de un canal entre los dos. Hay ocasiones en que se necesitan canales alternativos, puede que por razones de seguridad o tal vez porque a veces interese elegir una velocidad de entrega mayor comparada con la del tráfico de mensajes habitual.

Para configurar un segundo canal deberá definir otro canal y otra cola de transmisión y crear una definición de cola remota especificando la ubicación y el nombre de la cola de transmisión. A continuación, las aplicaciones pueden utilizar cualquier canal, pero los mensajes siguen entregándose a las mismas colas de destino. Esto se muestra en la Figura 14 en la página 52.

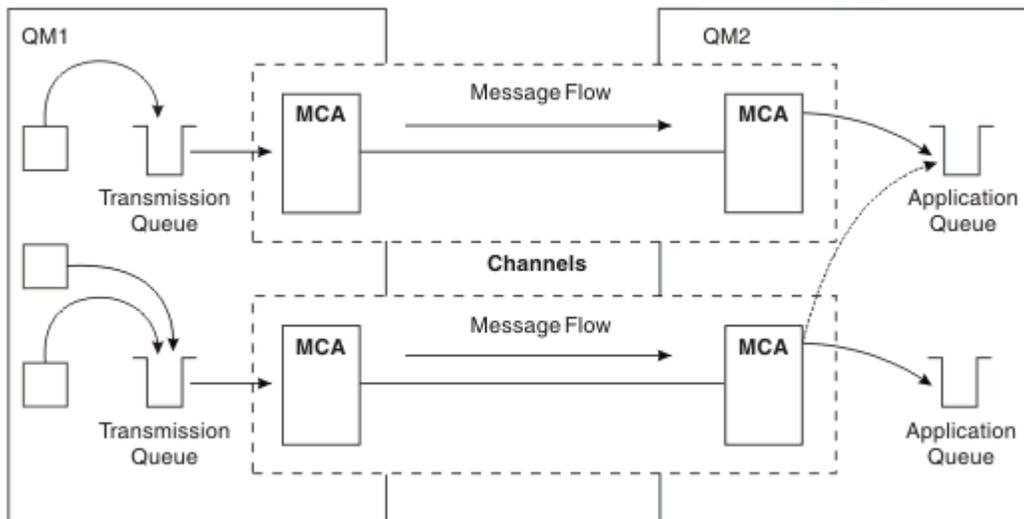


Figura 14. Utilización de varios canales

Cuando se utilizan definiciones de colas remotas para especificar una cola de transmisión, las aplicaciones **no** deben especificar la ubicación (es decir, el gestor de colas de destino) por sí mismas. Si lo hacen, el gestor de colas no utiliza las definiciones de cola remota. Las definiciones de colas remotas le ofrecen independencia de la ubicación. Las aplicaciones pueden poner mensajes en una cola *lógica* sin saber dónde se encuentra y se puede modificar la cola *física* sin tener que cambiar las aplicaciones.

## Utilización de clústeres

Cada gestor de colas de un clúster define un canal de clúster receptor. Cuando otro gestor de colas desea enviar un mensaje a ese gestor de colas, define el correspondiente canal de clúster emisor automáticamente. Por ejemplo, si hay más de una instancia de una cola en un clúster, el canal de clúster emisor se puede definir para cualquiera de los gestores de colas que aloja la cola. IBM MQ utiliza un algoritmo de gestión de carga de trabajo que utiliza una rutina en rueda para seleccionar un gestor de colas disponible adonde se dirige un mensaje. Puede obtener información adicional consultando [Clústeres](#).

## Información de direccionamiento

Cuando una aplicación transfiere mensajes que están destinados a un gestor de colas remoto, el gestor de colas local añade una cabecera de transmisión antes de colocarlos en la cola de transmisión. Esta cabecera contiene el nombre de la cola y el gestor de colas de destino, es decir, la *información de direccionamiento*.

En un entorno de gestor de colas único, la dirección de una cola de destino se establece cuando una aplicación abre una cola para colocar mensajes en ella. Como la cola de destino está en el mismo gestor de colas, no se necesita información de direccionamiento.

En un entorno de gestión de colas distribuidas, el gestor de colas necesita saber no sólo el nombre de la cola de destino, sino también la ubicación de esa cola (es decir, el nombre del gestor de colas), y la ruta a la ubicación remota (es decir, la cola de transmisión). Esta información de direccionamiento se encuentra en la cabecera de transmisión. El canal receptor elimina la cabecera de transmisión y utiliza la información que hay en ella para localizar la cola de destino.

Puede evitar la necesidad de que las aplicaciones especifiquen el nombre del gestor de colas de destino si utiliza una definición de cola remota. Esta definición especifica el nombre de la cola remota, el nombre del gestor de colas remoto al que van destinados los mensajes y el nombre de la cola de transmisión utilizado para transportar los mensajes.

## ¿Qué son los alias?

Los alias se utilizan para ofrecer una calidad de servicio a los mensajes. El alias del gestor de colas permite al administrador del sistema modificar el nombre de un gestor de colas de destino sin tener que cambiar las aplicaciones. También permite que el administrador del sistema modifique la ruta a un gestor de colas de destino o establecer una ruta que implica pasar por otros gestores de colas (saltos múltiples). El alias de la cola de respuestas proporciona una calidad de servicio a las respuestas.

Los alias de gestor de colas y los alias de cola de respuestas se crean utilizando una definición de cola remota que tiene un RNAME vacío. Estas definiciones no definen colas reales; las utiliza el gestor de colas para determinar nombres de colas físicas, nombres de gestores de colas y colas de transmisión.

Las definiciones de alias se caracterizan por tener un RNAME vacío.

## Resolución de nombres de colas

La resolución de nombres de colas se lleva a cabo en cada gestor de colas cada vez que se abre una cola. Su finalidad es identificar la cola de destino, el gestor de colas de destino (que puede ser local) y la ruta a ese gestor de colas (que puede ser un valor nulo). El nombre resuelto tiene tres partes: el nombre del gestor de colas, el nombre de la cola y, si el gestor de colas es remoto, la cola de transmisión.

Si existe una definición de cola remota, no se hace referencia a las definiciones de alias. El nombre de cola proporcionado por la aplicación se resolverá en el nombre de la cola de destino, el gestor de colas remoto y la cola de transmisión especificada en la definición de cola remota. Para obtener información más detallada sobre la resolución de nombres de colas, consulte [Resolución de nombres de colas](#).

Si no hay ninguna definición de cola remota y se especifica un nombre de gestor de colas o se ha resuelto mediante el servicio de nombres, el gestor de colas comprueba si hay una definición de alias de gestor de colas que coincida con el nombre del gestor de colas suministrado. Si existe, se utiliza la información de la definición para resolver el nombre del gestor de colas en el nombre del gestor de colas de destino. La

definición de alias de gestor de colas también se puede utilizar para determinar la cola de transmisión del gestor de colas de destino.

Si el nombre de cola resuelto no es una cola local, tanto el nombre del gestor de colas como el nombre de la cola se incluyen en la cabecera de transmisión de todos los mensajes transferidos por la aplicación a la cola de transmisión.

La cola de transmisión utilizada normalmente tiene el mismo nombre que el gestor de colas resuelto, a menos que lo modifique una definición de cola remota o una definición de alias de gestor de colas. Si no ha definido una cola de transmisión de este tipo pero sí ha definido una cola de transmisión predeterminada, se utilizará éste.

 Los nombres de gestores de colas que se ejecutan en z/OS tienen un límite de cuatro caracteres.

## Definiciones de alias de gestor de colas

Las definiciones de alias de gestor de colas se aplican cuando una aplicación que abre una cola para transferir un mensaje, especifica el nombre de la cola **y** el nombre del gestor de colas.

Las definiciones de alias de gestor de colas tienen tres usos:

- Al enviar mensajes, volver a correlacionar el nombre del gestor de colas
- Al enviar mensajes, modificar o especificar la cola de transmisión
- Al recibir mensajes, determinar si el gestor de colas local es el destino previsto para los mensajes

## Mensajes de salida: volver a correlacionar el nombre del gestor de colas

Las definiciones de alias de gestor de colas pueden utilizarse para correlacionar el nombre del gestor de colas especificado en una llamada MQOPEN. Por ejemplo, una llamada MQOPEN especifica el nombre de cola THISQ y el nombre de gestor de colas YOURQM. En el gestor de colas local, hay la siguiente definición de alias de gestor de colas:

```
DEFINE QREMOTE (YOURQM) QMNAME (REALQM)
```

Esto muestra que el gestor de colas real que se va a utilizar, cuando una aplicación transfiere mensajes al gestor de colas YOURQM, es REALQM. Si el gestor de colas local es REALQM, pone los mensajes en la cola THISQ, que es una cola local. Si el gestor de colas local no se llama REALQM, direcciona el mensaje a una cola de transmisión de clúster denominada REALQM. El gestor de colas cambia la cabecera de transmisión para que diga REALQM en vez de YOURQM.

## Mensajes de salida: modificar o especificar la cola de transmisión

[Figura 15 en la página 55](#) un ejemplo en el que los mensajes llegan al gestor de colas QM1 con cabeceras de transmisión que muestran nombres de cola del gestor de colas QM3. En este ejemplo, se puede acceder a QM3 mediante saltos múltiples a través de QM2.

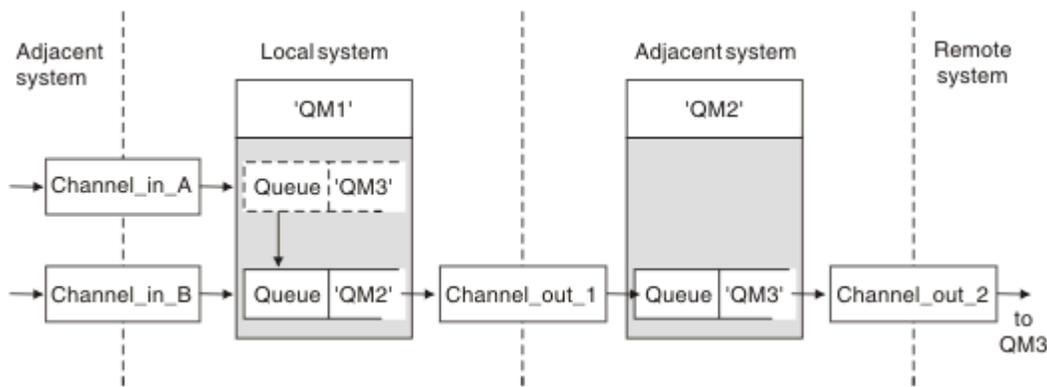


Figura 15. Alias del gestor de colas

Todos los mensajes para QM3 se capturan en QM1 con un alias de gestor de colas. El alias de gestor de colas se denomina QM3 y contiene la definición QM3 a través de la cola de transmisión QM2. La definición se parece a la del ejemplo siguiente:

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

El gestor de colas coloca los mensajes en la cola de transmisión QM2 pero no modifica la cabecera de la cola de transmisión debido a que el nombre del gestor de colas de destino, QM3, no se modifica.

Todos los mensajes que llegan a QM1 y muestran una cabecera de transmisión que contiene un nombre de cola en QM2 también se colocan en la cola de transmisión QM2. De este modo, los mensajes con destinos diferentes se reúnen en una cola de transmisión común de un sistema adyacente adecuado para transmitirlos a sus destinos.

## Mensajes de entrada: determinar el destino

Un MCA receptor abre la cola a la que se hace referencia en la cabecera de transmisión. Si existe una definición de alias de gestor de colas con el mismo nombre que el gestor de colas al que se hace referencia, entonces el nombre del gestor de colas recibido en la cabecera de transmisión es sustituido por el RQMNAME de esa definición.

Este proceso tiene dos usos:

- Dirigir mensajes a otro gestor de colas
- Modificar el nombre del gestor de colas para que sea el mismo que el del gestor de colas local

## Definiciones de alias de colas de respuesta

Una definición de alias de cola de respuestas especifica nombres alternativos para la información de respuesta en el descriptor de mensaje. La ventaja que tiene es que se puede cambiar el nombre de una cola o de un gestor de colas sin tener que modificar las aplicaciones.

## Resolución de nombres de colas

Cuando una aplicación responde a un mensaje, utiliza los datos del *descriptor de mensaje* del mensaje que ha recibido para averiguar el nombre de la cola a la que tiene que responder. La aplicación emisora indica donde hay que enviar las respuestas y asocia esta información a sus mensajes. Este concepto debe coordinarse como parte del diseño de la aplicación.

La resolución de nombres de colas tiene lugar en el extremo emisor de la aplicación, antes de que el mensaje se transfiera a una cola. Por lo tanto, la resolución de nombres de colas se produce antes de la interacción con la aplicación remota a la que el mensaje se está enviando. Esta es la única situación en la que la resolución de nombres tiene lugar en un momento en el que no se abre la cola.

## Resolución de nombres de cola utilizando un alias de gestor de colas

Normalmente, una aplicación especifica una cola de respuestas y deja el nombre del gestor de colas de respuestas en blanco. El gestor de colas completa su nombre en el momento de la transferencia. Este método funciona bien excepto cuando se desea utilizar un canal alternativo para las respuestas, por ejemplo, un canal que utilice la cola de transmisión "QM1\_relief" en lugar del canal de retorno por defecto que utiliza la cola de transmisión "QM1". En esta situación, los nombres de los gestores de colas especificados en las cabeceras de la cola de transmisión no coinciden con los nombres "reales" de los gestores de colas, pero se vuelven a especificar utilizando las definiciones de alias del gestor de colas. Para devolver las respuestas siguiendo rutas alternativas, también es necesario correlacionar los datos de la cola de respuestas, utilizando las definiciones de alias de cola de respuestas.

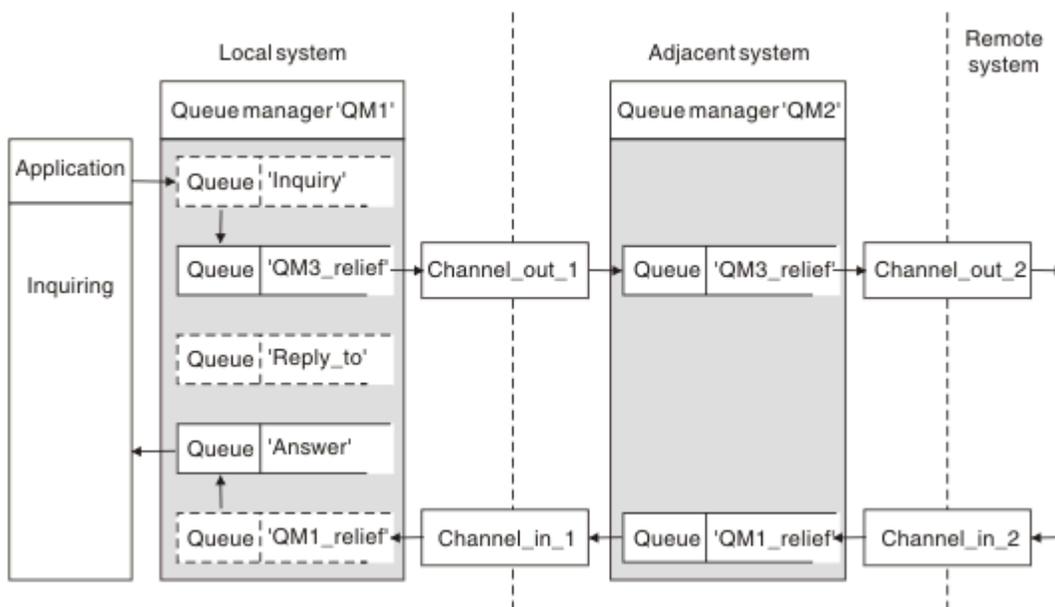


Figura 16. Alias de cola de respuestas utilizados para cambiar la ubicación de las respuestas

En el ejemplo de Figura 16 en la página 56:

1. La aplicación transfiere un mensaje utilizando la llamada MQPUT y especificando la información siguiente en el descriptor de mensaje:

```
ReplyToQ='Reply_to'
ReplyToQMgr=''
```

ReplyToQMgr debe estar en blanco para poder utilizar el alias de cola de respuestas.

2. Se crea una definición de alias de cola de respuesta llamada 'Reply\_to', que contiene el nombre 'Answer', y el nombre de gestor de cola 'QM1\_relief'.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
RQMNAME ('QM1_relief')
```

3. The messages are sent with a message descriptor showing ReplyToQ='Answer' and ReplyToQMgr='QM1\_relief'.
4. La especificación de la aplicación debe incluir la información de que las respuestas se encuentran en la cola "Answer" en lugar de "Reply\_to".

Para prepararse para las respuestas debe crear el canal de retorno paralelo, definiendo:

- En QM2, la cola de transmisión denominada "QM1\_relief"

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- En QM1, el alias de gestor de colas 'QM1\_relief'

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

Este alias de gestor de colas termina la cadena de canales de retorno paralelos y captura los mensajes destinados a QM1.

Si cree que podría necesitar hacerlo en el futuro, asegúrese de que las aplicaciones utilizan el nombre de alias desde el principio. Por ahora, es un alias de cola normal para la cola de respuestas, pero más adelante puede cambiarse por un alias de gestor de colas.

## Nombre de cola de respuestas

Hay que prestar atención a la hora de denominar las colas de respuestas. El motivo por el que una aplicación transfiere un nombre de cola de respuestas al mensaje es que puede especificar la cola a la que se envían sus respuestas. Cuando se crea una definición de alias de cola de respuestas con este nombre, la cola de respuestas real (es decir, una definición de cola local) no puede tener el mismo nombre. Por lo tanto, la definición de alias de cola de respuestas debe contener un nombre de cola nuevo además del nombre del gestor de colas y la especificación de la aplicación debe incluir la información de que sus respuestas se encuentra en esta otra cola.

Las aplicaciones ahora tienen que recuperar los mensajes de una cola distinta de la que habían denominado como cola de respuestas cuando transmitieron el mensaje original.

## Componentes de clúster

Los clústeres están formados por gestores de colas, repositorios de clúster, canales de clúster y colas de clúster.

Consulte los subtemas siguientes para obtener información sobre cada uno de los componentes del clúster:

### Conceptos relacionados

[Comparación de agrupación en clúster y gestión de colas distribuidas](#)

### Tareas relacionadas

[Configuración de un clúster de gestores de colas](#)

[Configurar un nuevo clúster](#)

## Repositorio de clúster

Un repositorio es una recopilación de información sobre los gestores de colas que son miembros de un clúster.

La información del repositorio incluye los nombres de los gestores de colas, sus ubicaciones, sus canales, las colas que alojan y otra información. La información se almacena en forma de mensajes en una cola llamada `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Esta cola es uno de los objetos

predeterminados.  En [Multiplataformas](#), se define al crear un gestor de colas de IBM

MQ.  En IBM MQ for z/OS, se define como parte de la personalización del gestor de colas.

## Depósito completo y depósito parcial

Normalmente, dos gestores de colas en un clúster contienen un repositorio completo. Los gestores de colas restantes contienen todos un depósito parcial.

Un gestor de colas que aloja un conjunto completo de información sobre todos los gestores de colas del clúster tiene un repositorio completo. Los otros gestores de colas del clúster tienen repositorios parciales que contienen un subconjunto de la información de los repositorios completos.

Un repositorio parcial contiene información sólo sobre los gestores de colas con los que el gestor de colas tiene que intercambiar mensajes. Los gestores de colas solicitan actualizaciones de la información que necesitan, de modo que si ésta cambia, el gestor de colas de repositorio completo les envía la nueva información. Durante la mayor parte del tiempo, un repositorio parcial contiene toda la información que un gestor de colas necesita para funcionar en el clúster. Cuando un gestor de colas necesita información adicional, realiza consultas al depósito completo y actualiza su depósito parcial. Los gestores de colas utilizan la cola `SYSTEM.CLUSTER.COMMAND.QUEUE` para solicitar y recibir actualizaciones de los repositorios.

Cuando migre los gestores de colas que son miembros de un clúster, migre los repositorios completos antes que los repositorios parciales. Esto se debe a que un repositorio antiguo no puede almacenar los atributos nuevos introducidos en un release más reciente. Los tolera, pero no los almacena.

## Gestor de colas de clúster

Un gestor de colas de clúster es un gestor de colas que es miembro de un clúster.

Un gestor de colas puede ser miembro de más de un clúster. Cada gestor de colas de clúster debe tener un nombre que sea exclusivo en todos los clústeres de los que es miembro.

Un gestor de colas de clúster puede alojar colas, lo cual anuncia a los otros gestores de colas del clúster. Sin embargo, no tiene por qué hacerlo. En vez de ello, puede suministrar mensajes a las colas alojadas en otro lugar del clúster y recibir únicamente las respuestas que vayan dirigidas explícitamente a él.

 En IBM MQ for z/OS, un gestor de colas de clúster puede ser miembro de un grupo de compartición de colas. En este caso, comparte sus definiciones de cola con otros gestores de colas del mismo grupo de compartición de colas.

Los gestores de colas de clúster son autónomos. Tienen control total sobre las colas y los canales que definen. Sus definiciones no sea pueden modificar con otros gestores de colas excepto los gestores de colas del mismo grupo de compartición de colas. Los gestores de colas de repositorio no controlan las definiciones de otros gestores de colas del clúster. Contienen un conjunto completo de todas las definiciones, para utilizarlas cuando sea necesario. Un clúster es una federación de gestores de colas.

Después de crear o modificar una definición en un gestor de colas de clúster, la información se envía al gestor de colas de repositorio completo. Los otros repositorios del clúster se actualizan posteriormente.

## Gestor de colas de repositorio completo

Un gestor de colas de repositorio completo es un gestor de colas de clúster que contiene una representación completa de los recursos del clúster. Para asegurar la disponibilidad, configure dos o más gestores de colas de repositorio completo en cada clúster. Los gestores de colas de repositorio completo reciben información enviada por los otros gestores de colas del clúster y actualizan sus repositorios. Se envían mensajes entre sí para asegurarse de que ambos se mantienen actualizados con información nueva sobre el clúster.

## Gestores de colas y repositorios

Cada clúster tiene al menos un gestor de colas, preferiblemente dos, que contienen repositorios completos de información sobre los gestores de colas, las colas y los canales de un clúster. Estos repositorios también contienen solicitudes de los otros gestores de colas del clúster de actualizaciones de la información.

Los otros gestores de colas contienen cada uno un repositorio parcial, que contiene información sobre el subconjunto de colas y gestores de colas con los tienen que comunicarse. Los gestores de colas crean sus repositorio parciales realizando consultas cuando tienen que acceder por primera vez a otra cola o gestor de colas. Solicitan que se les notifique de cualquier nueva información sobre esa cola o gestor de colas.

Cada gestor de colas almacena su información de repositorio en mensajes en una cola llamada SYSTEM.CLUSTER.REPOSITORY.QUEUE. Los gestores de colas intercambian información de repositorio en mensajes en una cola llamada SYSTEM.CLUSTER.COMMAND.QUEUE.

Cada gestor de colas que se une a un clúster define un canal de clúster emisor, CLUSSDR, a uno de los repositorios. Se entera inmediatamente de qué otros gestores de colas del clúster contienen repositorios completos. A partir de entonces, el gestor de colas puede solicitar información de cualquiera de los repositorios. Cuando el gestor de colas envía información al repositorio elegido, también envía información al otro repositorio (si hay uno).

Un repositorio completo se actualiza cuando el gestor de colas que lo aloja recibe nueva información de uno de los gestores de colas al que está enlazado. La nueva información también se envía al otro repositorio, para reducir el riesgo de que se retrase si un gestor de colas de repositorio está fuera de servicio. Debido a que toda la información se envía dos veces, los repositorios tienen que descartar duplicados. Cada elemento de información contiene un número de secuencia, que los repositorios utilizan para identificar los duplicados. Todos los repositorios se mantienen sincronizados entre sí mediante el intercambio de mensajes.

## Colas de clúster

Una cola de clúster es una cola que se aloja en un gestor de colas de clúster y que está disponible para otros gestores de colas del clúster.

Una definición de cola de clúster se anuncia en otros gestores de colas del clúster. Los otros gestores de colas del clúster pueden transferir mensajes a una cola de clúster sin necesidad de que haya una definición de cola remota correspondiente. Una cola de clúster se puede anunciar en más de un clúster utilizando una lista de nombres de clúster.

Cuando se anuncia una cola, cualquier gestor de colas del clúster puede poner mensajes en ella. Para transferir un mensaje, el gestor de colas debe averiguar, en los repositorios completos, donde está alojada la cola. A continuación, añade información de direccionamiento al mensaje y pone el mensaje a una cola de transmisión de clúster.

 Una cola de clúster puede ser una cola que se comparte entre miembros de un grupo de compartición de colas en IBM MQ for z/OS.

### Tareas relacionadas

[Definición de cola de clúster](#)

## Comparación entre las colas compartidas y las colas de clúster

Esta información está pensada para ayudarle a comparar las colas compartidas y las colas de clúster y a decidir cuáles pueden ser más adecuadas para su sistema.

### Costes del iniciador de canal

En las colas de clúster, los mensajes los envían los canales, por lo que deberán tenerse en cuenta los costes del iniciador de canal además de los costes de aplicación. Hay costes en la red porque los canales obtienen y transfieren mensajes. Estos costes no existen con las colas compartidas que, por lo tanto, utilizan menos potencia de proceso que las colas de clúster cuando mueven mensajes entre los gestores de colas de un grupo de compartición de colas.

### Disponibilidad de los mensajes

Al transferir a una cola, las colas de clúster envían el mensaje a uno de los gestores de colas con canales activos conectados al gestor de colas. En el gestor de colas remoto, si las aplicaciones utilizadas para procesar los mensajes no están funcionando, los mensajes no se procesan y esperan hasta que las aplicaciones se inician. De forma similar, si se concluye un gestor de colas, todos los mensajes en el gestor de colas no estarán disponible hasta que se reinicie el gestor de colas. Estos ejemplos muestran una menor disponibilidad de mensajes que al utilizar colas compartidas.

Cuando se utilizan colas compartidas, cualquier aplicación en el grupo de compartición de colas puede obtener mensajes que se envían. Si concluye un gestor de colas del grupo de compartición de colas, los mensajes están disponibles para los otros gestores de colas, lo que proporciona una mayor disponibilidad de mensajes que cuando se utilizan colas de clúster.

## Capacidad

Un recurso de acoplamiento es más costoso que un disco; por lo tanto, el coste de almacenar 1.000.000 mensajes en una cola local es menor que el que tener un recurso de acoplamiento con capacidad suficiente para almacenar el mismo número de mensajes.

## Envío a otros gestores de colas

Los mensajes de colas compartidas sólo están disponibles dentro de un grupo de compartición de colas. Si se desea utilizar un gestor de colas fuera del grupo de compartición de colas, deberán utilizarse canales. Puede utilizar la agrupación en clúster para equilibrar la carga de trabajo entre varios gestores de colas distribuidas remotos.

## Equilibrio de la carga de trabajo

Puede utilizar la agrupación en clúster para influir en qué canales y gestores de colas obtienen una parte de los mensajes enviados. Por ejemplo, puede enviar el 60% de mensajes a un gestor de colas, y el 40% de mensajes a otro gestor de colas. Este ejemplo no depende de la capacidad del gestor de colas remoto para procesar el trabajo. El sistema con el primer gestor de colas puede estar sobrecargado y el sistema con el segundo gestor de colas puede estar desocupado, pero la mayoría de los mensajes irán de todos modos al primer gestor de colas.

Con las colas compartidas, dos sistemas CICS pueden obtener mensajes. Si un sistema está sobrecargado, el otro sistema asume la mayor parte de la carga de trabajo.

## Canales de clúster

En cada repositorio completo, puede definir manualmente un canal de clúster receptor y un conjunto de canales de clúster emisor para conectarse a sendos repositorios completos en el clúster. Cuando añada un repositorio parcial, debe definir manualmente un canal de clúster receptor y un canal de clúster emisor individual que se conectarán a uno de los repositorios completos. El clúster definirá automáticamente los canales de clúster emisor adicionales cuando sea necesario. Los canales de clúster emisor definidos automáticamente toman sus atributos de la definición de canal de clúster receptor correspondiente en el gestor de colas receptor.

### Canal de clúster receptor: CLUSRCVR

Una definición de canal CLUSRCVR define el extremo de un canal en el que un gestor de colas de clúster puede recibir mensajes de otros gestores de colas del clúster.

Debe definir al menos un canal CLUSRCVR para cada gestor de colas de clúster. Al definir el canal CLUSRCVR, el gestor de colas indica a los otros gestores de colas de clúster que está disponible para recibir mensajes.

Una definición de canal CLUSRCVR permite a otros gestores de colas definir automáticamente las definiciones de canal de clúster emisor correspondientes. Consulte la sección [“Canales de clúster emisor definidos automáticamente”](#) en la página 61 de este artículo.

### Canal de clúster emisor: CLUSSDR

El administrador debe definir un canal CLUSSDR desde cada gestor de colas de repositorio completo a cualquier otro gestor de colas de repositorio completo del clúster. Todas las actualizaciones intercambiadas por los repositorios completos fluyen exclusivamente en estos canales. Al definir manualmente estos canales, puede controlar la red de repositorios completos de forma explícita.

Cuando añada un gestor de colas de repositorio parcial a un clúster, puede definir manualmente un solo canal CLUSSDR para conectarse a uno de los repositorios completos. Existen pocas diferencias en el repositorio completo que elija, porque, después de realizar el contacto inicial, los objetos adicionales del gestor de colas del clúster del gestor de colas, incluidos los canales CLUSSDR, se definen automáticamente según sea necesario. Esto permite que el gestor de colas envíe información de clúster a cualquier repositorio completo y enviar mensajes a cualquier gestor de colas del clúster.

Tal como se explica en la sección de este artículo, los canales de emisor definidos automáticamente se basan en la configuración del canal de clúster receptor. Por lo tanto, las propiedades de canal que establezca en los canales de clúster deben definirse de forma idéntica en CLUSSDR y los canales del clúster receptor coincidentes, o sólo en los canales del clúster receptor.

Sólo debe definir manualmente canales CLUSSDR por los motivos descritos anteriormente. Es decir, para conectar inicialmente un repositorio parcial a un repositorio completo, o para conectar dos repositorios completos. La configuración manual de un canal CLUSSDR que se conecta a un repositorio parcial o a un gestor de colas que no está en el clúster, hace que se emitan mensajes de error como [AMQ9427](#) y [AMQ9428](#). Aunque es posible que algunas veces esto sea inevitable como situación temporal, por ejemplo, al modificar la ubicación de un repositorio completo, la definición manual se deberá suprimir lo antes posible.

## Canales de clúster emisor definidos automáticamente

Normalmente, cuando añade un gestor de colas de repositorio parcial a un clúster, sólo puede definir manualmente dos canales de clúster en el gestor de colas:

- Un canal de clúster emisor (CLUSSDR) a un gestor de colas de repositorio completo para el clúster.
- Un canal de clúster receptor (CLUSRCVR).

El canal CLUSSDR que defina permitirá que el gestor de colas establezca un contacto inicial con el clúster. Tras realizar el contacto inicial, el clúster definirá automáticamente canales CLUSSDR adicionales cuando sea necesario.

Un canal CLUSSDR definido automáticamente toma sus atributos de la definición de canal CLUSRCVR correspondiente en el gestor de colas receptor. Incluso si hay un canal CLUSSDR definido manualmente, se utilizan los atributos del canal CLUSSDR definido automáticamente. Supongamos, por ejemplo, que define un canal CLUSRCVR sin especificar un número de puerto en el parámetro **CONNNAME**, y define manualmente un canal CLUSSDR que especifica un número de puerto. Cuando el canal CLUSSDR definido automáticamente sustituye el que se ha definido manualmente, el número de puerto (tomado del canal CLUSRCVR) queda en blanco. Se utiliza el número de puerto predeterminado y el canal falla.

Cuando existan diferencias de configuración entre un canal CLUSSDR definido manualmente y la definición de canal CLUSRCVR correspondiente, algunas diferencias entrarán en vigor inmediatamente (por ejemplo, los parámetros de equilibrio de carga de trabajo) y otras entrarán en vigor sólo tras reiniciar el canal (por ejemplo, la configuración de TLS).

Para evitar confusiones, en la medida de lo posible tenga en cuenta las directrices siguientes:

- Sólo defina manualmente canales CLUSSDR para que apunten a repositorios completos.
- Cuando haya definido manualmente canales CLUSSDR, configúrelos para que coincidan exactamente con la definición de canal CLUSRCVR correspondiente en el gestor de colas receptor.

Consulte también [Trabajar con canales definidos automáticamente](#).

### Conceptos relacionados

[Trabajar con canales definidos automáticamente](#)

[Cómo trabajar con colas de transmisión de clúster y canales de clúster emisor](#)

### Tareas relacionadas

[Configurar un nuevo clúster](#)

[Añadir un gestor de colas a un clúster](#)

## Temas de clúster

Los temas de clúster son temas administrativos con el atributo **cluster** definido. La información sobre temas de clúster se envía a todos los miembros de un clúster y se combina con temas locales para crear partes de un espacio de tema que abarque varios gestores de colas. Esto permite que los mensajes publicados sobre un tema en un gestor de colas se entreguen a las suscripciones de otros gestores de colas del clúster.

Cuando se define un tema de clúster en un gestor de colas, la definición de tema de clúster se envía a los gestores de colas de depósito completo. Los depósitos completos propagan entonces la definición de tema de clúster a todos los gestores de colas del clúster, dejando el mismo tema de clúster disponible para publicadores y suscriptores en cualquier gestor de colas del clúster. El gestor de colas en los que se crea un tema de clúster se conoce como host de tema de clúster. El tema de clúster puede utilizarlo cualquier gestor de colas del clúster, pero las modificaciones de un tema de clúster deben realizarse en el gestor de colas donde se ha definido dicho tema (el host), momento en el cual la modificación se propaga a todos los miembros del clúster a través de los depósitos completos.

Para obtener información sobre cómo configurar temas de clúster para utilizar el *direccionamiento directo* o el *direccionamiento de host de tema* y sobre la herencia de temas de clúster y suscripciones de comodín, consulte [Definición de temas de clúster](#).

Para obtener información sobre los mandatos a utilizar para visualizar temas de clúster, consulte la información relacionada.

### Conceptos relacionados

[Trabajar con temas administrativos](#)

[Trabajar con suscripciones](#)

### Referencia relacionada

[DISPLAYTOPIC](#)

[DISPLAYTPSTATUS](#)

[DISPLAYSUB](#)

## Objetos de clúster predeterminado

 En Multiplatforms, los objetos de clúster predeterminados se incluyen en el conjunto de objetos predeterminados creados automáticamente cuando se define un gestor de colas.  En z/OS, las definiciones de objetos de clúster predeterminadas se pueden encontrar en los ejemplos de personalización.

**Nota:** Puede modificar las definiciones de canal predeterminadas igual que cualquier otra definición de canal, ejecutando mandatos MQSC o PCF. No altere las definiciones de cola predeterminadas, excepto SYSTEM.CLUSTER.HISTORY.QUEUE.

### SYSTEM.CLUSTER.COMMAND.QUEUE

Cada gestor de colas de un clúster tiene una cola local llamada SYSTEM.CLUSTER.COMMAND.QUEUE que se usa para transferir mensajes al repositorio completo. El mensaje contiene cualquier información nueva o modificada sobre el gestor de colas, o cualquier solicitud de información sobre otros gestores de colas. SYSTEM.CLUSTER.COMMAND.QUEUE normalmente está vacía.

### SYSTEM.CLUSTER.HISTORY.QUEUE

Cada gestor de colas de un clúster tiene una cola local denominada SYSTEM.CLUSTER.HISTORY.QUEUE. SYSTEM.CLUSTER.HISTORY.QUEUE se utiliza para almacenar el historial de información de estado del clúster con fines de servicio.

En los valores de objeto predeterminados, SYSTEM.CLUSTER.HISTORY.QUEUE se establece en PUT (ENABLED). Para suprimir la recopilación de historial, cambie el valor a PUT (DISABLED).

### SYSTEM.CLUSTER.REPOSITORY.QUEUE

Cada gestor de colas de un clúster tiene una cola local denominada SYSTEM.CLUSTER.REPOSITORY.QUEUE. Esta cola se utiliza para almacenar toda la información del depósito completo. Esta cola normalmente no está vacía.

## SYSTEM.CLUSTER.TRANSMIT.QUEUE

Cada gestor de colas tiene una definición para una cola local denominada SYSTEM.CLUSTER.TRANSMIT.QUEUE. SYSTEM.CLUSTER.TRANSMIT.QUEUE es la cola de transmisión predeterminada para todos los mensajes a todas las colas y gestores de colas que están dentro de clústeres. Puede cambiar la cola de transmisión predeterminada para cada canal de clúster emisor a SYSTEM.CLUSTER.TRANSMIT.ChannelName, cambiando el atributo de gestor de colas DEFCLXQ. No puede suprimir SYSTEM.CLUSTER.TRANSMIT.QUEUE. También se utiliza para definir comprobaciones de autorización si la cola de transmisión predeterminada que se utiliza es SYSTEM.CLUSTER.TRANSMIT.QUEUE o SYSTEM.CLUSTER.TRANSMIT.ChannelName.

## SYSTEM.DEF.CLUSRCVR

Cada clúster tiene una definición de canal CLUSRCVR predeterminada llamada SYSTEM.DEF.CLUSRCVR. SYSTEM.DEF.CLUSRCVR se utiliza para proporcionar valores predeterminados para todos los atributos que no especifique al crear un canal de clúster receptor en un gestor de colas del clúster.

## SYSTEM.DEF.CLUSSDR

Cada clúster tiene una definición de canal CLUSSDR predeterminada llamada SYSTEM.DEF.CLUSSDR. SYSTEM.DEF.CLUSSDR se utiliza para proporcionar valores predeterminados para todos los atributos que no especifique al crear un canal de clúster emisor en un gestor de colas del clúster.

## Conceptos relacionados

[Cómo trabajar con objetos de clúster predeterminado](#)

## Mensajería de publicación/suscripción

La mensajería de publicación/suscripción le permite separar el proveedor de información de los consumidores de esa información. La aplicación emisora y la aplicación receptor no necesitan saber nada una de la otra para la información que va a enviarse y recibirse.

Antes de que una aplicación IBM MQ de punto a punto pueda enviar un mensaje a otra aplicación, necesita saber algo sobre ella. Por ejemplo, necesita conocer el nombre de la cola a la que se va a enviar la información, y puede ser que también especifique un nombre de gestor de colas.

La publicación/suscripción de IBM MQ elimina la necesidad de que la aplicación sepa algo sobre la aplicación de destino. Todas las aplicaciones emisoras tienen que hacer esto:

- Colocar un mensaje IBM MQ que contenga la información que la aplicación desea.
- Asignar el mensaje a un tema que indique el asunto de la información.
- Deje que IBM MQ maneje la distribución de esa información.

Del mismo modo, no es necesario que la aplicación de destino sepa nada sobre el origen de la información que recibe.

La figura siguiente muestra el sistema de publicación/suscripción más simple. Hay un publicador, un gestor de colas, y un suscriptor. El suscriptor realiza una suscripción en un gestor de colas, se envía una publicación desde el publicador al gestor de colas y luego el gestor de colas reenvía la publicación al suscriptor.

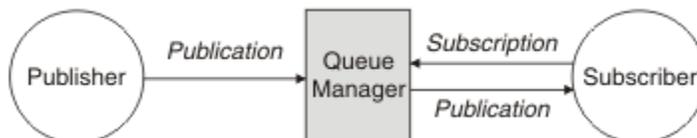


Figura 17. Configuración de publicación/suscripción simple

Un sistema de publicación/suscripción típico tiene más de un publicador y más de un suscriptor en muchos temas diferentes y con frecuencia tiene más de un gestor de colas. Una aplicación puede ser tanto un publicador como un suscriptor.

Otra diferencia significativa entre la mensajería de publicación/suscripción y la de punto a punto es que un mensaje enviado a una cola punto a punto solo es procesado por una única aplicación consumidora. Un mensaje publicado en un tema de publicación/suscripción, donde más de un suscriptor ha registrado un interés, es procesado por cualquier suscriptor interesado.

## Componentes de publicación/suscripción

La publicación/suscripción es el mecanismo por el que los suscriptores pueden recibir información, en forma de mensajes, de los publicadores. Los gestores de colas controlan las interacciones entre publicadores y suscriptores utilizando los recursos estándar de IBM MQ.

Un sistema de publicación/suscripción típico tiene más de un publicador y más de un suscriptor en muchos temas diferentes y con frecuencia tiene más de un gestor de colas. Una aplicación puede ser tanto un publicador como un suscriptor.

El proveedor de la información recibe el nombre de *publicador*. Los publicadores suministran información sobre un asunto sin necesidad de saber nada acerca de las aplicaciones que están interesadas en la información. Los publicadores generan esta información en forma de mensajes, denominados *publicaciones*, que quieren publicar y definir el tema de estos mensajes.

El consumidor de la información recibe el nombre de *suscriptor*. Los suscriptores crean *suscripciones* que describen el tema en el que está interesado el suscriptor. Por lo tanto, la suscripción determina qué publicaciones se reenvían al suscriptor. Los suscriptores pueden hacer varias suscripciones y pueden recibir información de muchos publicadores diferentes.

La información publicada se envía en un mensaje de IBM MQ y el asunto de la información se identifica por su *tema*. El publicador especifica el tema cuando publica la información, y el suscriptor especifica los temas sobre los que desea recibir publicaciones. Al suscriptor se le envía información sólo de los temas a los que se ha suscrito.

Es la existencia de temas lo que permite que los proveedores y consumidores de la información que se debe desacoplarse de la mensajería de publicación/suscripción eliminando la necesidad de incluir un destino específico en cada mensaje como se requiere en la mensajería punto a punto.

Todas las interacciones entre publicadores y suscriptores están controladas por un gestor de colas. El gestor de colas recibe los mensajes de los publicadores, y las suscripciones de los suscriptores (a un rango de temas). El trabajo del gestor de colas es direccionar los mensajes publicados para los suscriptores que hayan registrado su interés en el tema de los mensajes.

Los recursos estándar de IBM MQ se utilizan para distribuir mensajes, de modo que las aplicaciones pueden utilizar todas las características que están disponibles para las aplicaciones de IBM MQ existentes. Esto significa que puede utilizar mensajes permanentes para obtener la entrega asegurada una única vez, y que los mensajes pueden ser parte de una unidad transaccional de trabajo para asegurarse de que los mensajes se entregan al suscriptor sólo si son confirmados por el publicador.

## Publicadores y publicaciones

En la publicación/suscripción de IBM MQ, un publicador es una aplicación que hace que la información sobre un tema especificado esté disponible para un gestor de colas en forma de un mensaje IBM MQ estándar llamado publicación. Un publicador puede publicar información sobre más de un tema.

Los publicadores utilizan el verbo MQPUT para poner un mensaje en un tema abierto anteriormente, este mensaje es una publicación. A continuación, el gestor de colas local direcciona la publicación a todos los suscriptores que tengan suscripciones al tema de la publicación. Un mensaje publicado puede ser consumido por más de un suscriptor.

Además de distribuir aplicaciones a todos los suscriptores locales que tengan las suscripciones adecuadas, un gestor de colas también puede distribuir la publicación a otros gestores de colas conectados al mismo, directamente o a través de una red de gestores de colas que tengan suscriptores al tema.

En una red de publicación/suscripción de IBM MQ, una aplicación de publicación también puede ser un suscriptor.

## **Publicaciones bajo punto de sincronismo**

Los editores pueden emitir llamadas MQPUT o MQPUT1 en el punto de sincronismo para incluir todos los mensajes entregados a los suscriptores en una unidad de trabajo. Si se ha especificado la opción MQPMO\_RETAIN, o las opciones de entrega de temas NPMSGDLV o PMSGDLV con los valores ALL o ALLDUR, el gestor de colas utiliza llamadas MQPUT o MQPUT1 internas en el punto de sincronismo, dentro del ámbito de la llamada MQPUT o MQPUT1 de el publicador.

### ***Información de estado y de sucesos***

Las publicaciones se pueden clasificar como publicaciones de estado tales como, por ejemplo, el precio actual de una acción o como publicaciones de sucesos tales como, por ejemplo, el intercambio de dicha acción.

## **Publicaciones de estado**

Las *publicaciones de estado* contienen información sobre el estado actual de algo como, por ejemplo, el precio de una acción o el resultado actual de un partido de fútbol. Cuando sucede algo (por ejemplo, el precio de la acción cambia o el resultado del partido de fútbol cambia), la información de estado anterior ya no es necesaria porque se reemplaza por la nueva información.

Un suscriptor desea recibir la versión actual de la información de estado al iniciarse, y que se le envíe nueva información siempre que el estado cambie.

Si una publicación contiene información de estado, se publica frecuentemente como una publicación retenida. Normalmente un nuevo suscriptor desea la información de estado actual inmediatamente; el suscriptor no desea esperar a que el suceso causante de la información vuelva a publicarse.

Los suscriptores recibirán automáticamente la publicación retenida de un tema cuando se suscriban, a menos que el suscriptor utilice las opciones MQSO\_PUBLICATIONS\_ON\_REQUEST o MQSO\_NEW\_PUBLICATIONS\_ONLY.

## **Publicaciones de sucesos**

Las *publicaciones de sucesos* contienen información sobre sucesos específicos que se producen, como intercambiar acciones o marcar un gol. Cada suceso es independiente de los demás.

Un suscriptor desea recibir información sobre sucesos a medida que éstos se producen.

### ***Publicaciones retenidas***

De forma predeterminada, después de enviar una publicación a todos los suscriptores interesados, se descarta. Sin embargo, un publicador puede especificar que una copia de una publicación está retenida para que se pueda enviar a suscriptores futuros que estén interesados en el tema.

La supresión de publicaciones después de haberlas enviado a todos los suscriptores interesados es adecuada para la información de sucesos, pero no siempre es adecuada para la información de estado. Al retener un mensaje, los nuevos suscriptores no tienen que esperar a que se publique la información de nuevo antes de recibir información de estado inicial. Por ejemplo, un suscriptor con una suscripción al precio de una acción recibirá el precio actual inmediatamente, sin tener que esperar a que el precio de la acción cambie (y, por consiguiente, vuelva a publicarse).

El gestor de colas puede retener sólo una publicación para cada tema, por lo que la publicación retenida existente de un tema se suprime cuando una nueva publicación retenida llega al gestor de colas. Sin embargo, la supresión de la publicación existente no puede producirse de forma síncrona con la llegada de la nueva publicación retenida. Por lo tanto, siempre que sea posible, no debe tener más de un publicador enviando publicaciones retenidas sobre cualquier tema.

Los suscriptores pueden especificar que no desean recibir publicaciones retenidas mediante la opción de suscripción MQSO\_NEW\_PUBLICATIONS\_ONLY. Los suscriptores existentes pueden solicitar que les sean enviadas copias duplicadas de publicaciones retenidas.

Hay ocasiones en que tal vez no desee retener publicaciones, ni siquiera para la información de estado:

- Si todas las suscripciones a un tema se realizan antes de haber efectuado ninguna publicación sobre ese tema, y no espera o no permite nuevas suscripciones, no es necesario retener publicaciones porque se entregan al conjunto completo de los suscriptores la primera vez que se publiquen.
- Si las publicaciones se producen con frecuencia como, por ejemplo, cada segundo, un nuevo suscriptor (o un suscriptor se recupere de una anomalía) recibe el estado actual casi inmediatamente después de su suscripción inicial, por lo que no es necesario retener estas publicaciones.
- Si las publicaciones son grandes, puede terminar necesitando una cantidad considerable de espacio de almacenamiento para almacenar la publicación retenida para cada tema. En un entorno de varios gestores de colas, las publicaciones retenidas las almacenan todos los gestores de colas de la red que tienen una suscripción coincidente.

A la hora de decidir si se utilizan publicaciones retenidas, considere cómo las aplicaciones suscriptoras pueden recuperarse de una anomalía. Si el publicador no utiliza publicaciones retenidas, la aplicación suscriptora podría tener que almacenar localmente su estado actual.

Para asegurarse de que una publicación está retenida, utilice la opción de colocar mensaje MQPMO\_RETAIN. Si se utiliza esta opción y la publicación no se puede retener, el mensaje no se publica y la llamada falla con MQRC\_PUT\_NOT\_RETAINED.

Si un mensaje es una publicación retenida, esto se indica mediante la propiedad de mensaje MQIsRetained. La persistencia de un mensaje es la que era cuando fue publicado originalmente.

### **Conceptos relacionados**

Consideraciones de diseño acerca de las publicaciones retenidas en los clústeres de publicación/suscripción

### ***Publicaciones bajo punto de sincronismo***

En la publicación/suscripción de IBM MQ, el punto de sincronización lo pueden utilizar los publicadores o internamente el gestor de colas.

Los publicadores de sincronización al utilizar MQPUT/MQPUT1 pueden emitir llamadas con la opción MQPMO\_SYNCPOINT. Todos los mensajes entregados a los suscriptores se cuentan hasta el número máximo de mensajes no confirmados en una unidad de trabajo. El atributo de gestor de colas MAXUMSGS especifica este límite. Si se alcanza el límite, el publicador recibe el código de razón 2024 (07E8) (RC2024): MQRC\_SYNCPOINT\_LIMIT\_LLEGED.

Cuando un publicador emite llamadas MQPUT/MQPUT1 mediante MQPMO\_NO\_SYNCPOINT con la opción MQPMO\_RETAIN, o las opciones de entrega NPMGDLV/PMSGDLV con los valores ALL o ALLDUR, el gestor de colas utiliza puntos de sincronismo internos para garantizar que los mensajes se entreguen según se ha solicitado. El publicador puede recibir el código de razón 2024 (07E8) (RC2024): MQRC\_SYNCPOINT\_LIMIT\_LLEGED si se alcanza el límite dentro del ámbito de la llamada del publicador MQPUT/MQPUT1.

## **Suscriptores y suscripciones**

En la publicación/suscripción de IBM MQ, un suscriptor es una aplicación que solicita información sobre un tema específico a un gestor de colas en una red de publicación/suscripción. Un suscriptor puede recibir mensajes, sobre los mismos temas o temas distintos, de más de un publicador.

Las suscripciones pueden crearse manualmente utilizando un mandato MQSC o por las aplicaciones. Estas suscripciones se emiten para el gestor de colas local y contienen información acerca de las publicaciones que el suscriptor desea recibir:

- El tema en el que está interesado el suscriptor; esto puede resolverse en varios temas si se utilizan comodines.
- Se debe aplicar una serie de selección opcional para los mensajes publicados.
- Un manejador en una cola (conocida como *cola de suscriptores*), en la que deben colocarse las publicaciones seleccionadas y el CorrelId opcional.

El gestor de colas local almacena información de suscripción y cuando recibe una publicación, explora la información para determinar si existe una suscripción que coincida con el tema de la publicación y la serie de selección. Para cada suscripción coincidente, el gestor de colas dirige la publicación a la cola de suscriptores del suscriptor. La información que un gestor de colas almacena sobre las suscripciones se puede visualizar utilizando los mandatos DIS SUB y DIS SBSTATUS.

Una suscripción se suprime sólo cuando se produce uno de los siguientes sucesos:

- El suscriptor cancela la suscripción mediante la llamada MQCLOSE (si la suscripción se realizó como no duradera).
- La suscripción caduca.
- El administrador del sistema suprime la suscripción mediante el mandato DELETE SUB.
- La aplicación suscriptora finaliza (si la suscripción se realizó como no duradera).
- El gestor de colas se detiene o se reinicia (si la suscripción se realizó como no duradera).

Cuando obtenga mensajes, utilice las opciones adecuadas en la llamada MQGET. Si su aplicación solo procesa mensajes para una suscripción, como mínimo, debe utilizar `get-by-correlid`, como se muestra en el programa C de ejemplo `amqssbxa.c` y en el [suscriptor MQ no gestionado](#). El **CorrelId** que se va a utilizar se devuelve de MQSUB en el MQSD.Campo **SubCorrelId**.

### **Conceptos relacionados**

[Suscripciones clonadas y compartidas](#)

### **Referencia relacionada**

[Ejemplos de cómo definir la propiedad sharedSubscription](#)

## ***Colas gestionadas y publicación/suscripción***

Al crear una suscripción, puede elegir utilizar colas gestionadas. Si utiliza colas gestionadas, una cola de suscripciones se crea automáticamente al crear una suscripción. Las colas gestionadas se ordenan automáticamente según la durabilidad de la suscripción. La utilización de colas gestionadas significa que no tiene que preocuparse de la creación de las colas para recibir publicaciones y que las publicaciones no consumidas se eliminan de las colas de suscriptores automáticamente si se cierra una conexión de suscripción no duradera.

Si una aplicación no necesita utilizar una cola en concreto como su cola suscriptor, el destino de las publicaciones que recibe, puede hacer uso de las *suscripciones gestionadas* mediante la opción de suscripción MQSO\_MANAGED. Si crea una suscripción gestionada, el gestor de colas devuelve un manejador de objetos al suscriptor para una cola de suscriptores que el gestor de colas crea donde se recibirán publicaciones. Esto se debe a que una *suscripción gestionada* es aquella en la que IBM MQ maneja la suscripción. Se devolverá el manejador de objetos de la cola permitiéndole examinar, obtener o consultar en la cola (no es posible transferir o establecer atributos de una cola gestionada a menos que se dé explícitamente acceso a las colas dinámicas temporales).

La durabilidad de la suscripción determina si la cola gestionada permanece cuando se interrumpe la conexión de la aplicación suscriptora con el gestor de colas.

Las suscripciones gestionadas son especialmente útiles cuando se utilizan con las suscripciones no duraderas porque, cuando la conexión de la aplicación finaliza, los mensajes no consumidos permanecerán en la cola de suscriptores ocupando espacio en el gestor de colas indefinidamente. Si está utilizando una suscripción gestionada, la cola gestionada será una cola dinámica temporal y como tal se suprimirá junto con los mensajes no consumidos cuando se interrumpa la conexión por alguna de las razones siguientes:

- Se utiliza MQCLOSE con MQCO\_REMOVE\_SUB y se cierra el Hobj gestionado.
- Se pierde una conexión con una aplicación utilizando una suscripción no duradera (MQSO\_NON\_DURABLE).
- Se elimina una suscripción debido a que ha caducado y se ha cerrado el Hobj gestionado.

También se pueden utilizar suscripciones gestionadas con suscripciones duraderas pero es posible que desee dejar mensajes no consumidos en la cola de suscriptores para que se puedan recuperar cuando se

reabre la conexión. Por este motivo, las colas gestionadas para suscripciones duraderas toman la forma de una cola dinámica permanente y permanecerá cuando se interrumpa la conexión de la aplicación suscriptor con el gestor de colas.

Puede establecer una caducidad en la suscripción si desea utilizar una cola gestionada dinámica permanente, de forma que aunque la cola permanezca existiendo tras rota la conexión, no seguirá existiendo de forma indefinida.

Si se suprime la cola gestionada recibirá un mensaje de error.

Las colas gestionadas que se crean se denominan con números al final (indicaciones horarias) para que cada una sea exclusiva.

### ***Durabilidad de suscripción***

Las suscripciones pueden configurarse para ser duraderas o no duraderas. La perdurabilidad de la suscripción determina qué sucede con las suscripciones cuando las aplicaciones suscriptoras se desconectan de un gestor de colas.

### **Suscripciones duraderas**

Las suscripciones duraderas siguen existiendo cuando se cierra la conexión con el gestor de colas de la aplicación de suscripción. Si una suscripción es duradera, cuando la aplicación de suscripción se desconecta, la suscripción permanece en su lugar y la puede utilizar la aplicación de suscripción cuando se vuelva a conectar solicitando la suscripción de nuevo utilizando el **SubName** que se devolvió cuando se creó la suscripción.

Cuando se suscribe de forma duradera, un nombre de suscripción ( **SubName** ) es obligatorio. Los nombres de suscripción deben ser exclusivos en un gestor de colas de modo que pueda utilizarse para identificar una suscripción. Este medio de identificación es necesario al especificar una suscripción que se desea reanudar, si ha cerrado deliberadamente la conexión con la suscripción (utilizando la opción MQCO\_KEEP\_SUB) o si se ha desconectado del gestor de colas. Puede reanudar una suscripción existente utilizando la llamada MQSUB con la opción MQSO\_RESUME. Los nombres de suscripción también se visualizan si utiliza el mandato DISPLAY SBSTATUS con SUBTYPE ALL o ADMIN.

Cuando una aplicación ya no requiere una suscripción duradera, puede eliminarse utilizando la llamada de función MQCLOSE con la opción MQCO\_REMOVE\_SUB o se puede suprimir manualmente mediante el mandato MQSC DELETE SUB.

Puede utilizar el atributo de tema **DURSUB** para especificar si las suscripciones duraderas pueden realizarse en un tema o no.

En la devolución de una llamada MQSUB mediante la opción MQSO\_RESUME, la caducidad de la suscripción se ha establecido en la caducidad original de la suscripción y no en la hora de caducidad restante.

Un gestor de colas continúa enviando publicaciones para satisfacer una suscripción duradera aunque la aplicación de suscriptor no esté conectada. Esto conduce a una acumulación de mensajes en la cola de suscriptores. La forma más sencilla de evitar este problema es utilizar una suscripción no duradera cuando sea adecuado. No obstante, donde sea necesario utilizar suscripciones duraderas, puede evitarse una acumulación de mensajes si el suscriptor se suscribe utilizando la opción Publicaciones retenidas. Un suscriptor puede controlar cuándo recibe publicaciones utilizando la llamada MQSUBRQ.

### **Suscripciones no duraderas**

Las suscripciones no duraderas sólo existen mientras permanece abierta la conexión de la aplicación suscriptor al gestor de colas. La suscripción se elimina cuando la aplicación suscriptor se desconecta del gestor de colas deliberadamente o por pérdida de la conexión. Cuando se cierra la conexión, la información sobre la suscripción se elimina del gestor de colas y ya no se muestran suscripciones mediante el mandato DISPLAY SBSTATUS. No se envían más mensajes a la cola de suscriptores.

Lo que le sucede con cualquier publicación no consumida en la cola de suscriptores para las suscripciones no duraderas se determina de la manera siguiente.

- Si una aplicación de suscripción utiliza un destino gestionado, las publicaciones que no se hayan consumido se eliminan automáticamente.
- Si la aplicación suscriptora proporciona un manejador para su propia cola de suscriptores cuando se suscribe, los mensajes no consumidos no se eliminan automáticamente. Es responsabilidad de la aplicación borrar la cola si es lo pertinente. Si la cola es compartida por más de un suscriptor u otras aplicaciones punto a punto, puede que no sea adecuado borrar la cola por completo.

Aunque no es necesario para las suscripciones no duraderas, el gestor de colas utiliza un nombre de suscripción si se ha proporcionado uno. Los nombres de suscripción deben ser exclusivos en el gestor de colas de modo que pueda utilizarse para identificar una suscripción.

### **Conceptos relacionados**

Suscripciones clonadas y compartidas

### **Tareas relacionadas**

Utilización de suscripciones compartidas de JMS 2.0

### **Referencia relacionada**

Ejemplos de cómo definir la propiedad `sharedSubscription`

### ***Series de selección***

Una *serie de selección* es una expresión que se aplica a una publicación para determinar si coincide con una suscripción. Las series de selección pueden incluir caracteres comodín.

Al suscribirse, además de especificar un tema, puede especificar una serie de selección para seleccionar publicaciones en función de sus propiedades de mensaje.

La serie de selección se evalúa contra el mensaje como transferidos por el publicador antes de que se modifique para entregar a cada suscriptor. Tenga en cuenta que al usar campos en la serie de selección, es posible que se modifiquen como parte de la operación de publicación. Por ejemplo, los campos MQMD `UserIdentifier`, `MsgIdy CorrelId`.

Las series de selección no deben hacer referencia a ninguno de los campos de propiedad del mensaje añadidos por el gestor de colas como parte de la operación de publicación (consulte Propiedades del mensaje de publicación/suscripción), excepto para la propiedad del mensaje `MQTopicString`, que contiene la serie del tema para la publicación.

### **Conceptos relacionados**

Restricciones y reglas de series de selección

## **Temas**

Un tema es el asunto de la información que se publica en un mensaje de publicación/suscripción.

Los mensajes en los sistemas de punto a punto se envían a una dirección de destino específica. Los mensajes de sistemas de publicación/suscripción basados en el asunto se envían a los suscriptores basándose en el asunto que describe el contenido del mensaje. En los sistemas basados en contenido, los mensajes se envían a los suscriptores basándose en el contenido del mensaje propiamente dicho.

El sistema de publicación/suscripción de IBM MQ es un sistema de publicación/suscripción basado en temas. Un publicador crea un mensaje y lo publica con una serie de tema que se ajusta mejor al asunto de la publicación. Para recibir publicaciones, un suscriptor crea una suscripción con una serie de tema que coincide con un patrón para seleccionar temas de publicación. El gestor de colas entrega publicaciones a los suscriptores que tienen suscripciones que coinciden con el tema de publicación y tienen autorización para recibir las publicaciones. El artículo “Series de tema” en la página 70 describe la sintaxis de series de temas que identifican el asunto de una publicación. Los suscriptores también crean series de temas para seleccionar los temas que se van a recibir. Las series de temas que los suscriptores crean pueden contener uno de los dos esquemas comodín alternativos que coincida con el patrón con las series de temas en las publicaciones. La coincidencia de patrón se describe en “Esquemas de comodín” en la página 71.

En la publicación/suscripción basada en el asunto, los publicadores o administradores son responsables de clasificar los asuntos en temas. Los asuntos se organizan normalmente de forma jerárquica, en árboles de temas, utilizando el carácter ' / ' para crear subtemas en la serie de tema. Consulte [“Árboles de temas”](#) en la página 77 para ver ejemplos de árboles de temas. Los temas son nodos del árbol de temas. Los temas pueden ser nodos hoja sin subtemas, o nodos intermedios con subtemas.

En paralelo con la organización de asuntos en un árbol de temas jerárquico, puede asociar los temas con objetos de temas administrativos. Asigne atributos a un tema, como si el tema es distribuido en un clúster, asociándolo con un objeto de tema administrativo. La asociación se realiza dando nombre al tema utilizando el atributo TOPICSTR del objeto de tema administrativo. Si no asocia explícitamente un objeto de tema administrativo a un tema, el tema hereda los atributos de su antecesor más cercano en el árbol de temas que *tenga* asociado a un objeto de tema administrativo. Si no ha definido ningún padre, lo hereda de SYSTEM.BASE.TOPIC. Los objetos de tema administrativo se describen en [“Objetos de tema administrativo”](#) en la página 78.

**Nota:** Aunque herede todos los atributos de un tema de SYSTEM.BASE.TOPIC, defina un tema raíz para los temas que se hereda directamente de SYSTEM.BASE.TOPIC. Por ejemplo, en el espacio de temas de los estados de EE.UU., USA/Alabama USA/Alaska, etc., USA es el tema raíz. La finalidad principal del tema raíz es crear espacios de tema discretos y no solapados para evitar que las publicaciones coincidan con las suscripciones erróneas. También significa que puede cambiar los atributos del tema raíz para afectar a todo el espacio de temas. Por ejemplo, puede establecer el nombre del atributo **CLUSTER**.

Cuando hace referencia a un tema como un publicador o un suscriptor, tiene la opción de suministrar una serie de tema o hacer referencia a un objeto de tema. O bien puede hacer ambas cosas, en cuyo caso la serie de tema que suministre define un subtema del objeto de tema. El gestor de colas identifica el tema añadiendo la serie de tema al prefijo de serie de tema especificado en el objeto de tema, insertando un ' / ' adicional entre las dos series de tema, por ejemplo, *serie de tema/serie de objeto*. [“Combinación de series de tema”](#) en la página 75 lo describe de forma más detallada. La serie de tema resultante se utiliza para identificar el tema y asociarlo con un objeto de tema administrativo. El objeto de tema administrativo no es necesariamente el mismo objeto de tema que el objeto de tema correspondiente al tema maestro.

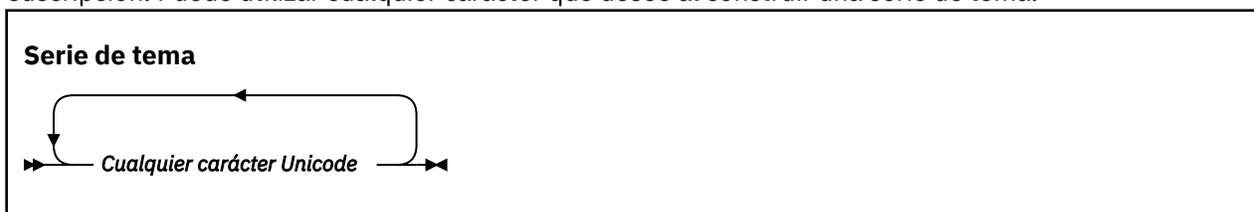
En la publicación/suscripción basada en contenido, defina qué mensajes desea recibir proporcionando series de selección que busquen en el contenido de todos los mensajes. IBM MQ ofrece una forma intermedia de publicación/suscripción basada en contenidos utilizando selectores de mensajes que exploran propiedades de mensajes en vez del contenido completo del mensaje; consulte [Selectores](#). El uso arquetípico de selectores de mensajes es suscribirse a un tema y, a continuación, calificar la selección con una propiedad numérica. El selector le permite especificar que está interesado en los valores sólo en un rango determinado; algo que no se puede hacer con comodines basados en caracteres o en temas. Si tiene que filtrar según el contenido completo del mensaje, debe utilizar IBM Integration Bus.

## Series de tema

Etiquetar información que se publica como tema utilizando una serie de tema. Suscríbase a grupos de temas utilizando series de temas comodín basados en caracteres o temas.

## Temas

Una *serie de tema* es una serie de caracteres que identifica el tema de un mensaje de publicación/suscripción. Puede utilizar cualquier carácter que desee al construir una serie de tema.



Tres caracteres tienen un significado especial en la publicación/suscripción de IBM WebSphere MQ 7. Están permitidos en cualquier lugar de una serie de tema, pero debe utilizarlos con precaución. El uso de los caracteres especiales se explica en [“Esquema de comodín basado en temas”](#) en la página 71.

## Una barra inclinada (/)

El separador de nivel de tema. Utilice el carácter '/' para estructurar el tema en un árbol de temas.

Evite niveles de temas vacíos, '//', si puede. Corresponden a nodos en la jerarquía de temas sin series de temas. Un '/' inicial o final en una serie de tema corresponde a un nodo vacío inicial o final y también debe evitarse.

## El signo de hash (#)

Se utiliza en combinación con '/' para construir un carácter comodín multinivel en las suscripciones. Tenga cuidado al utilizar '#' junto a '/' en series de temas utilizadas para referirse a temas publicados. “Ejemplos de series de temas” en la página 71 muestra un uso racional de '#'.

Las series '.../#/...', '#/...' y '.../#' tienen un significado especial en las series de temas de suscripción. Las series coinciden con todos los temas en uno o varios niveles de la jerarquía de temas. Por lo tanto, si ha creado un tema con una de esas secuencias, no podría suscribirse a él sin tener que suscribirse también a todos los temas en varios niveles de la jerarquía de temas.

## El signo más (+)

Se utiliza en combinación con '/' para construir un carácter comodín de un solo nivel en las suscripciones. Tenga cuidado al utilizar '+' junto a '/' en series de temas utilizadas para referirse a temas publicados.

Las series '.../+/...', '+/...' y '.../+' tienen un significado especial en las series de temas de suscripción. Las cadenas coinciden con todos los temas en un nivel de la jerarquía de temas. Por lo tanto, si ha creado un tema con una de esas secuencias, no podría suscribirse a él sin tener que suscribirse también a todos los temas en un nivel de la jerarquía de temas.

## Ejemplos de series de temas

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

## Referencia relacionada

### TOPIC

#### *Esquemas de comodín*

Existen dos esquemas comodín que se utilizan para suscribirse a varios temas. La elección del esquema es una opción de suscripción.

#### **MQSO\_WILDCARD\_TOPIC**

Seleccione los temas a los que se va a suscribir mediante el esquema comodín basado en temas.

Este es el valor predeterminado si no se selecciona explícitamente ningún esquema comodín.

#### **MQSO\_WILDCARD\_CHAR**

Seleccione los temas a los que se va a suscribir mediante el esquema comodín basado en caracteres.

Establezca el esquema especificando el parámetro **wschema** en el mandato DEFINE SUB. Para obtener más información, consulte [DEFINE SUB](#).

**Nota:** Las suscripciones que se han creado antes de IBM WebSphere MQ 7.0 siempre utilizan el esquema comodín basado en caracteres.

## Ejemplos

```
IBM+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

#### *Esquema de comodín basado en temas*

Los comodines basados en temas permiten a los suscriptores suscribirse a más de un tema a la vez.

Los comodines basados en temas son una característica potente del sistema de temas en la publicación/suscripción de IBM MQ. El comodín multinivel y el comodín de un solo nivel pueden utilizarse para suscripciones, pero el publicador de un mensaje no puede utilizarlos dentro de un tema.

El esquema de comodín basado en temas le permite seleccionar publicaciones agrupadas por nivel de tema. En cada nivel de la jerarquía de temas, se puede elegir si la cadena de la suscripción del nivel de tema tiene que coincidir exactamente con la cadena de la publicación o no. Por ejemplo, la suscripción IBM/+/Results selecciona todos los temas,

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

Hay dos tipos de comodines.

### Comodín multinivel

- El comodín multinivel se utiliza en suscripciones. Cuando se utiliza en una publicación se trata como un literal.
- El carácter comodín multinivel '#' se utiliza para buscar coincidencias con cualquier número de niveles dentro de un tema. Por ejemplo, al utilizar el árbol de temas de ejemplo, si se suscribe a 'USA/Alaska/#', recibirá mensajes sobre los temas 'USA/Alaska' y 'USA/Alaska/Juneau'.
- El comodín multinivel puede representar cero o más niveles. Por consiguiente, 'USA/#' también puede coincidir con el singular 'USA', donde '#' representa cero niveles. El separador de nivel de tema no tiene sentido en este contexto ya que no hay ningún nivel para separar.
- El comodín multinivel sólo es eficaz cuando se especifica solo o junto al carácter separador de nivel de tema. Por lo tanto, '#' y 'USA/#' son temas válidos cuando el carácter '#' se trata como un comodín. Sin embargo, aunque 'USA#' también es una serie de tema válida, el carácter '#' no se considera un comodín y no tiene ningún significado especial. Para obtener más información, consulte [“Cuando los comodines basados en temas no son tales”](#) en la página 74.

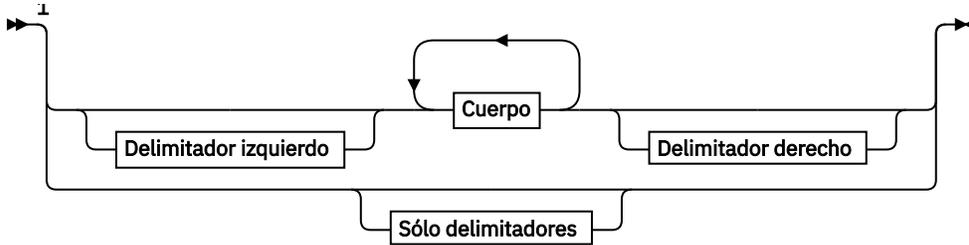
### Comodín de un solo nivel

- El comodín de un solo nivel se utiliza en suscripciones. Cuando se utiliza en una publicación se trata como un literal.
- El carácter comodín de un solo nivel '+' coincide con un nivel de tema, y sólo uno. Por ejemplo, 'USA/+' coincide con 'USA/Alabama', pero no con 'USA/Alabama/Auburn'. Dado que el comodín de un solo nivel coincide con un solo nivel, 'USA/+' no coincide con 'USA'.
- El comodín de un solo nivel puede utilizarse en cualquier nivel del árbol de temas, y junto con el comodín multinivel. El comodín de un solo nivel debe especificarse junto al separador de nivel de tema, excepto en el caso de que se especifique solo. Por lo tanto, '+' y 'USA/+' son temas válidos cuando el carácter '+' se trata como un comodín. Sin embargo, aunque 'USA+' también es una serie de tema válida, el carácter '+' no se considera un comodín y no tiene ningún significado especial. Para obtener más información, consulte [“Cuando los comodines basados en temas no son tales”](#) en la página 74.

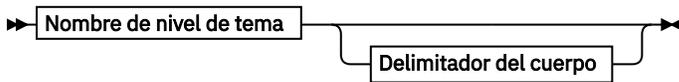
La sintaxis del esquema de comodín basado en temas no tiene caracteres de escape. Si '#' y '+' se tratan como comodines o no, depende de su contexto. Consulte [“Cuando los comodines basados en temas no son tales”](#) en la página 74 para obtener más información.

**Nota:** El inicio y el fin de una serie de tema se tratan de una manera especial. Utilizando '\$' para indicar el final de la serie, '\$#/...' es un comodín multinivel y '\$/#/...' es un nodo vacío en la raíz, seguido de un comodín multinivel.

### Serie de comodín basado en temas



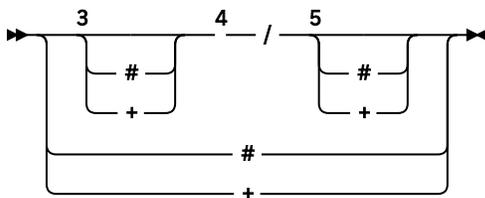
### Cuerpo



### Nombre de nivel de tema

► Cualquier carácter Unicode, excepto / <sup>2</sup> ►

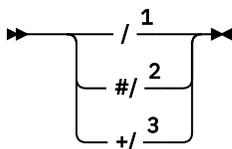
### Sólo delimitadores



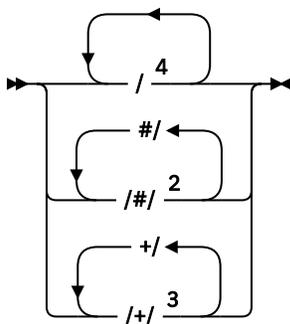
### Notas:

- <sup>1</sup> Una serie de tema de longitud cero o nula no es válida
- <sup>2</sup> Se recomienda no utilizar ninguno de los \*, ?, % en las series de nombre de nivel para la compatibilidad entre esquemas de comodín basados en caracteres y basados en temas.
- <sup>3</sup> Estos casos son equivalentes al patrón de *delimitador izquierdo*.
- <sup>4</sup> / sin comodines coincide con un solo tema vacío.
- <sup>5</sup> Estos casos son equivalentes al patrón de *delimitador derecho*.
- <sup>6</sup> Coincide con todos los temas.
- <sup>7</sup> Coincide con todos los temas en los que sólo hay un nivel.

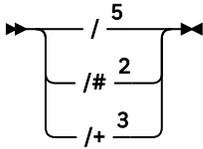
### Delimitador izquierdo



### Delimitador de cuerpo



## Delimitador derecho



### Notas:

- <sup>1</sup> La serie de tema empieza con un tema vacío
- <sup>2</sup> Coincide con cero o más niveles. Varias series de coincidencias multinivel tienen el mismo efecto que una serie de coincidencias multinivel.
- <sup>3</sup> Coincide exactamente con un nivel.
- <sup>4</sup> // es un tema vacío: un objeto de tema sin serie de tema.
- <sup>5</sup> La serie de tema finaliza con un tema vacío

## Cuando los comodines basados en temas no son tales

Los caracteres comodín '+' y '#' no tienen ningún significado especial cuando están mezclados con otros caracteres (incluidos ellos mismos) en un nivel de tema.

Esto significa que se pueden publicar los temas que contienen '+' o '#' junto con otros caracteres en un nivel de tema.

Por ejemplo, considere los dos temas siguientes:

1. level0/level1/+/level4/#
2. level0/level1/#+/level4/level#

En el primer ejemplo, los caracteres '+' y '#' se tratan como comodines y, por tanto, no son válidos en una serie de tema en el que debe publicarse, pero son válidos en una suscripción.

En el segundo ejemplo, los caracteres '+' y '#' no se tratan como comodines y, por lo tanto, la serie de tema puede publicarse y es posible suscribirse a ella.

## Ejemplos

```
IBM/+/Results
#/Results
IBM/Software/Results
```

### Esquema de comodín basado en caracteres

El esquema de comodín basado en caracteres permite seleccionar temas en base a la coincidencia de caracteres tradicional.

Puede seleccionar todos los temas en varios niveles de una jerarquía de temas utilizando la serie '\*'. Utilizar '\*' en el esquema de comodín basado en temas es equivalente a utilizar la serie de comodín basada en temas '#'

'x\*/y' equivale a 'x#/y' en el esquema basado en temas, y selecciona todos los temas de la jerarquía de temas entre los niveles "x" y "y", donde "x" y "y" son nombres de temas que no están en el conjunto de niveles devueltos por el comodín.

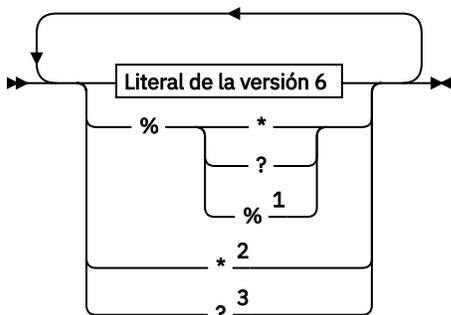
'/+' en el esquema basado en temas no tiene un equivalente exacto en el esquema basado en caracteres. 'IBM\*/Results' también seleccionaría 'IBM/Patents/Software/Results'. Puede construir siempre consultas con los dos esquemas que dan lugar a coincidencias exactas sólo si el conjunto de nombres de tema en cada nivel de la jerarquía es exclusivo.

Utilizados de forma general, '\*' y '?' en el esquema basado en caracteres no tienen equivalente en el esquema basado en temas. El esquema basado en temas no realiza la coincidencia parcial

mediante comodines. La suscripción de comodines basada en caracteres 'IBM/\*ware/Results' no tiene equivalente basado en temas.

**Nota:** Las coincidencias que utilizan suscripciones de comodines basadas en caracteres son más lentas que las coincidencias que utilizan suscripciones basadas en temas.

### Serie de comodines basada en caracteres



### Literal de la versión 6

►► Cualquier carácter excepto \*,? y % ◄◄

Notas:

- <sup>1</sup> Significa escapar el siguiente carácter, para tratarlo como un literal. '%' debe ir seguido de "'\*', " '?' o "'%'. Consulte "Ejemplos de series de temas" en la página 71.
- <sup>2</sup> Significa la coincidencia con ninguno o más caracteres en una suscripción.
- <sup>3</sup> Significa la coincidencia exacta de un carácter en una suscripción.

### Ejemplos

```
IBM/*/Results
IBM/*ware/Results
```

### Combinación de series de tema

Cuando se crean suscripciones o se abren temas para que pueda publicar mensajes en ellos, la serie de tema se puede formar combinando dos series de subtema o "subtemas" separados. La aplicación o el mandato administrativo proporcionan un subtema como serie de tema y el otro es la serie de tema asociada a un objeto de tema. Puede utilizar cualquiera de los dos subtemas como tema de serie único o bien combinarlos para formar un nombre de tema nuevo.

Por ejemplo, cuando se define una suscripción utilizando el tema MQSC **DEFINE SUB**, el mandato puede adoptar como atributo **TOPICSTR** (serie de tema) o **TOPICOBJ** (objeto de tema) o bien ambos. Si sólo se proporciona **TOPICOBJ**, la serie de tema asociada a dicho objeto de tema se utiliza como serie de tema. Si sólo se proporciona **TOPICSTR**, se utiliza como serie de tema. Si se proporcionan ambos, se concatenan para formar una única serie de tema con el formato **TOPICOBJ / TOPICSTR**, donde la serie de tema **TOPICOBJ** configurada siempre es la primera y las dos partes de la serie siempre van separadas por un carácter "/".

Asimismo, en un programa MQI, el nombre de tema completo se crea mediante MQOPEN. Está compuesto por los campos utilizados en las llamadas de MQI de publicación/suscripción en el orden listado:

1. El atributo **TOPICSTR** del objeto de tema, nombrado en el campo **ObjectName**.
2. El parámetro **ObjectString** que define el subtema que la aplicación suministra.

La serie de tema resultante se devuelve en el parámetro **ResObjectString**.

Se considera que estos campos están presentes si el primer carácter de cada campo no es un valor en blanco o un carácter nulo y la longitud de campo es mayor que cero. Si sólo uno de los campos está

presente, se utiliza sin modificaciones como nombre del tema. Si ninguno de los campos tiene un valor, la llamada no se ejecuta correctamente y genera el código de razón MQRC\_UNKOWN\_OBJECT\_NAME o MQRC\_TOPIC\_STRING\_ERROR si el nombre de tema completo no es válido.

Si ambos campos están presentes, se inserta un carácter "/" entre los dos elementos del nombre de tema combinado resultante.

La tabla siguiente muestra ejemplos de concatenación de series de tema:

<i>Tabla 2. Ejemplo de concatenación de series de tema</i>			
<b>TOPICSTR del objeto de tema</b>	<b>Serie de tema proporcionada por la aplicación o el mandato DEFINE SUB</b>	<b>Nombre de tema completo</b>	<b>Comentario</b>
Fútbol/Resultados	' '	Fútbol/Resultados	El TOPICSTR del objeto de tema se utiliza sin nada más.
' '	Fútbol/Resultados	Fútbol/Resultados	El ObjectString/ TOPICSTR del objeto de tema se utiliza sin nada más.
Fútbol	Resultados	Fútbol/Resultados	Se añade un carácter "/" en el punto de concatenación.
Fútbol	/Resultados	Fútbol//Resultados	Se produce un 'nodo vacío' entre las dos series. Esto es diferente de "Fútbol/Resultados".
/Fútbol	Resultados	/Fútbol/Resultados	El tema empieza por un 'nodo vacío'. Esto es diferente de "Fútbol/Resultados".

Se considera que el carácter "/" es un carácter especial que proporciona la estructura para el nombre de tema completo en ["Árboles de temas"](#) en la página 77. El carácter "/" no se debe utilizar con ninguna otra finalidad, sino la estructura del árbol de tema se verá afectada. El tema "/Football" no es el mismo que el tema "Football".

**Nota:** Si utiliza un objeto de tema al crear una suscripción, el valor de la serie de tema del objeto de tema se fija en la suscripción durante la definición. Los cambios posteriores en el objeto de tema no afectan a la serie de tema en la que se define la suscripción.

## Caracteres comodín en las series de tema

Los siguientes caracteres comodín son caracteres especiales:

- signo más (+)
- signo de número (#)
- asterisco (\*)
- signo de interrogación (?)

Los caracteres comodín sólo tienen un significado especial cuando se utilizan en una suscripción. Estos caracteres no se consideran como no válidos cuando se utilizan en otros lugares, sin embargo, debe asegurarse de que comprende cómo se utilizan y tal vez prefiera no utilizarlos en series de tema al publicar o definir objetos de tema.

Si publica en una serie de tema con # o + junto con otros caracteres (incluidos ellos mismos) dentro de un nivel de tema, la serie de tema se puede suscribir a cualquiera de los dos esquemas comodín.

Si publica en una serie de tema con # o + como únicos caracteres entre dos caracteres /, una aplicación no se podrá suscribir de forma explícita a una serie de tema utilizando el esquema de comodín MQSO\_WILDCARD\_TOPIC. El resultado de esta situación es que la aplicación obtiene más publicaciones de lo previsto.

No debe utilizar un carácter comodín en la serie de tema de un objeto de tema definido. Si lo hace, el carácter se tratará como un carácter literal cuando el objeto lo utilice un publicador y como carácter comodín cuando lo utilice una suscripción. Esto puede llevar a la confusión.

### Fragmento de código de ejemplo

Este fragmento de código, extraído del programa de ejemplo, [Ejemplo 2: Publicador para una serie de tema](#) combina un objeto de tema con una serie de tema variable:

```
MQOD td = {MQOD_DEFAULT}; /* Object Descriptor */
td.ObjectType = MQOT_TOPIC; /* Object is a topic */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
```

### Árboles de temas

Cada tema que se define es un elemento, o nodo, del árbol de temas. El árbol de temas puede estar vacío para empezar o puede contener temas que se hayan definido anteriormente utilizando mandatos MQSC o PCF. Puede definir un nuevo tema utilizando los mandatos crear tema o especificando el tema por primera vez en una publicación o suscripción.

Aunque puede utilizar cualquier serie de caracteres para definir la serie de tema de un tema, es aconsejable elegir una serie de tema que se ajuste a una estructura de árbol jerárquica. Un diseño cuidadoso de series de temas y árboles de temas puede ser de ayuda en las siguientes operaciones:

- Suscribirse a varios temas.
- Establecer políticas de seguridad.

Aunque puede construir un árbol de temas como una estructura plana lineal, es mejor crear un árbol de temas en una estructura jerárquica con uno o más temas raíz. Para obtener más información sobre la planificación de seguridad y los temas, consulte [Seguridad de publicación/suscripción](#).

La [Figura 18](#) en la [página 77](#) muestra un ejemplo de un árbol de temas con un tema raíz.

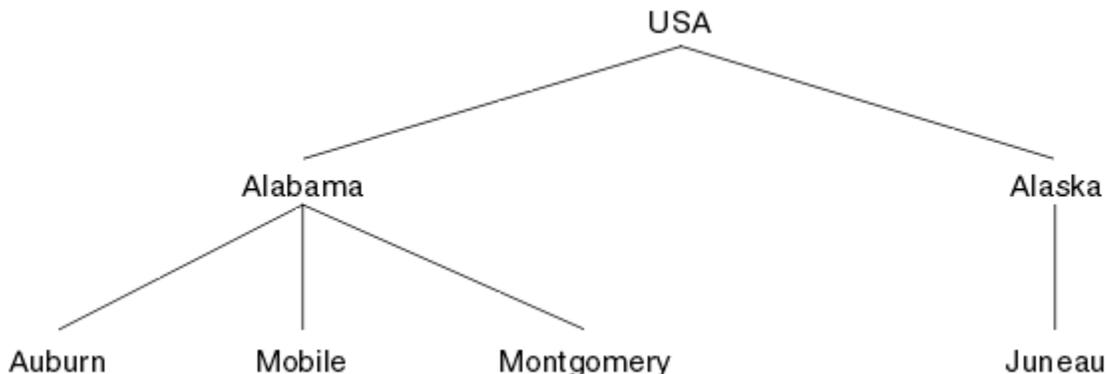


Figura 18. Ejemplo de un árbol de temas

Cada serie de caracteres de la figura representa un nodo en el árbol de temas. Una serie de tema completa se crea agregando nodos de uno o más niveles del árbol de temas. Los niveles se separan

mediante el carácter "/". El formato de una serie de tema especificada al completo es: "raíz/nivel2/nivel3".

Los temas válidos del árbol de temas mostrado en la [Figura 18](#) en la [página 77](#) son:

"USA"  
"USA/Alabama"  
"USA/Alaska"  
"USA/Alabama/Auburn"  
"USA/Alabama/Mobile"  
"USA/Alabama/Montgomery"  
"USA/Alaska/Juneau"

Cuando diseñe series de temas y árboles de temas, recuerde que el gestor de colas no interpreta la serie de tema propiamente dicha ni intenta deducir un significado de la misma. Simplemente utiliza la serie de tema para enviar mensajes seleccionados a los suscriptores de ese tema.

Los siguientes principios se aplican a la creación y el contenido de un árbol de temas:

- No hay ningún límite en cuanto al número de niveles de un árbol de temas.
- No hay ningún límite en cuanto a la longitud del nombre de un nivel de un árbol de temas.
- Puede haber un número ilimitado de nodos "raíz"; es decir, puede haber un número ilimitado de árboles de temas.

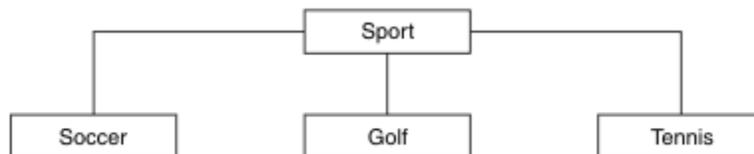
### Tareas relacionadas

[Reducción del número de temas no deseados en el árbol de temas](#)

### Objetos de tema administrativo

Mediante un objeto de tema administrativo puede asignar a los temas atributos específicos que no son predeterminados.

En la [Figura 19](#) en la [página 78](#) se muestra cómo un tema de alto nivel de Sport dividido en distintos temas que cubren distintos deportes puede visualizarse como árbol de temas:



*Figura 19. Visualización de un árbol de temas*

En la [Figura 20](#) en la [página 78](#) se muestra cómo el árbol de temas puede dividirse todavía más para separar distintos tipos de información sobre cada deporte:



*Figura 20. Árbol de temas ampliado*

Para crear el árbol de temas ilustrado, no es necesario definir ningún objeto de tema administrativo. Cada uno de los nodos de este árbol se define mediante una serie de tema creada en una operación de publicación o suscripción. Cada tema del árbol hereda los atributos de su padre. Los atributos se heredan del objeto de tema padre, porque, de forma predeterminada, todos los atributos están establecidos en

ASPARENT. En este ejemplo, cada tema tiene los mismos atributos que el tema Sport. El tema Sport no tiene objeto de tema de administración y hereda sus atributos desde `SYSTEM.BASE.TOPIC`.

Tenga en cuenta que no se recomienda otorgar autorizaciones a los usuarios no mqm en el nodo raíz del árbol de temas, que es `SYSTEM.BASE.TOPIC`, ya que las autorizaciones se heredan, pero no pueden restringirse. Por lo tanto, si se otorgan autorizaciones en este nivel, se otorgan autorizaciones a todo el árbol. Debe otorgar la autorización en un nivel de tema más abajo en la jerarquía.

Los objetos de temas administrativos pueden utilizarse para definir atributos específicos para nodos concretos en el árbol de temas. En el ejemplo siguiente, el objeto de tema administrativo se define para establecer la propiedad de suscripciones duraderas `DURSUB` del tema de fútbol en el valor `NO`:

```
DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
```

El árbol de temas ahora se puede visualizar como:

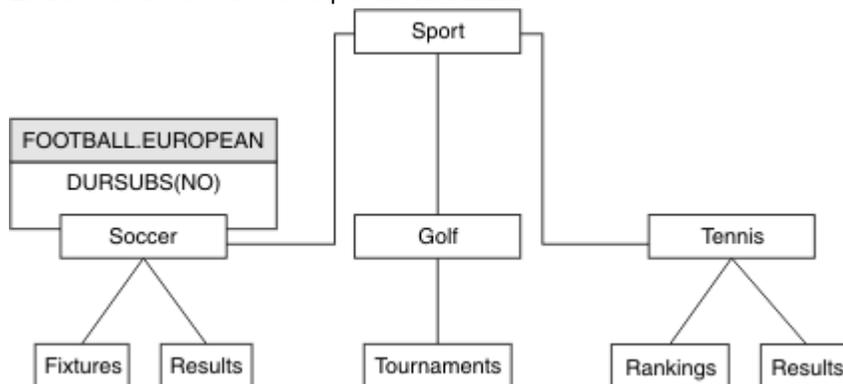


Figura 21. Visualización de un objeto de tema administrativo asociado con el tema Sport/Soccer

Las aplicaciones que se suscriban a temas por debajo de Soccer en el árbol pueden seguir utilizando las series de tema que utilizaban antes de añadir el objeto de tema administrativo. Sin embargo, ahora se puede escribir una aplicación para suscripción utilizando el nombre de objeto `FOOTBALL.EUROPEAN` en lugar de la serie `/Sport/Soccer`. Por ejemplo, para suscribir en `/Sport/Soccer/Results`, una aplicación puede especificar `MQSD.ObjectName` como `FOOTBALL.EUROPEAN` y `MQSD.ObjectString` como `Results`.

Con esta característica, puede ocultar una parte del árbol de temas ante los desarrolladores de aplicaciones. Defina un objeto de tema administrativo en un nodo determinado del árbol de temas, de modo que los desarrolladores de aplicaciones puedan definir sus propios temas como hijos del nodo. Los desarrolladores deben tener conocimiento sobre el tema padre, pero no sobre los demás nodos del árbol padre.

## Herencia de atributos

Si un árbol de temas tiene muchos objetos de temas administrativos, de forma predeterminada, cada objeto de tema administrativo hereda sus atributos de su tema administrativo padre más cercano. El ejemplo anterior se ha ampliado en [Figura 22 en la página 80](#):

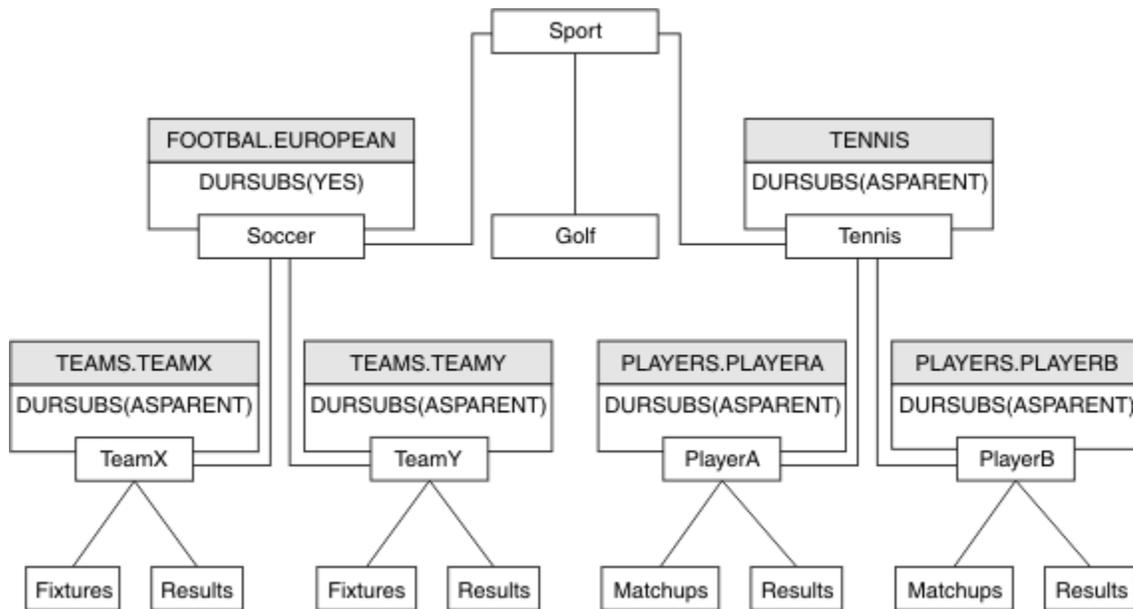


Figura 22. Árbol de temas con varios objetos de temas administrativos

Por ejemplo, utilice la herencia para asignar a todos los temas hijo de /Sport/Soccer la propiedad que establece que las suscripciones no son duraderas. Cambie el atributo DURSUB de FOOTBALL . EUROPEAN a NO.

Este atributo puede establecerse utilizando el siguiente mandato:

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Todos los objetos de tema administrativo de temas hijo de Sport/Soccer tienen la propiedad DURSUB establecida en el valor predeterminado ASPARENT. Después de cambiar el valor de la propiedad DURSUB de FOOTBALL . EUROPEAN a NO, los temas hijo de Sport/Soccer heredan el valor de la propiedad DURSUB NO. Todos los temas hijo de Sport/Tennis heredan el valor de DURSUB del objeto SYSTEM . BASE . TOPIC . SYSTEM . BASE . TOPIC tiene el valor YES.

Intentar crear una suscripción duradera para el tema Sport/Soccer/TeamX/Results ahora daría un error; no obstante, intentar realizar una suscripción duradera a Sport/Tennis/PlayerB/Results daría buen resultado.

## Control del uso de comodines con la propiedad WILDCARD

Utilice la propiedad WILDCARD de **Topic** de MQSC o la propiedad WildcardOperation de Topic de PFC equivalente para controlar la entrega de publicaciones a las aplicaciones de suscriptor que utilizan nombres de serie de tema comodín. La propiedad WILDCARD puede tener uno de los dos valores siguientes:

### WILDCARD

El comportamiento de las suscripciones comodín con respecto a este tema.

### PASSTHRU

Las suscripciones realizadas en un tema con comodines menos específico que la serie de tema en este objeto de tema reciben publicaciones creadas para este tema y para series de tema más específicas que este tema.

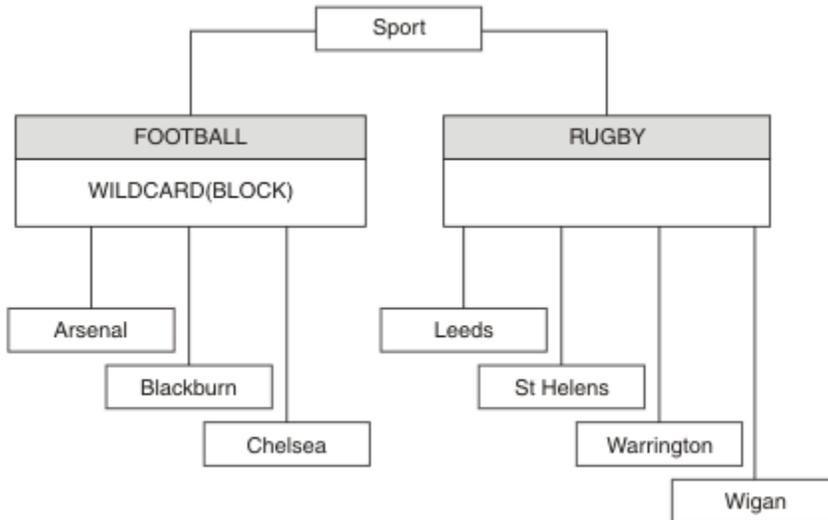
### BLOCK

Las suscripciones realizadas en un tema con comodín menos específico que la serie de tema en este objeto de tema no reciben publicaciones realizadas para este tema o para series de tema más específicas que este tema.

El valor de este atributo se utiliza cuando se definen las suscripciones. Si modifica este atributo, el conjunto de temas que abarcan las suscripciones existentes no se ve afectado por la modificación.

Este escenario también se aplica si se cambia la topología cuando se crean o suprimen objetos de tema; el conjunto de temas que coinciden con las suscripciones creadas después de la modificación del atributo WILDCARD se crea utilizando la topología modificada. Si desea forzar que el conjunto de temas coincidentes se vuelva a evaluar para las suscripciones existentes, debe reiniciar el gestor de colas.

En el ejemplo, “Ejemplo: crear el clúster de publicación/suscripción Sport” en la página 84, puede seguir los pasos para crear la estructura de árbol de temas mostrada en [Figura 23 en la página 81](#).



*Figura 23. Árbol de temas que utiliza la propiedad WILDCARD BLOCK*

Un suscriptor que utiliza la serie de tema comodín # recibe todas las publicaciones para el tema Sport y el subárbol Sport/Rugby. El suscriptor no recibe publicaciones en el subárbol Sport/Football, debido a que el valor de la propiedad WILDCARD del tema Sport/Football es BLOCK.

PASSTHRU es el valor predeterminado. Puede establecer la propiedad WILDCARD en el valor PASSTHRU en los nodos del árbol Sport. Si los nodos no tienen la propiedad WILDCARD con el valor BLOCK, definir PASSTHRU no altera el comportamiento observado por los suscriptores en los nodos del árbol Sports.

En el ejemplo, cree suscripciones para ver cómo el valor de comodín afecta a las publicaciones que se entregan; consulte [Figura 27 en la página 86](#). Ejecute el mandato de publicación de [Figura 30 en la página 87](#) para crear algunas publicaciones.

pub QMA

*Figura 24. Publicación en QMA*

Los resultados se muestran en [Tabla 3 en la página 81](#). Observe cómo el establecimiento de la propiedad WILDCARD en BLOCK impide que las suscripciones con comodines puedan recibir publicaciones para temas del ámbito del comodín.

<b>Suscripción</b>	<b>Serie de tema</b>	<b>Publicaciones recibidas</b>	<b>Notas</b>
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas las publicaciones del subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football

Tabla 3. Publicaciones recibidas en QMA (continuación)

Suscripción	Serie de tema	Publicaciones recibidas	Notas
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.

**Nota:**

Supongamos que una suscripción tiene un comodín que coincide con un objeto de tema con la propiedad WILDCARD con el valor BLOCK. Si la suscripción también tiene una serie de tema a la derecha del comodín coincidente, la suscripción nunca recibe una publicación. El conjunto de publicaciones que no están bloqueadas son publicaciones para temas que son padres del comodín bloqueado. Las publicaciones para temas que son hijos del tema que tiene el valor BLOCK para la propiedad están bloqueadas por el comodín. Por consiguiente, las series de publicación que incluyen un tema a la derecha del comodín nunca reciben publicaciones coincidentes.

Definir el valor de la propiedad WILDCARD en BLOCK no significa que no pueda suscribirse utilizando una serie de tema que incluya comodines. Una suscripción de este tipo es normal. La suscripción tiene un tema explícito que coincide con el tema que tiene un objeto de tema con la propiedad WILDCARD establecida en el valor BLOCK. Utiliza comodines para temas que son padres o hijos del tema con la propiedad WILDCARD establecida en el valor BLOCK. En el ejemplo de [Figura 23](#) en la [página 81](#), una suscripción como Sports/Football/# puede recibir publicaciones.

**Comodines y temas de clúster**

Las definiciones de tema de clúster se propagan a todos los gestores de colas de un clúster. Una suscripción a un tema de clúster en un gestor de colas de un clúster da lugar a que el gestor de colas cree suscripciones de proxy. Se crea una suscripción de proxy cada dos gestores de colas del clúster. Las suscripciones que utilizan series de temas que contienen caracteres, combinados con temas de clúster, pueden dificultar la capacidad de predecir el comportamiento. El comportamiento se describe en el siguiente ejemplo.

En el clúster configurado para el ejemplo, [“Ejemplo: crear el clúster de publicación/suscripción Sport”](#) en la [página 84](#), QMB tiene el mismo conjunto de suscripciones que QMA, pero QMB no ha recibido ninguna publicación después de que el editor haya publicado en QMA, consulte [Figura 24](#) en la [página 81](#). Aunque los temas Sports/Football y Sports/Rugby son temas de clúster, las suscripciones definidas en `fullsubs.tst` no hacen referencia al tema de clúster. No se propaga ninguna suscripción de proxy de QMB a QMA. Sin suscripciones de proxy, ninguna publicación para QMA se reenvía a QMB.

Podría parecer que algunas de las suscripciones, como Sports/#/Leeds, hacen referencia a un tema de clúster, en este caso Sports/Rugby. En realidad, la suscripción Sports/#/Leeds se resuelve en el objeto de tema SYSTEM.BASE.TOPIC.

A continuación se indica la regla para resolver el objeto de tema al que hace referencia una suscripción como Sports/#/Leeds. Trunque la serie de tema en el primer comodín. Explore a la izquierda en la serie del tema para buscar el primer tema que tiene un objeto de tema administrativo asociado. El objeto de tema puede especificar un nombre de clúster, o definir un objeto de tema local. En el ejemplo, Sports/#/Leeds, la serie de tema después del truncamiento es Sports, que no tiene objeto de tema, y, por tanto, Sports/#/Leeds hereda de SYSTEM.BASE.TOPIC, que es un objeto de tema local.

Para ver cómo la suscripción a temas agrupados en clúster puede cambiar el modo en que funciona la propagación de comodín, ejecute el script por lotes `upsubs.bat`. El script borra las colas de suscripción y agrega las suscripciones de tema de clúster de `fullsubs.tst`. Ejecute `puba.bat` de nuevo para crear un lote de publicaciones; consulte [Figura 24](#) en la [página 81](#).

En la [Tabla 4](#) en la [página 83](#) se muestra el resultado de añadir dos nuevas suscripciones al mismo gestor de colas en el que se han publicado las publicaciones. El resultado es como se esperaba, las suscripciones nuevas reciben una publicación cada una, y el número de publicaciones recibidas por las demás suscripciones no cambia. Los resultados inesperados se producen en el otro gestor de colas de clúster; consulte [Tabla 5](#) en la [página 83](#).

<i>Tabla 4. Publicaciones recibidas en QMA</i>			
<b>Suscripción</b>	<b>Serie de tema</b>	<b>Publicaciones recibidas</b>	<b>Notas</b>
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas las publicaciones del subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal recibe una publicación porque la suscripción no tiene ningún comodín.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds recibirá una publicación en cualquier caso.

En la [Tabla 5](#) en la [página 83](#) se muestra el resultado de añadir las dos nuevas suscripciones en QMB y publicar en QMA. Recuerde que QMB no recibe publicaciones sin estas dos nuevas suscripciones. Tal como se esperaba, las dos nuevas suscripciones reciben publicaciones, ya que Sports/FootBall y Sports/Rugby son ambos temas de clúster. QMB ha reenviado suscripciones de proxy para Sports/Football/Arsenal y Sports/Rugby/Leeds a QMA, que luego ha reenviado las publicaciones a QMB.

El resultado inesperado es que las dos suscripciones Sports/# y Sports/#/Leeds que antes no recibían publicaciones, ahora reciben publicaciones. El motivo es que las publicaciones Sports/Football/Arsenal y Sports/Rugby/Leeds reenviadas a QMB para las demás suscripciones están ahora disponibles para cualquier suscriptor conectado a QMB. En consecuencia las suscripciones a los temas locales Sports/# y Sports/#/Leeds reciben la publicación Sports/Rugby/Leeds. Sports/#/Arsenal continúa sin recibir una publicación, porque Sports/Football tiene su propiedad WILDCARD con el valor establecido en BLOCK.

<i>Tabla 5. Publicaciones recibidas en QMB</i>			
<b>Suscripción</b>	<b>Serie de tema</b>	<b>Publicaciones recibidas</b>	<b>Notas</b>
SPORTS	Sports/#	Sports/Rugby/ Leeds	Todas las publicaciones al subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal

Tabla 5. Publicaciones recibidas en QMB (continuación)

Suscripción	Serie de tema	Publicaciones recibidas	Notas
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado de WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal recibe una publicación porque la suscripción no tiene ningún comodín.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds recibirá una publicación en cualquier caso.

En muchas aplicaciones, no es aconsejable que una suscripción influya en el comportamiento de otra suscripción. Un uso importante de la propiedad WILDCARD con el valor BLOCK es hacer que las suscripciones a la misma serie de tema que contengan comodines se comporten de manera uniforme. Si la suscripción se encuentra en el mismo gestor de colas que el publicador o uno diferente, los resultados de la suscripción son los mismos.

### Comodines y corrientes

Para una nueva aplicación escriba en la API de publicación/suscripción, el efecto es que una suscripción a \* no recibe publicaciones. Para recibir todas las publicaciones de Sports, debe suscribirse a Sports/\*, o Sports/#, y lo mismo para las publicaciones de Business.

El comportamiento de una aplicación de publicación/suscripción en cola existente no cambia cuando el intermediario de publicación/suscripción se migra a una versión posterior de IBM MQ. La propiedad **StreamName** de los mandatos **Publish**, **Register Publisher** o **Subscriber** se correlaciona con el nombre del tema al que se ha migrado la corriente.

### Comodines y puntos de suscripción

Para una nueva aplicación escriba en la API de publicación/suscripción, el efecto de la migración es que una suscripción a \* no recibe publicaciones. Para recibir todas las publicaciones de Sports, debe suscribirse a Sports/\*, o Sports/#, y lo mismo para las publicaciones de Business.

El comportamiento de una aplicación de publicación/suscripción en cola existente no cambia cuando el intermediario de publicación/suscripción se migra a una versión posterior de IBM MQ. La propiedad **SubPoint** de los mandatos **Publish**, **Register Publisher** o **Subscriber** se correlaciona con el nombre del tema al que se ha migrado la suscripción.

### Ejemplo: crear el clúster de publicación/suscripción Sport

Los pasos que se indican a continuación crean un clúster, CL1, con cuatro gestores de colas: dos repositorios completos, CL1A y CL1B, y dos repositorios parciales, QMA y QMB. Los repositorios completos se utilizan sólo para las definiciones de clúster. QMA se designa como host de temas de clúster. Las suscripciones duraderas se definen en QMA y QMB.

**Nota:** El ejemplo está codificado para Windows. Debe volver a codificar [Create qmgrs.bat](#) y [Create pub.bat](#) para configurar y probar el ejemplo en otras plataformas.

1. Cree los archivos de script.
  - a. [Create topics.tst](#)
  - b. [Crear wildsubs.tst](#)
  - c. [Crear fullsubs.tst](#)
  - d. [Crear qmgrs.bat](#)

e. crear pub.bat

2. Ejecute `Create qmgrs.bat` para crear la configuración.

```
qmgrs
```

Cree los temas de [Figura 23 en la página 81](#). El script de la figura 5 crea los temas de clúster Sports/ Football y Sports/Rugby.

**Nota:** La opción REPLACE no sustituye las propiedades TOPICSTR de un tema. TOPICSTR es una propiedad que normalmente cambia en el ejemplo para probar distintos árboles de temas. Para cambiar temas, suprima primero el tema.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

*Figura 25. Suprimir y crear temas: topics.tst*

**Nota:** Suprima los temas, ya que REPLACE no sustituye series de tema.

Cree suscripciones con comodines. Los comodines corresponden a los temas con objetos de temas en [Figura 23 en la página 81](#). Cree una cola para cada suscripción. Las colas se borran y las suscripciones se suprimen cuando se ejecuta o vuelve a ejecutar el script.

**Nota:** La opción REPLACE no sustituye las propiedades TOPICOBJ o TOPICSTR de una suscripción. TOPICOBJ o TOPICSTR son las propiedades que cambian normalmente en el ejemplo para probar distintas suscripciones. Para cambiarlas, suprima primero la suscripción.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

*Figura 26. Cree suscripciones comodín: wildsubs.tst*

Cree suscripciones que hagan referencia a lo objetos de tema de clúster.

**Nota:**

El delimitador, /, se inserta automáticamente entre la serie de tema a la que hace referencia TOPICOBJ y la serie de tema definida por TOPICSTR.

La definición, `DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal')` `DEST(QFARSENAL)` crea la misma suscripción. TOPICOBJ se utiliza como método rápido para hacer

referencia a la serie de tema que ya se ha definido. La suscripción, cuando se crea, ya no hace referencia al objeto de tema.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Figura 27. Suprima y cree suscripciones: fullsubs.tst

Cree un clúster con dos depósitos. Cree dos repositorios parciales para publicación y suscripción. Vuelva a ejecutar el script para suprimirlo todo y empezar de nuevo. El script también crea la jerarquía de temas y las suscripciones comodín iniciales.

**Nota:**

En otras plataformas, escriba un script parecido o escriba todos los mandatos. El uso de un script permite suprimirlo todo con mayor rapidez y empezar de nuevo con una configuración idéntica.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Figura 28. Cree gestores de colas: qmgrs.bat

Actualice la configuración añadiendo las suscripciones a los temas de clúster.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Figura 29. Actualice las suscripciones: upsubs.bat

Ejecute pub.bat, con un gestor de colas como parámetro, para publicar mensajes que contengan la serie de tema de publicación. Pub.bat utiliza el programa de ejemplo **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figura 30. Publique: pub.bat

## Corrientes y temas

La publicación/suscripción en cola tiene el concepto de una corriente de publicación que no existe en el modelo de publicación/suscripción integrada. En la publicación/suscripción en cola, las corrientes ofrecen un modo de separar el flujo de información para distintos temas. Una corriente se implementa como un tema de nivel superior que se puede correlacionar con un identificador de tema diferente administrativamente.

La corriente predeterminada `SYSTEM.BROKER.DEFAULT.STREAM` se configura automáticamente para todos los intermediarios y gestores de colas de una red y no es necesaria ninguna configuración adicional para utilizar la corriente predeterminada. Piense en la corriente por omisión como en un espacio de temas por omisión no indicado. Los temas publicados en la corriente predeterminada están disponibles inmediatamente para todos los gestores de colas conectados, con la publicación/suscripción en cola habilitada. Las corrientes indicadas son como espacios de temas independientes indicados. La corriente indicada debe definirse en todos los intermediarios donde se utilice.

Si los publicadores y suscriptores están en distintos gestores de colas, después de los intermediarios se conecten en la misma jerarquía de intermediarios, no será necesaria ninguna configuración adicional para que las publicaciones y las suscripciones fluyan entre ellos. La misma interoperabilidad funciona a la inversa.

## Corrientes indicadas

Un diseñador de soluciones que trabaja con el modelo de programación de publicación/suscripción en cola, puede haber decidido colocar todas las publicaciones de deporte en una corriente con nombre denominada `Sport`. Para que la corriente esté disponible para un gestor de colas que se ejecuta en IBM MQ con la publicación/suscripción en cola habilitada, la corriente debe añadirse manualmente.

Las aplicaciones de publicación/suscripción en cola que se suscriben a `Soccer/Results` en la corriente `Sport` funcionan sin cambios. Las aplicaciones de publicación/suscripción integradas que se suscriben al tema `Sport` utilizando `MQSUB` y proporcionando la serie de tema `Soccer/Results`, reciben también las mismas publicaciones.

La tarea de añadir una corriente se describe en el tema [Adición de una corriente](#). Puede que tenga que añadir corrientes manualmente por dos razones.

1. Sigue desarrollando sus aplicaciones de publicación/suscripción en cola que se ejecutan en gestores de colas de versiones posteriores, en lugar de migrar las aplicaciones a la interfaz MQI de publicación/suscripción integrada.
2. La correlación por omisión de corrientes a temas lleva a una "colisión" en el espacio de temas, y las publicaciones de una corriente tienen la misma serie de temas que las publicaciones de otros sitios.

## Autorizaciones

De forma predeterminada, en la raíz del árbol de temas hay varios objetos de tema: `SYSTEM.BASE.TOPIC`, `SYSTEM.BROKER.DEFAULT.STREAM` y `SYSTEM.BROKER.DEFAULT.SUBPOINT`. Las autoridades (por ejemplo, para la publicación o suscripción) las determinan las autoridades en el `SYSTEM.BASE.TOPIC`; las autorizaciones sobre `SYSTEM.BROKER.DEFAULT.STREAM` o `SYSTEM.BROKER.DEFAULT.SUBPOINT` se ignoran. Si `SYSTEM.BROKER.DEFAULT.STREAM` o `SYSTEM.BROKER.DEFAULT.SUBPOINT` se suprimen y se vuelven a crear con una serie de tema no vacía, las autorizaciones definidas en dichos objetos se utilizan del mismo modo que un objeto de tema normal.

## Correlación entre corrientes y temas

Una corriente de publicación/suscripción en cola se imita en IBM MQ creando una cola y dándole el mismo nombre que la corriente. En algunas ocasiones la cola se llama cola de corriente, porque así es como aparece en las aplicaciones de publicación/suscripción en cola. La cola es identificada al motor de publicación/suscripción añadiéndola a la lista de nombres especial llamada `SYSTEM.QPUBSUB.QUEUE.NAMELIST`. Puede añadir tantas corrientes como necesite, añadiendo colas especiales adicionales a la lista de nombres. Para finalizar debe añadir temas, con los mismos nombres que las corrientes, y las mismas series de temas que el nombre de tema, para poder publicar y suscribirse a los temas.

Sin embargo, en circunstancias excepcionales, puede dar a los temas correspondientes a las corrientes cualquier serie de tema que decida cuando defina los temas. La finalidad de la serie de tema es otorgar al tema un nombre exclusivo en el espacio del tema. Generalmente, el nombre de corriente cumple ese propósito a la perfección. A veces, un nombre de corriente y un nombre de tema existente entran en conflicto. Para resolver el problema, elija otra serie de tema para el tema asociado a la corriente. Elija cualquier serie de tema asegurándose de que sea exclusivo.

La serie de tema definida en la definición del tema recibe prefijo de modo normal a la serie de tema ofrecida por los publicadores y suscriptores utilizando las llamadas de `MQI MQOPEN` o `MQSUB`. Las aplicaciones que hacen referencia a temas utilizando objetos de tema no se ven afectados por la elección de serie de tema con prefijo, por eso puede elegir cualquier serie de tema que mantenga las publicaciones como únicas en el espacio de tema.

Para volver a correlacionar corrientes diferentes en temas diferentes se necesita que los prefijos utilizados para las series de temas sean únicos para separar totalmente un conjunto de temas de otro. Debe definir una convención de nombres de tema universal a la que debe adherirse estrictamente para que la correlación funcione.

En IBM MQ, utilice el mecanismo de prefijo para volver a correlacionar una serie de tema con otro lugar del espacio de temas.

**Nota:** Cuando elimine una corriente, elimine primero todas las suscripciones en la corriente. Esta acción es esencial si alguna de las suscripciones es origina desde otros intermediarios en la jerarquía de intermediarios.

### ***Puntos de suscripción y temas***

Los puntos de suscripción con nombre se emulan mediante temas y objetos de tema.

Para añadir puntos de suscripción manualmente, consulte [Adición de un punto de suscripción](#).

## Puntos de suscripción de IBM MQ

IBM MQ correlaciona puntos de suscripción con distintos espacios de temas dentro del árbol de temas de IBM MQ. Los temas de los mensajes de mandato sin un punto de suscripción se correlacionan sin ningún cambio en la raíz del árbol de temas de IBM MQ y heredan las propiedades de `SYSTEM.BASE.TOPIC`.

Los mensajes de mandato con un punto de suscripción se procesan utilizando la lista de objetos de tema de `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`. El nombre del punto de suscripción en el mensaje de mandato se compara con la serie de tema para cada uno de los objetos de tema en la lista. Si se encuentra una coincidencia, el nombre de punto de suscripción se antepone, como un nodo de tema, a la serie de tema. El tema hereda sus propiedades del objeto de tema asociado que se encuentra en `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`.

El efecto de utilizar puntos de suscripción es crear un espacio de temas distinto para cada punto de suscripción. El espacio de temas se basa en un tema que tiene el mismo nombre que el punto de suscripción. Los temas de cada espacio de temas heredan sus propiedades del objeto de tema con el mismo nombre que el punto de suscripción.

Las propiedades no establecidas en el objeto de tema coincidente se heredan, de forma normal, de `SYSTEM.BASE.TOPIC`.

Las aplicaciones de publicación/suscripción en cola existentes, utilizando cabeceras de mensajes MQRFH2, siguen funcionando estableciendo la propiedad **SubPoint** en los mensajes de mandato Publish o Register subscriber. El punto de suscripción se combina con la serie del tema en el mensaje de mandato y el tema lo procesa como cualquier otro.

Las aplicaciones de IBM MQ no se ven afectadas por los puntos de suscripción. Si una aplicación utiliza un tema que hereda información de uno de los objetos de tema coincidentes, dicha aplicación interoperará con una aplicación encolada utilizando el punto de suscripción coincidente.

## Ejemplo

Una aplicación existente de publicación/suscripción de WebSphere Message Broker (ahora llamado IBM Integration Bus) migrada a IBM MQ ha creado dos objetos de tema, GBP y USD, con las correspondientes cadenas de tema 'GBP' y 'USD'.

Los publicadores existentes en el tema NYSE/IBM/SPOT, que han migrado para ejecutarse en IBM MQ y que utilizan el punto de suscripción USD crean publicaciones sobre el tema USD/NYSE/IBM/SPOT. Asimismo, los suscriptores existentes en NYSE/IBM/SPOT, utilizando el punto de suscripción USD crean suscripciones a USD/NYSE/IBM/SPOT.

Suscríbase al precio al contado en dólares en un programa de publicación/suscripción de IBM MQ llamando a MQSUB. Cree una suscripción utilizando el objeto de tema USD y la serie de tema 'NYSE/IBM/SPOT', tal como se muestra en el fragmento de código 'C'.

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

1. Establezca el atributo CLUSTER de los objetos de tema USD y GBP en el host de tema de clúster.
2. Suprima todas las copias de los objetos de tema USD y GBP en otros gestores de colas del clúster.
3. Asegúrese de que USD y GBP estén definidos en SYSTEM.QPUBSUB.SUBPOINT.NAMELIST en cada gestor de colas del clúster.

## Ejemplo de publicación/suscripción de gestor de colas individual

Figura 31 en la página 90 ilustra una única configuración básica de publicación/suscripción de gestor de colas. El ejemplo muestra la configuración de un servicio de noticias, donde hay información disponible de los publicadores sobre varios temas:

- Publicador 1 es la información de publicación acerca de los resultados deportivos utilizando el tema Deporte
- Publicador 2 es la información de publicación sobre los precios de las acciones utilizando el tema Acciones
- Publicador 3 es la información de publicación sobre las reseñas de películas utilizando el tema Películas, y sobre los programas de televisión utilizando el tema TV

Tres suscriptores han registrado su interés en distintos temas, de forma que el gestor de colas les envía información en la que están interesados:

- El suscriptor 1 recibe los resultados deportivos y los precios de las acciones
- El suscriptor 2 recibe las reseñas de películas
- El suscriptor 3 recibe los resultados deportivos

Ninguno de los suscriptores han registrado estar interesado en la programación de televisión, por lo que esta no se distribuye.

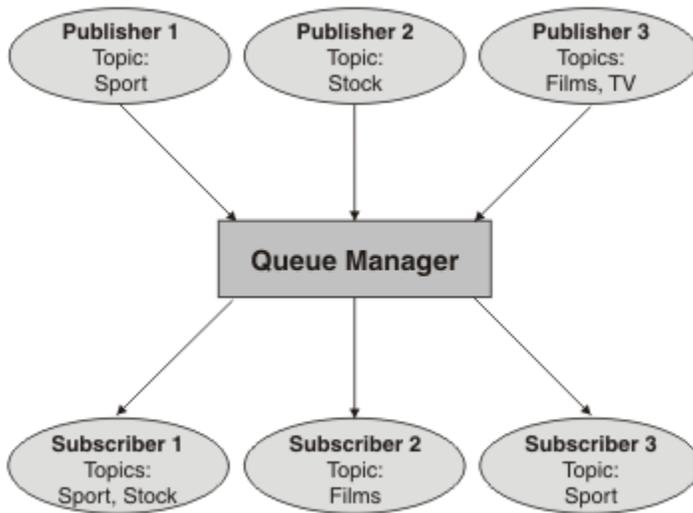


Figura 31. Ejemplo de publicación/suscripción de gestor de colas único

## Redes de publicación/suscripción distribuidas

Cada gestor de colas compara los mensajes publicados en un tema con las suscripciones creadas localmente que se han suscrito a ese tema. Puede configurarse una red de gestores de colas de forma que los mensajes publicados por una aplicación conectada a un gestor de colas se entreguen a las suscripciones coincidentes creadas en otros gestores de colas de la red. Esto requiere una configuración adicional a simples canales entre gestores de colas.

Una configuración de publicación/suscripción distribuida es un conjunto de gestores de colas conectados entre sí. Los gestores de colas pueden estar todos en el mismo sistema físico o pueden estar distribuidos entre varios sistemas físicos. Cuando se conectan gestores de colas entre sí, los suscriptores pueden suscribirse a un gestor de colas y recibir mensajes publicados inicialmente en otro gestor de colas. Para ilustrar esto, la siguiente figura añade un segundo gestor de colas a la configuración descrita en [“Ejemplo de publicación/suscripción de gestor de colas individual”](#) en la página 89.

- El gestor de colas 2 es utilizado por el editor 4 para publicar información de previsión meteorológica, utilizando un tema Tiempo e información sobre las condiciones de tráfico en las rutas principales utilizando un tema Tráfico.
- El editor 4 también utiliza el gestor de colas, y se suscribe a la información sobre las condiciones de tráfico utilizando el tema Tráfico.
- El editor 3 también se suscribe a información sobre las condiciones meteorológicas, aunque utiliza un gestor de colas diferente del editor. Esto es posible debido a que los gestores de colas se enlazan entre sí.

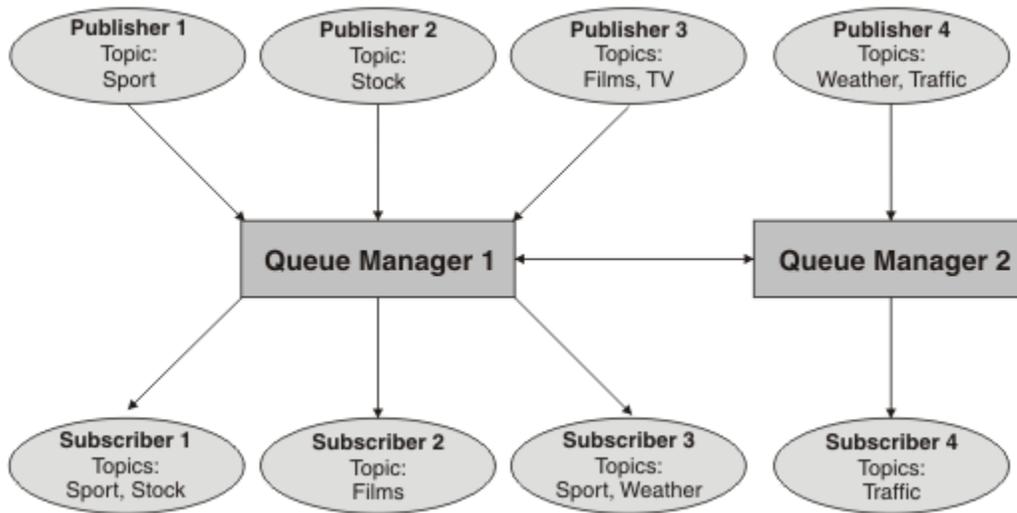


Figura 32. Ejemplo de publicación/suscripción con dos gestores de colas

Puede conectar manualmente los gestores de colas en una jerarquía padre e hijo o puede crear un clúster de publicación/suscripción y dejar que IBM MQ defina automáticamente gran parte del detalle de conexión. También pueden utilizarse ambas topologías conjuntamente, por ejemplo, uniendo varios clústeres en una jerarquía.

### Visión general de un clúster de publicación/suscripción

Un clúster de publicación/suscripción es un clúster estándar al que se le han añadido uno o más objetos de tema. Cuando se define un objeto de tema administrativo en cualquier gestor de colas de un clúster, y se pone dicho objeto de tema en clúster especificando un nombre de clúster, los publicadores y suscriptores del tema podrán conectar con cualquiera de los gestores de colas del clúster y los mensajes publicados se direccionarán a los suscriptores a través de canales de clúster entre gestores de colas.

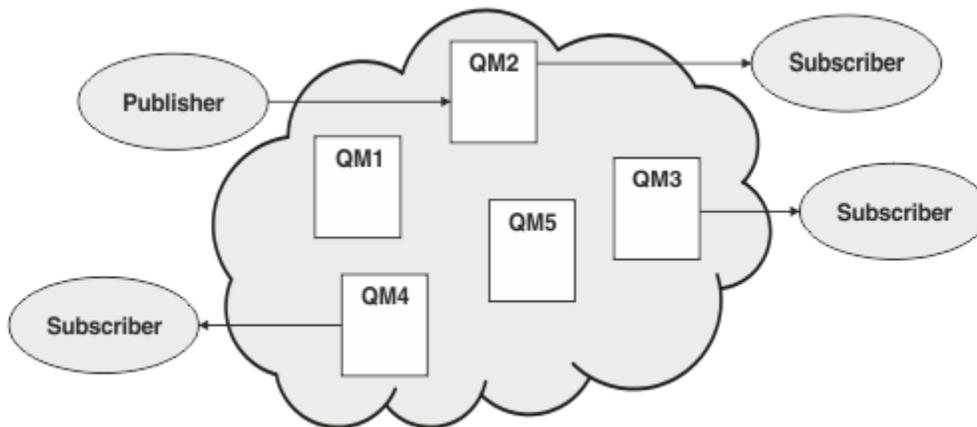


Figura 33. Clúster de publicación/suscripción

Hay dos maneras de configurar el modo en que los mensajes de publicación/suscripción se direccionan en un clúster:

- direccionamiento directo
- direccionamiento de host de tema

Cuando se configura un tema en clúster por direccionamiento directo, los mensajes publicados en un gestor de colas se envían directamente desde ese gestor de colas a todas las suscripciones en cualquier otro gestor de colas del clúster. Esto puede proporcionar la ruta más directa para las publicaciones, pero

da lugar a que todos los gestores de colas del clúster tengan conocimiento de todos los demás gestores de colas, pudiendo en potencia haber canales de clúster establecidos entre todos ellos.

Cuando se utiliza el direccionamiento de host de tema, los mensajes publicados en un gestor de colas se envían desde ahí a un gestor de colas que aloja una definición del objeto de tema administrado. Ese *gestor de colas de host de temas* direcciona el mensaje a cada suscripción en cualquier otro gestor de colas del clúster. Si los publicadores o los suscriptores no están ubicados en los gestores de colas de host de tema, las publicaciones seguirán una ruta más larga. Sin embargo, la ventaja es que solo los gestores de cola de host de tema tienen conocimiento de todos los demás gestores de colas del clúster y (posiblemente) tienen canales de clúster establecidos con ellos.

Para obtener más información, consulte [“Clústeres de publicación/suscripción”](#) en la página 93.

## Visión general de las jerarquías de publicación/suscripción

Una jerarquía de publicación/suscripción es un conjunto de gestores de colas conectados mediante canales en una estructura jerárquica. Cada gestor de colas identifica su gestor de colas *padre*, tal y como se describe en [Conexión de un gestor de colas a una jerarquía de publicación/suscripción](#).

Los publicadores y suscriptores de un tema pueden conectarse con cualquier gestor de colas de la jerarquía y los mensajes fluyen entre ellos valiéndose de la conectividad jerárquica de los gestores de colas.

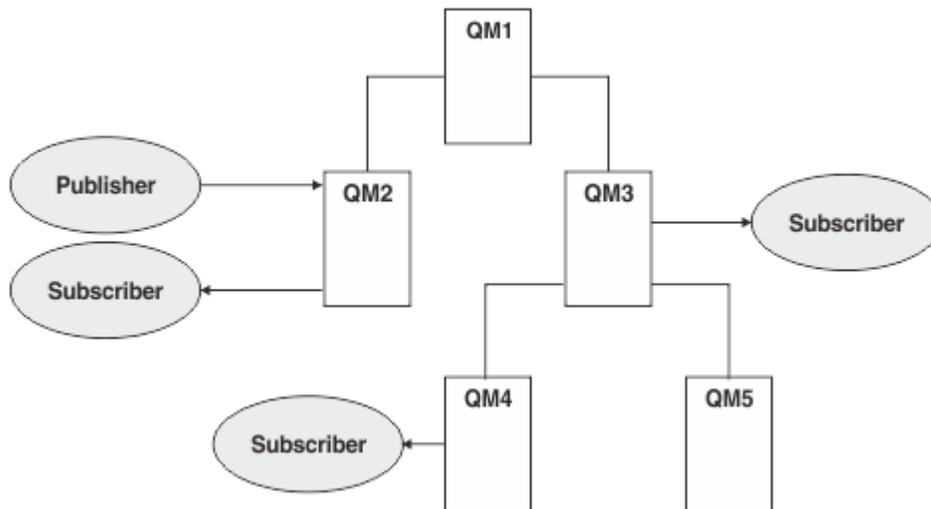


Figura 34. Jerarquía de publicación/suscripción

En la figura anterior, las publicaciones entregadas a los suscriptores en QM3 y QM4 se han direccionado de QM2 a QM1 y luego a QM3 y finalmente a QM4.

Las jerarquías dan un control directo sobre las relaciones existentes entre cada gestor de colas en la jerarquía. Esto permite un control preciso sobre el direccionamiento de los mensajes desde los publicadores a los suscriptores y resulta de especial utilidad cuando se direccionan entre redes de gestores de colas con conectividad limitada. Debe prestarse especial atención a la disponibilidad y capacidad de cada uno de los gestores de colas a través de los cuales se direcciona un mensaje en su camino desde el publicador al suscriptor.

Para obtener más información, consulte [“Jerarquías de publicación/suscripción”](#) en la página 96.

## Distribución de una publicación entre gestores de colas

Además de las opciones de direccionamiento, existen dos enfoques en la distribución de las publicaciones en una red de gestores de colas:

- Solo enviar las publicaciones desde un gestor de colas a los gestores de colas que en ese momento alojen una suscripción a esa publicación.
- Enviar todas las publicaciones a todos los gestores de colas, y dejar que ellos las comparen con sus suscripciones.

El primer enfoque da lugar a que solo se envíen mensajes de publicación cuando es necesario, pero requiere que los gestores de colas compartan un cierto nivel de conocimiento de las suscripciones. El segundo enfoque no requiere compartición de conocimiento de suscripciones, pero puede dar lugar al envío innecesario de mensajes de publicación entre los gestores de colas.

De forma predeterminada, IBM MQ utiliza el método anterior, en el que las publicaciones sólo se envían a los gestores de colas que tienen suscripciones para ellos. El conocimiento de las suscripciones se propaga entre los gestores de colas en forma de *suscripciones de proxy*. Cuál de los dos será el más eficiente en el uso de una topología distribuida de publicación/suscripción dependerá de la distribución y del tiempo de vida de las suscripciones, y de la frecuencia de las publicaciones. Consulte [Rendimiento de suscripción en redes de publicación/suscripción](#).

### Conceptos relacionados

[“Árboles de temas” en la página 77](#)

Cada tema que se define es un elemento, o nodo, del árbol de temas. El árbol de temas puede estar vacío para empezar o puede contener temas que se hayan definido anteriormente utilizando mandatos MQSC o PCF. Puede definir un nuevo tema utilizando los mandatos crear tema o especificando el tema por primera vez en una publicación o suscripción.

[Escenarios de jerarquías de publicación/suscripción](#)

### Tareas relacionadas

[Diseño de clústeres de publicación/suscripción](#)

## Clústeres de publicación/suscripción

Un clúster de publicación es un clúster estándar de gestores de colas interconectados en el que las publicaciones se mueven automáticamente desde las aplicaciones de publicación hasta las suscripciones que existen en cualquiera de los gestores de colas en el clúster. Hay dos opciones para direccionar publicaciones a través de un clúster de publicación/suscripción: *direccionamiento directo* y *direccionamiento de host de temas*. El direccionamiento elegido dependerá del tamaño y de los patrones de actividad esperados del clúster.

Un clúster que se utiliza para la mensajería de publicación/suscripción no es diferente de un clúster IBM MQ estándar. Como tal, los gestores de colas del clúster de publicación/suscripción pueden existir en sistemas separados físicamente y cada par de gestores de colas se conectan automáticamente entre ellos con canales de clúster cuando sea necesario. Para obtener más información, consulte [Clústeres](#).

Para configurar un clúster estándar de gestores de colas para la mensajería de publicación/suscripción, defina uno o más objetos de tema administrados en un gestor de colas del clúster. Para hacer que el tema sea un tema de clúster, configure la propiedad **CLUSTER** con el nombre del clúster. Al hacerlo, cualquier tema que utilice un publicador o suscriptor en ese punto o por debajo en el [árbol de temas](#) será compartido entre todos los gestores de colas del clúster y los mensajes publicados en una rama en clúster del árbol de temas se direccionarán de forma automática a suscripciones en otros gestores de colas del clúster.

Solo se envía una copia de cada mensaje entre el gestor de colas publicador y cada uno de los demás gestores de colas, independientemente del número de suscriptores al mensaje en el gestor de colas de destino. Al llegar a un gestor de colas con una o más suscripciones, el mensaje se duplica en todas las suscripciones.

Cualquier gestor de colas que se una al clúster tendrá conocimiento automático de todos los temas en clúster y los publicadores y suscriptores en ese gestor de colas participarán de forma automática en el clúster.

Una actividad de publicación/suscripción no agrupada en clúster también puede tener lugar en un clúster de publicación/suscripción, trabajando con series de tema que no entran en un objeto de tema agrupado en clúster.

Hay dos opciones para direccionar publicaciones a través de un clúster de publicación/suscripción: *direccionamiento directo* y *direccionamiento de host de temas*. Para elegir el direccionamiento de mensajes a utilizar en el clúster, establezca la propiedad **CLROUTE** en el objeto de tema administrado en uno de los valores siguientes:

- **DIRECT**
- **TOPICHOST**

De forma predeterminada, el direccionamiento de temas es **DIRECT**. Era la única opción antes de la IBM MQ 8.0. Cuando se configura un tema de clúster de direccionamiento directo en un gestor de colas, todos los gestores de colas del clúster reconocen los otros gestores de colas del clúster. Al realizar operaciones de publicación y suscripción, cada gestor de colas puede conectarse directamente a todos los otros gestores de colas del clúster.

A partir de IBM MQ 8.0, en su lugar, puede configurar el direccionamiento de temas como **TOPICHOST**. Cuando se utiliza el direccionamiento de host de temas, todos los gestores de colas del clúster pasan a reconocer los gestores de colas del clúster que alojan la definición del tema direccionado (es decir, los gestores de colas en los que se ha definido el objeto de tema). Cuando se realizan operaciones de publicación y suscripción, los gestores de colas del clúster sólo se conectan a estos gestores de colas de host de temas, no directamente entre sí. Los gestores de colas de host de temas son responsables del direccionamiento de publicaciones desde los gestores de colas en los que se publican publicaciones y los gestores de colas con suscripciones coincidentes.

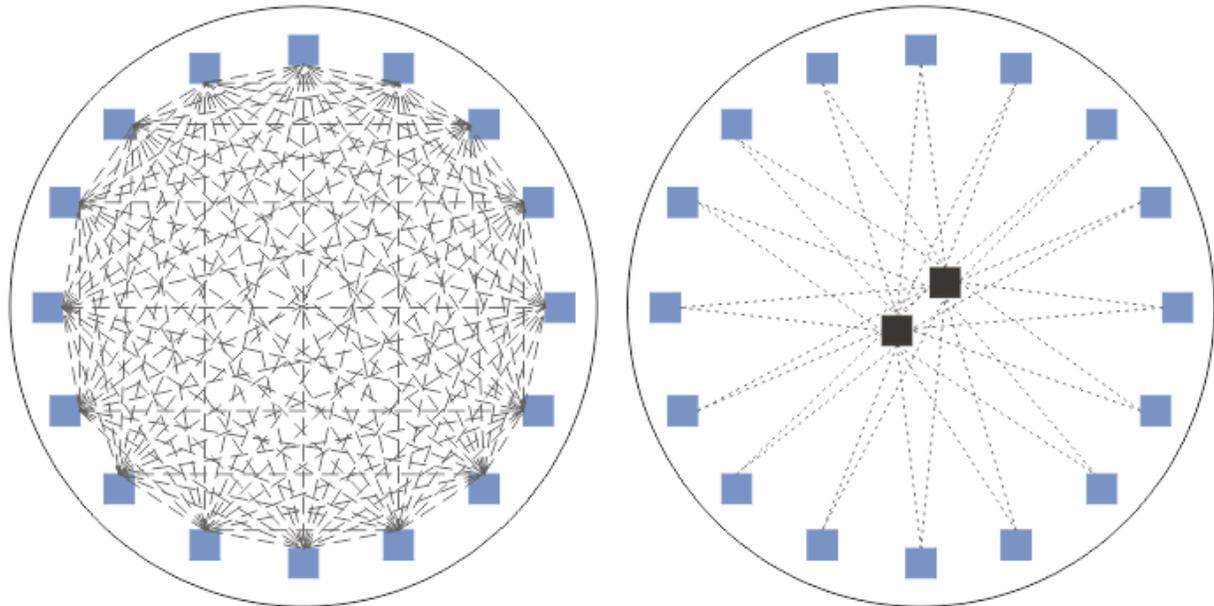


Figura 35. Direccionamiento directo y direccionamiento de host de temas

### Descripción general del direccionamiento directo

Cuando se configura un objeto de tema administrado para direccionamiento directo, solo hay que definir dicho objeto de tema en uno de los gestores de colas del clúster para que todos los gestores de colas tengan conocimiento de él. La elección del gestor de colas en el que se define el tema no afecta al comportamiento de la mensajería de publicación/suscripción del tema.

Cada mensaje fluye directamente del gestor de colas publicador a cada suscripción en los demás gestores de colas del clúster sin pasar por gestores de colas intermedios.

De forma predeterminada, los mensajes solo se envían a otros gestores de colas en el clúster que alojan una o más suscripciones.

- Esto depende de que cada gestor de colas informe directamente a todos los demás gestores de colas del clúster de todos los temas que en ese momento tengan una o más suscripciones a él. Esto da

como resultado que todos los gestores de colas del clúster tengan conocimiento de todos los temas suscritos y que cualquier gestor de colas que aloje una suscripción establezca un canal a todos los demás gestores de colas. Esto es independiente de que cada gestor de colas tenga o no un publicador.

- El conocimiento de cada tema suscrito individual en todos los gestores de colas puede eliminarse cambiando a un modelo de envío de todas las publicaciones a todos los gestores de colas del clúster independientemente de que tengan o no suscripciones. Esto disminuye el tráfico de conocimiento de las suscripciones pero es probable que aumente el tráfico de publicaciones y el número de canales que establece cada gestor de colas. Consulte [Rendimiento de suscripción en redes de publicación/suscripción](#).

Los flujos de mensajes de publicación/suscripción que utilizan temas en clúster de direccionamiento directo pueden abarcar múltiples clústeres de publicación/suscripción añadiendo un gestor de colas de cada clúster a una jerarquía de publicación/suscripción. Consulte [Combinación de espacios de temas de varios clústeres](#).

Para obtener una exposición más detallada del direccionamiento directo, consulte [Direccionamiento directo en clústeres de publicación/suscripción](#).

## Descripción general del direccionamiento de host de tema

Cuando un objeto de tema administrado está configurado para el direccionamiento de host de tema, las publicaciones procedentes de un gestor de colas en el clúster se dirigen a través de un gestor de colas en el que está configurado el objeto de tema (un "host de tema"), y desde ahí van a los gestores de colas donde están las suscripciones.

- Esto depende de que cada gestor de colas informe a todos los hosts de tema de cada tema que en ese momento tenga una o más suscripciones. Todo gestor de colas que aloje una suscripción establece un canal con cada host de tema para el tema al que está asociada la suscripción.
- A los gestores de colas que no alojan temas no se les informa de otros gestores de colas que no alojan temas en el clúster a efectos de publicación/suscripción, y no se establecen canales entre ellos a tal fin.
- Si la aplicación de publicación está conectada con un gestor de colas que aloja el tema, los mensajes publicados se dirigen directamente a los gestores de colas en los que se han creado suscripciones coincidentes sin necesidad de un 'salto' adicional. De forma similar, si las suscripciones coincidentes se crean en el único gestor de colas que aloja el tema, los mensajes publicados en ese tema se dirigen directamente a ese gestor de colas sin necesidad de un salto adicional.
- Las suscripciones en el mismo gestor de colas que el publicador se satisfacen sin dirigen antes las publicaciones a los hosts del objeto de tema.

En cuanto a las colas en clúster, múltiples gestores de colas pueden configurar el mismo objeto de tema administrativo. Esto proporciona una mayor disponibilidad del direccionamiento de mensajes y un escalado horizontal mediante un equilibrado de cargas. En el caso de los objetos de tema dirigidos por host, cuando múltiples gestores de colas configuran el mismo tema nombrado para la misma rama del árbol de temas, se hace que cada host de tema tenga conocimiento de los temas suscritos por cada gestor de colas que aloja una suscripción.

- Cuando se publica un mensaje, se envía a uno de los gestores de colas de host de tema para que lo reenvíe a los gestores de colas que alojan la suscripción. La elección del gestor de colas de host de tema se ajusta a las mismas reglas de equilibrado de carga predeterminadas que las colas punto a punto en clúster.
- Si un gestor de colas publicador no puede contactar con uno o más gestores de colas de host de tema, los mensajes se dirigen a los restantes gestores de colas que alojan tema disponibles.

Cada publicación en un tema de una rama dirigida del árbol de temas se reenvía a uno de los hosts de tema incluso si no hay suscripciones a este tema en ninguna parte del clúster. De forma predeterminada, los mensajes solo se envían desde aquí a otros gestores de colas en el clúster que alojan una o más suscripciones.

- Esto se apoya en que se informe a cada gestor de colas de host de tema de todas las cadenas de tema suscritas en cada gestor de colas del clúster.

- El conocimiento de cada tema suscrito individual puede eliminarse cambiando a un modelo de envío de todas las publicaciones direccionadas a un host de tema en todos los gestores de colas del clúster independientemente de que tengan o no suscripciones. Esto reduce el tráfico de conocimiento de suscripciones, pero es probable que incremente el tráfico de publicaciones y potencialmente el número de canales establecidos con cada gestor de colas que aloja temas. Consulte [Rendimiento de suscripción en redes de publicación/suscripción](#).

Los flujos de mensajes de publicación/suscripción que utilizan temas en clúster direccionados por host de tema **no pueden** abarcar múltiples clústeres de publicación/suscripción mediante el uso de una jerarquía de publicación/suscripción.

Para obtener una exposición más detallada del direccionamiento de host de tema, consulte [Direccionamiento de host de tema en clústeres de publicación/suscripción](#).

## Jerarquías de publicación/suscripción

Una jerarquía de publicación/suscripción se crea enlazando los gestores de colas entre sí mediante canales y luego definiendo una relación padre-hijo entre pares de gestores de colas. Un mensaje fluye desde un publicador a las suscripciones a través de relaciones directas en una jerarquía. Observe que esto podría suponer múltiples "saltos" para llegar ahí.

Solo se envía una copia de un mensaje entre cualquier par de gestores de colas, independientemente del número de suscriptores al mensaje en el gestor de colas de destino. Al llegar a un gestor de colas con una o más suscripciones, el mensaje se duplica en todas las suscripciones.

De forma predeterminada, los mensajes solo se envían a otros gestores de colas en la jerarquía que están en la ruta a una suscripción en otro gestor de colas:

- Esto requiere que cada gestor de colas informe a cada relación directa de todos los temas que en ese momento tienen una o más suscripciones a él, ya sea en este gestor de colas o en uno de sus otras relaciones. Esto da lugar a que todos los gestores de colas en la jerarquía tengan conocimiento de todos los temas a los que se está suscrito.
- Este comportamiento puede modificarse para que siempre se envíen publicaciones a todos los gestores de colas de la jerarquía independientemente de las suscripciones que pueda haber. Esto elimina la necesidad de propagar la información de suscripciones por la jerarquía, pero puede incrementar el tráfico de publicaciones.

Cuando se crea un clúster, hay que procurar no crear un bucle que provoque que los mensajes estén dando vueltas por la red sin parar. Tales bucles no pueden crearse en una jerarquía.

Todo gestor de colas debe tener un nombre de gestor de colas exclusivo.

Los flujos de mensajes de publicación/suscripción pueden abarcar múltiples clústeres de publicación/suscripción. Para hacerlo, añada un gestor de colas de cada clúster a una jerarquía de publicación/suscripción.

Para obtener una exposición más detallada, consulte [Direccionamiento en jerarquías de publicación/suscripción](#).

## Suscripciones de proxy en una red de publicación/suscripción

Una suscripción de proxy es una suscripción realizada por un gestor de colas para temas publicados en otro gestor de colas. Una suscripción de proxy fluye entre los gestores de colas para cada serie de tema individual suscrita mediante una suscripción. Las suscripciones de proxy no se crean explícitamente, el gestor de colas lo hace por usted.

Puede conectar gestores de colas entre sí en un clúster de publicación/suscripción, o en una jerarquía de publicación/suscripción. Las suscripciones de proxy fluyen entre los gestores de colas conectados. Las suscripciones de proxy provocan que las publicaciones en un tema creado por un publicador conectado a un gestor de colas sean recibidas por los suscriptores a dicho tema que están conectados a otros gestores de colas. Consulte ["Redes de publicación/suscripción distribuidas"](#) en la página 90.

En las topologías de publicación/suscripción con muchos miles de suscripciones a cadenas de tema individuales, o donde la existencia de estas suscripciones puede cambiar rápidamente, se debe tener en cuenta la sobrecarga de la propagación de suscripciones de proxy. Además de la agregación automática descrita en el resto de este tema, pueden efectuarse cambios manuales de configuración que limiten aún más el flujo de publicaciones y suscripciones de proxy entre gestores de colas conectados, y eso reduce la latencia de espera en la propagación de una suscripción de proxy a todos los gestores de colas conectados. Consulte [Rendimiento de suscripción en redes de publicación/suscripción](#).

Las suscripciones de proxy no contienen ningún selector utilizado por suscripciones locales, y las cadenas de tema de suscripción que contengan comodines podrían simplificarse. Esto puede resultar en que haya publicaciones que coincidan con suscripciones de proxy donde las suscripciones reales no lo hacen, lo que da lugar a un flujo de publicaciones adicional entre gestores de colas. El gestor de colas que aloja las suscripciones filtra tales discrepancias de forma que no se devuelvan publicaciones adicionales a la suscripción.

## **Agregación de suscripciones de proxy**

Las suscripciones de proxy se agregan utilizando un sistema de eliminación de duplicados. Para una determinada serie de tema resuelta, se envía una suscripción de proxy en la primera suscripción local o suscripción de proxy recibida. Las posteriores suscripciones a la misma serie de tema utiliza esta suscripción de proxy existente.

La suscripción de proxy se cancela después de cancelarse la última suscripción local o suscripción de proxy recibida.

## **Agregación de publicaciones**

Cuando hay más de una suscripción a la misma serie de tema en un gestor de colas, sólo se envía una sola copia de cada publicación que coincida con dicha serie de tema de otros gestores de colas en la topología de publicación/suscripción. Al llegar el mensaje, el gestor de colas local entrega una copia del mensaje a cada suscripción coincidente.

Es posible que más de una suscripción de proxy coincida con la serie de tema de una sola publicación cuando las suscripciones de proxy contienen comodines. Si se publica un mensaje en un gestor de colas que coincide con dos o más suscripciones de proxy creadas por un único gestor de colas conectado, sólo una copia de la publicación se reenvía al gestor de colas remoto para satisfacer las diversas suscripciones de proxy.

### **Conceptos relacionados**

[Detección de bucles en una red distribuida de publicación/suscripción](#)

### **Comodines en suscripciones de proxy**

Las suscripciones pueden utilizar comodines en cadenas de tema para coincidir con múltiples cadenas de tema en las publicaciones.

Hay dos esquemas de comodín que puede utilizar una suscripción: *basado en tema* y *basado en carácter*. Consulte [“Esquemas de comodín” en la página 71](#).

En IBM MQ, todas las suscripciones de proxy para suscripciones comodín se convierten para utilizar comodines basados en temas. Si se encuentra un comodín basado en carácter, se sustituirá por el carácter # hasta la / más cercana. Por ejemplo, /aaa/bbb/c\*d se convierte a /aaa/bbb/#. La conversión produce que los gestores de colas remotos envíen algunas publicaciones más de aquellas a las que se suscribieron de forma explícita. El gestor de colas local filtra las publicaciones adicionales cuando entrega las publicaciones a sus suscriptores locales.

## **Control del uso de comodines con la propiedad WILDCARD**

Utilice la propiedad WILDCARD de **Topic** de MQSC o la propiedad WildcardOperation de Topic de PFC equivalente para controlar la entrega de publicaciones a las aplicaciones de suscriptor que

utilizan nombres de serie de tema comodín. La propiedad WILDCARD puede tener uno de los dos valores siguientes:

### WILDCARD

El comportamiento de las suscripciones comodín con respecto a este tema.

### PASSTHRU

Las suscripciones realizadas en un tema con comodines menos específico que la serie de tema en este objeto de tema reciben publicaciones creadas para este tema y para series de tema más específicas que este tema.

### BLOCK

Las suscripciones realizadas en un tema con comodín menos específico que la serie de tema en este objeto de tema no reciben publicaciones realizadas para este tema o para series de tema más específicas que este tema.

El valor de este atributo se utiliza cuando se definen las suscripciones. Si modifica este atributo, el conjunto de temas que abarcan las suscripciones existentes no se ve afectado por la modificación. Este escenario también se aplica si se cambia la topología cuando se crean o suprimen objetos de tema; el conjunto de temas que coinciden con las suscripciones creadas después de la modificación del atributo WILDCARD se crea utilizando la topología modificada. Si desea forzar que el conjunto de temas coincidentes se vuelva a evaluar para las suscripciones existentes, debe reiniciar el gestor de colas.

En el ejemplo, “Ejemplo: crear el clúster de publicación/suscripción Sport” en la página 84, puede seguir los pasos para crear la estructura de árbol de temas mostrada en [Figura 23](#) en la página 81.

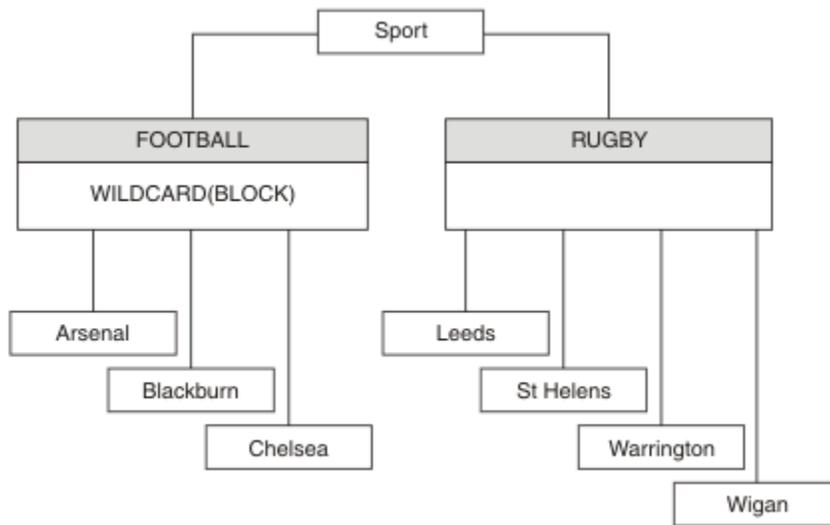


Figura 36. Árbol de temas que utiliza la propiedad WILDCARD BLOCK

Un suscriptor que utiliza la serie de tema comodín # recibe todas las publicaciones para el tema Sport y el subárbol Sport/Rugby. El suscriptor no recibe publicaciones en el subárbol Sport/Football, debido a que el valor de la propiedad WILDCARD del tema Sport/Football es BLOCK.

PASSTHRU es el valor predeterminado. Puede establecer la propiedad WILDCARD en el valor PASSTHRU en los nodos del árbol Sport. Si los nodos no tienen la propiedad WILDCARD con el valor BLOCK, definir PASSTHRU no altera el comportamiento observado por los suscriptores en los nodos del árbol Sports.

En el ejemplo, cree suscripciones para ver cómo el valor de comodín afecta a las publicaciones que se entregan; consulte [Figura 27](#) en la página 86. Ejecute el mandato de publicación de [Figura 30](#) en la página 87 para crear algunas publicaciones.

```
pub QMA
```

Figura 37. Publicación en QMA

Los resultados se muestran en [Tabla 3](#) en la [página 81](#). Observe cómo el establecimiento de la propiedad WILDCARD en BLOCK impide que las suscripciones con comodines puedan recibir publicaciones para temas del ámbito del comodín.

<i>Tabla 6. Publicaciones recibidas en QMA</i>			
<b>Suscripción</b>	<b>Serie de tema</b>	<b>Publicaciones recibidas</b>	<b>Notas</b>
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas las publicaciones del subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.

**Nota:**

Supongamos que una suscripción tiene un comodín que coincide con un objeto de tema con la propiedad WILDCARD con el valor BLOCK. Si la suscripción también tiene una serie de tema a la derecha del comodín coincidente, la suscripción nunca recibe una publicación. El conjunto de publicaciones que no están bloqueadas son publicaciones para temas que son padres del comodín bloqueado. Las publicaciones para temas que son hijos del tema que tiene el valor BLOCK para la propiedad están bloqueadas por el comodín. Por consiguiente, las series de publicación que incluyen un tema a la derecha del comodín nunca reciben publicaciones coincidentes.

Definir el valor de la propiedad WILDCARD en BLOCK no significa que no pueda suscribirse utilizando una serie de tema que incluya comodines. Una suscripción de este tipo es normal. La suscripción tiene un tema explícito que coincide con el tema que tiene un objeto de tema con la propiedad WILDCARD establecida en el valor BLOCK. Utiliza comodines para temas que son padres o hijos del tema con la propiedad WILDCARD establecida en el valor BLOCK. En el ejemplo de [Figura 23](#) en la [página 81](#), una suscripción como Sports/Football/# puede recibir publicaciones.

**Comodines y temas de clúster**

Las definiciones de tema de clúster se propagan a todos los gestores de colas de un clúster. Una suscripción a un tema de clúster en un gestor de colas de un clúster da lugar a que el gestor de colas cree suscripciones de proxy. Se crea una suscripción de proxy cada dos gestores de colas del clúster. Las suscripciones que utilizan series de temas que contienen caracteres, combinados con temas de clúster, pueden dificultar la capacidad de predecir el comportamiento. El comportamiento se describe en el siguiente ejemplo.

En el clúster configurado para el ejemplo, [“Ejemplo: crear el clúster de publicación/suscripción Sport”](#) en la [página 84](#), QMB tiene el mismo conjunto de suscripciones que QMA, pero QMB no ha recibido ninguna publicación después de que el editor haya publicado en QMA, consulte [Figura 24](#) en la [página 81](#). Aunque los temas Sports/Football y Sports/Rugby son temas de clúster, las suscripciones definidas en fullsubs.tst no hacen referencia al tema de clúster. No se propaga ninguna suscripción de proxy de QMB a QMA. Sin suscripciones de proxy, ninguna publicación para QMA se reenvía a QMB.

Podría parecer que algunas de las suscripciones, como Sports/#/Leeds, hacen referencia a un tema de clúster, en este caso Sports/Rugby. En realidad, la suscripción Sports/#/Leeds se resuelve en el objeto de tema SYSTEM.BASE.TOPIC.

A continuación se indica la regla para resolver el objeto de tema al que hace referencia una suscripción como Sports/#/Leeds. Trunque la serie de tema en el primer comodín. Explore a la izquierda en la

serie del tema para buscar el primer tema que tiene un objeto de tema administrativo asociado. El objeto de tema puede especificar un nombre de clúster, o definir un objeto de tema local. En el ejemplo, Sports/#/Leeds, la serie de tema después del truncamiento es Sports, que no tiene objeto de tema, y, por tanto, Sports/#/Leeds hereda de SYSTEM.BASE.TOPIC, que es un objeto de tema local.

Para ver cómo la suscripción a temas agrupados en clúster puede cambiar el modo en que funciona la propagación de comodín, ejecute el script por lotes [upsubs.bat](#). El script borra las colas de suscripción y agrega las suscripciones de tema de clúster de [fullsubs.tst](#). Ejecute [puba.bat](#) de nuevo para crear un lote de publicaciones; consulte [Figura 24](#) en la [página 81](#).

En la [Tabla 4](#) en la [página 83](#) se muestra el resultado de añadir dos nuevas suscripciones al mismo gestor de colas en el que se han publicado las publicaciones. El resultado es como se esperaba, las suscripciones nuevas reciben una publicación cada una, y el número de publicaciones recibidas por las demás suscripciones no cambia. Los resultados inesperados se producen en el otro gestor de colas de clúster; consulte [Tabla 5](#) en la [página 83](#).

*Tabla 7. Publicaciones recibidas en QMA*

Suscripción	Serie de tema	Publicaciones recibidas	Notas
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Todas las publicaciones del subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal recibe una publicación porque la suscripción no tiene ningún comodín.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds recibirá una publicación en cualquier caso.

En la [Tabla 5](#) en la [página 83](#) se muestra el resultado de añadir las dos nuevas suscripciones en QMB y publicar en QMA. Recuerde que QMB no recibe publicaciones sin estas dos nuevas suscripciones. Tal como se esperaba, las dos nuevas suscripciones reciben publicaciones, ya que Sports/FootBall y Sports/Rugby son ambos temas de clúster. QMB ha reenviado suscripciones de proxy para Sports/Football/Arsenal y Sports/Rugby/Leeds a QMA, que luego ha reenviado las publicaciones a QMB.

El resultado inesperado es que las dos suscripciones Sports/# y Sports/#/Leeds que antes no recibían publicaciones, ahora reciben publicaciones. El motivo es que las publicaciones Sports/Football/Arsenal y Sports/Rugby/Leeds reenviadas a QMB para las demás suscripciones están ahora disponibles para cualquier suscriptor conectado a QMB. En consecuencia las suscripciones a los temas locales Sports/# y Sports/#/Leeds reciben la publicación Sports/Rugby/Leeds. Sports/#/Arsenal continúa sin recibir una publicación, porque Sports/Football tiene su propiedad WILDCARD con el valor establecido en BLOCK.

Tabla 8. Publicaciones recibidas en QMB

Suscripción	Serie de tema	Publicaciones recibidas	Notas
SPORTS	Sports/#	Sports/Rugby/Leeds	Todas las publicaciones al subárbol Football bloqueadas por WILDCARD (BLOCK) en Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) en Sports/Football impide la suscripción de comodín en Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	El valor predeterminado de WILDCARD en Sports/Rugby no impide la suscripción comodín en Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal recibe una publicación porque la suscripción no tiene ningún comodín.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds recibirá una publicación en cualquier caso.

En muchas aplicaciones, no es aconsejable que una suscripción influya en el comportamiento de otra suscripción. Un uso importante de la propiedad WILDCARD con el valor BLOCK es hacer que las suscripciones a la misma serie de tema que contengan comodines se comporten de manera uniforme. Si la suscripción se encuentra en el mismo gestor de colas que el publicador o uno diferente, los resultados de la suscripción son los mismos.

### Comodines y corrientes

Para una nueva aplicación escriba en la API de publicación/suscripción, el efecto es que una suscripción a \* no recibe publicaciones. Para recibir todas las publicaciones de Sports, debe suscribirse a Sports/\*, o Sports/#, y lo mismo para las publicaciones de Business.

El comportamiento de una aplicación de publicación/suscripción en cola existente no cambia cuando el intermediario de publicación/suscripción se migra a una versión posterior de IBM MQ. La propiedad **StreamName** de los mandatos **Publish**, **Register Publisher** o **Subscriber** se correlaciona con el nombre del tema al que se ha migrado la corriente.

### Comodines y puntos de suscripción

Para una nueva aplicación escriba en la API de publicación/suscripción, el efecto de la migración es que una suscripción a \* no recibe publicaciones. Para recibir todas las publicaciones de Sports, debe suscribirse a Sports/\*, o Sports/#, y lo mismo para las publicaciones de Business.

El comportamiento de una aplicación de publicación/suscripción en cola existente no cambia cuando el intermediario de publicación/suscripción se migra a una versión posterior de IBM MQ. La propiedad **SubPoint** de los mandatos **Publish**, **Register Publisher** o **Subscriber** se correlaciona con el nombre del tema al que se ha migrado la suscripción.

### Ejemplo: crear el clúster de publicación/suscripción Sport

Los pasos que se indican a continuación crean un clúster, CL1, con cuatro gestores de colas: dos repositorios completos, CL1A y CL1B, y dos repositorios parciales, QMA y QMB. Los repositorios completos se utilizan sólo para las definiciones de clúster. QMA se designa como host de temas de clúster. Las suscripciones duraderas se definen en QMA y QMB.

**Nota:** El ejemplo está codificado para Windows. Debe volver a codificar Create qmgrs.bat y Create pub.bat para configurar y probar el ejemplo en otras plataformas.

1. Cree los archivos de script.
  - a. Create topics.tst
  - b. Crear wildsubs.tst
  - c. Crear fullsubs.tst
  - d. Crear qmgrs.bat
  - e. crear pub.bat
2. Ejecute Create qmgrs.bat para crear la configuración.

```
qmgrs
```

Cree los temas de [Figura 23 en la página 81](#). El script de la figura 5 crea los temas de clúster Sports/ Football y Sports/Rugby.

**Nota:** La opción REPLACE no sustituye las propiedades TOPICSTR de un tema. TOPICSTR es una propiedad que normalmente cambia en el ejemplo para probar distintos árboles de temas. Para cambiar temas, suprima primero el tema.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

*Figura 38. Suprimir y crear temas: topics.tst*

**Nota:** Suprima los temas, ya que REPLACE no sustituye series de tema.

Cree suscripciones con comodines. Los comodines corresponden a los temas con objetos de temas en [Figura 23 en la página 81](#). Cree una cola para cada suscripción. Las colas se borran y las suscripciones se suprimen cuando se ejecuta o vuelve a ejecutar el script.

**Nota:** La opción REPLACE no sustituye las propiedades TOPICOBJ o TOPICSTR de una suscripción. TOPICOBJ o TOPICSTR son las propiedades que cambian normalmente en el ejemplo para probar distintas suscripciones. Para cambiarlas, suprima primero la suscripción.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

*Figura 39. Cree suscripciones comodín: wildsubs.tst*

Cree suscripciones que hagan referencia a los objetos de tema de clúster.

**Nota:**

El delimitador, /, se inserta automáticamente entre la serie de tema a la que hace referencia TOPICOBJ y la serie de tema definida por TOPICSTR.

La definición, DEFINE SUB (FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) crea la misma suscripción. TOPICOBJ se utiliza como método rápido para hacer referencia a la serie de tema que ya se ha definido. La suscripción, cuando se crea, ya no hace referencia al objeto de tema.

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

*Figura 40. Suprima y cree suscripciones: fullsubs.tst*

Cree un clúster con dos depósitos. Cree dos repositorios parciales para publicación y suscripción. Vuelva a ejecutar el script para suprimirlo todo y empezar de nuevo. El script también crea la jerarquía de temas y las suscripciones comodín iniciales.

**Nota:**

En otras plataformas, escriba un script parecido o escriba todos los mandatos. El uso de un script permite suprimirlo todo con mayor rapidez y empezar de nuevo con una configuración idéntica.

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

Figura 41. Cree gestores de colas: *qmgrs.bat*

Actualice la configuración añadiendo las suscripciones a los temas de clúster.

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

Figura 42. Actualice las suscripciones: *upsubs.bat*

Ejecute *pub.bat*, con un gestor de colas como parámetro, para publicar mensajes que contengan la serie de tema de publicación. *Pub.bat* utiliza el programa de ejemplo **amqspub**.

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

Figura 43. Publique: *pub.bat*

## Conceptos relacionados

[Suscripción de comodín y publicaciones retenidas](#)

## Ámbito de la publicación

Quando se configura un clúster o jerarquía de publicación/suscripción, el ámbito de una publicación controla si los gestores de colas reenvían una publicación a gestores de colas remotos. Utilice el atributo de tema **PUBSCOPE** para administrar el ámbito de las publicaciones.

Si una publicación no se reenvía a los gestores de colas remotos, sólo los suscriptores locales recibirán la publicación.

Quando se utiliza un clúster de publicación/suscripción, el ámbito de las publicaciones está controlado principalmente por la definición de objetos de tema en clúster en determinados puntos del árbol de temas. Debe configurarse el ámbito de publicación para que permita el flujo de publicaciones a otros gestores de colas del clúster. Solo debería restringirse el ámbito de publicación de un tema en clúster cuando se necesita un control de grano fin de determinados temas en gestores de colas concretos.

Cuando se utiliza una jerarquía de publicación/suscripción, el ámbito de las publicaciones está controlado principalmente por este atributo junto con el atributo de ámbito de suscripción.

El atributo **PUBSCOPE** se utiliza para determinar el ámbito de publicaciones realizadas en un tema específico. Puede establecer el atributo en uno de los valores siguientes:

#### **QMGR**

La publicación se entrega únicamente a los suscriptores locales. Estas publicaciones se denominan *publicaciones locales*. Las publicaciones locales no se reenvían a gestores de colas remotos y, por lo tanto, los suscriptores conectados a gestores de colas remotos no las reciben.

#### **TODOS**

La publicación se entrega a suscriptores locales y a suscriptores conectados a gestores de colas remotos en un clúster o jerarquía de publicación/suscripción. Estas publicaciones se denominan *publicaciones globales*.

#### **ASPARENT**

Utilice el valor **PUBSCOPE** del tema padre en el árbol de temas.

Los publicadores también pueden especificar si una publicación es local o global utilizando la opción de transmitir mensajes MQPMO\_SCOPE\_QMGR. Si se utiliza esta opción, altera temporalmente cualquier comportamiento que se ha establecido utilizando el atributo de tema **PUBSCOPE**.

#### **Conceptos relacionados**

“Objetos de tema administrativo” en la página 78

Mediante un objeto de tema administrativo puede asignar a los temas atributos específicos que no son predeterminados.

#### **Tareas relacionadas**

Configuración de redes de publicación/suscripción distribuidas

## **Ámbito de la suscripción**

El ámbito de una suscripción controla si una suscripción en un gestor de cola, recibe publicaciones publicadas en otro gestor de colas en una jerarquía o clúster de publicación/suscripción o sólo publicaciones desde publicadores locales.

Limitar el ámbito de suscripción a un gestor de colas detiene las suscripciones de proxy para que se reenvíen a otros gestores de cola en la topología de publicación/suscripción. Esto reduce el tráfico de mensajería de publicación/suscripción entre gestores de cola.

Cuando se utiliza un clúster de publicación/suscripción, el ámbito de las suscripciones está controlado principalmente por la definición de objetos de tema en clúster en determinados puntos del árbol de temas. Debe configurarse el ámbito de suscripción para que permita el flujo de suscripciones de proxy a otros gestores de colas del clúster. Solo debería restringirse el ámbito de suscripción de un tema en clúster cuando se necesita un control de grano fin de determinados temas en gestores de colas concretos.

Cuando se utiliza una jerarquía de publicación/suscripción, el ámbito de las suscripciones está controlado principalmente por este atributo junto con el atributo de ámbito de publicación.

El atributo de tema **SUBSCOPE** se utiliza para determinar el ámbito de las suscripciones realizadas a un tema específico. Puede establecer el atributo en uno de los valores siguientes:

#### **QMGR**

Una suscripción sólo recibe publicaciones locales y las suscripciones de proxy no se propagan a gestores de cola remotos.

#### **TODOS**

Una suscripción proxy se propaga a gestores de colas remotos en una jerarquía o clúster de publicación/suscripción y el suscriptor recibe publicaciones locales y remotas.

#### **ASPARENT**

Utilice el valor **SUBSCOPE** del tema padre en el árbol de temas.

Cuando el ámbito de suscripción para un tema se establece en ALL, ya sea directamente o resuelto a través de ASPARENT, las suscripciones individuales a ese tema pueden restringir su ámbito a QMGR especificando MQSO\_SCOPE\_QMGR al crear la suscripción. Una suscripción a un tema que tenga un ámbito de QMGR no puede ampliar el ámbito a ALL.

### Conceptos relacionados

[“Objetos de tema administrativo” en la página 78](#)

Mediante un objeto de tema administrativo puede asignar a los temas atributos específicos que no son predeterminados.

### Tareas relacionadas

[Configuración de redes de publicación/suscripción distribuidas](#)

## Espacios de temas

Un espacio de temas es un conjunto de temas al que uno puede suscribirse y en el que se puede publicar. Un gestor de colas en una topología de publicación/suscripción distribuida tiene un espacio de temas que en potencia incluye temas que se han suscrito y en los que se ha publicado en los gestores de colas conectados de dicha topología.

**Nota:** Para obtener una descripción general de los temas en un gestor de colas como, por ejemplo, objetos de tema administrativos, cadenas de tema y árboles de temas, consulte [“Temas” en la página 69](#). En adelante, cuando se mencione *temas* en este artículo se hará referencia a *cadenas de tema* a menos que se indique otra cosa.

Los temas se crean inicialmente de alguna de las maneras siguientes:

- administrativamente, cuando se define un objeto de tema o una suscripción duradera
- dinámicamente, cuando una aplicación crea dinámicamente una publicación o una suscripción en un tema nuevo.

Los temas se propagan a otros gestores de colas tanto a través de suscripciones proxy como mediante la creación de objetos de tema de clúster administrativo. Las suscripciones proxy dan como resultado que se reenvían las publicaciones desde el gestor de colas al que un publicador está conectado, a los gestores de colas de suscriptores.

Las suscripciones proxy se propagan entre todos los gestores de colas que están conectados entre sí mediante relaciones padre-hijo en una jerarquía de gestor de colas. El resultado es que puede suscribirse en un gestor de colas a un tema definido en cualquier otro gestor de colas de la jerarquía. Siempre que haya una vía de acceso conectada entre los gestores de colas, no importa cómo están conectados los gestores de colas.

Las suscripciones de proxy también se propagan en las suscripciones a temas de clúster en un clúster de publicación/suscripción. Un tema de clúster es un tema que está conectado a un objeto de tema que tiene el atributo **CLUSTER**, o hereda el atributo de su padre. Los temas que no son temas de clúster son conocidos como temas locales y no se replican en el clúster. No se propaga ninguna suscripción proxy al clúster desde suscripciones a temas locales.

En resumen, las suscripciones proxy se crean para los suscriptores en dos circunstancias.

1. Un gestor de colas es un miembro de una jerarquía, y una suscripción proxy se reenvía al padre e hijos del gestor de colas.
2. Un gestor de colas es un miembro de un clúster y la serie de tema de suscripción se resuelve en un tema que está asociado a un objeto de tema de clúster. Cuando el tema es un tema de clúster de *direccionamiento directo*, las suscripciones de proxy se reenvían a todos los miembros del clúster. Cuando el tema es un tema de clúster de *direccionamiento de host de temas*, las suscripciones de proxy solo se reenvían a los gestores de colas del clúster que hayan definido el objeto de tema en clúster. Para obtener más información, consulte [“Clústeres de publicación/suscripción” en la página 93](#).

Si un gestor de colas es un miembro de un clúster y una jerarquía, las suscripciones proxy son propagadas por ambos mecanismos sin entregar publicaciones duplicadas al suscriptor.

Los espacios de temas de tres topologías de publicación/suscripción se describen en la lista siguiente:

- [“Caso 1. Clústeres de publicación/suscripción”](#) en la página 107.
- [“Caso 2. Jerarquías de publicación/suscripción”](#) en la página 107.

En temas independientes, las tareas de configuración siguientes describen cómo combinar espacios de temas.

- [Creación de un único espacio de temas en un clúster de publicación/suscripción.](#)
- [Combinación de espacios de temas de múltiples clústeres.](#)
- [Combinación y aislamiento de espacios de temas en múltiples clústeres.](#)
- [Publicación y suscripción en espacios de temas en múltiples clústeres.](#)

### **Caso 1. Clústeres de publicación/suscripción**

En el ejemplo, presuponga que el gestor de colas no está conectado a una jerarquía de publicación/suscripción.

Si un gestor de colas es miembro de un clúster de publicación/suscripción, su espacio de temas se compone de temas locales y temas de clúster. Los temas locales se asocian con objetos de tema sin el atributo **CLUSTER**. Si un gestor de colas tiene definiciones de objetos de tema local, su espacio de temas es diferente de otro gestor de colas del clúster que también tenga sus propios objetos de tema definidos localmente.

En un clúster de publicación/suscripción, no puede suscribirse a un tema definido en otro gestor de colas, a menos que el tema al que se suscribe se resuelve en un objeto de tema de clúster.

Cuando se necesitan las mismas definiciones nombradas de un objeto de tema de clúster en múltiples gestores de colas (por ejemplo, cuando se utiliza un *direccionamiento de host de temas*) es importante que todas las definiciones coincidan donde sea necesario. Puede obtener información adicional consultando [Creación de un único espacio de temas en un clúster de publicación/suscripción](#).

Una definición local de un objeto de tema, si la definición es para un tema de clúster o un tema local, tiene prioridad sobre el mismo objeto de tema definido en otro lugar del clúster. Se utiliza el tema definido localmente, incluso si el objeto definido en otro lugar es más reciente.

Es importante que un objeto de tema de clúster esté asociado con la misma serie de tema en todas partes del clúster. No puede modificar la serie de tema con la que está asociado un objeto de tema. Para asociar el mismo objeto de tema con una serie de tema diferente, debe suprimir el objeto de tema y volver a crearlo con la nueva serie de tema. Si el tema está en clúster, tiene el efecto de suprimir las copias del objeto de tema almacenado en los demás miembros del clúster y, a continuación, crear copias del nuevo objeto de tema en todas partes del clúster. Todas las copias del objeto de tema hacen referencia a la misma serie de tema.

Es posible crear accidentalmente dos definiciones del mismo objeto de tema designado en diferentes gestores de colas en el clúster, con diferentes series de tema. Esto puede dar como resultado un comportamiento confuso, porque varias definiciones del mismo objeto de tema con distintas series de tema pueden producir resultados diferentes en función de cómo y dónde se hace referencia al tema. Consulte [Múltiples definiciones de temas de clúster con el mismo nombre](#) para obtener información sobre este aspecto importante.

### **Caso 2. Jerarquías de publicación/suscripción**

En el ejemplo, presuponga que el gestor de colas no es miembro de un clúster de publicación/suscripción.

En IBM MQ, si un gestor de colas es miembro de una jerarquía de publicación/suscripción, su espacio de temas consta de todos los temas definidos localmente y en gestores de colas conectados. El espacio de temas de todos los gestores de colas de una jerarquía es el mismo. No hay ninguna división de los temas en temas locales y temas de globales.

Establezca una de las opciones **PUBSCOPE** y **SUBSCOPE** en QMGR, para impedir que una publicación en un tema fluya de un publicador a un suscriptor conectado a gestores de colas distintos en la jerarquía.

Supongamos que define un objeto de tema Alabama con la serie de tema USA/Alabama en el gestor de colas QMA. El resultado es el siguiente:

1. Ahora, el espacio de temas en QMA incluye el objeto de tema Alabama y la serie de tema USA/Alabama.
2. Una aplicación o un administrador puede crear una suscripción en QMA utilizando el nombre de objeto de tema Alabama.
3. Una aplicación puede crear una suscripción a cualquier tema, incluido USA/Alabama, en cualquier gestor de colas de la jerarquía. Si QMA no se ha definido localmente, el tema USA/Alabama se resuelve en el objeto de tema SYSTEM.BASE.TOPIC.

### Conceptos relacionados

[“Ámbito de la publicación” en la página 104](#)

Cuando se configura un clúster o jerarquía de publicación/suscripción, el ámbito de una publicación controla si los gestores de colas reenvían una publicación a gestores de colas remotos. Utilice el atributo de tema **PUBSCOPE** para administrar el ámbito de las publicaciones.

[“Ámbito de la suscripción” en la página 105](#)

El ámbito de una suscripción controla si una suscripción en un gestor de cola, recibe publicaciones publicadas en otro gestor de colas en una jerarquía o clúster de publicación/suscripción o sólo publicaciones desde publicadores locales.

### Tareas relacionadas

[Configuración de redes de publicación/suscripción distribuidas](#)

## IBM MQ Multicast

---

IBM MQ Multicast ofrece baja latencia, alta diseminación y mensajería de multidifusión fiable.

La multidifusión es una forma eficaz de mensajería de publicación/suscripción porque se puede escalar a un elevado número de suscriptores sin efectos negativos en el rendimiento. IBM MQ permite una mensajería multidifusión fiable mediante el uso de acuses de recibo, acuses de recibo negativos y números de secuencia para lograr una mensajería de baja latencia y alta diseminación.

La entrega equilibrada de IBM MQ Multicast permite una entrega casi simultánea y garantiza que ningún destinatario tenga ventaja. Dado que IBM MQ Multicast utiliza la red para suministrar mensajes, no es necesario utilizar un motor de publicación/suscripción para diseminar los datos. Después de correlacionar un tema con una dirección de grupo, no es necesario utilizar un gestor de colas, porque los publicadores y los suscriptores pueden operar en una modalidad de igual a igual. Esto permite que la carga se reduzca en los servidores de gestores de colas y el servidor de gestores de colas deja de ser un punto de posibles anomalías.

## Conceptos iniciales sobre la multidifusión

IBM MQ Multicast se puede integrar fácilmente en los sistemas y aplicaciones existentes utilizando el objeto Communication Information (COMMINFO). Dos campos de objeto TOPIC permiten la configuración rápida de objetos TOPIC existentes para dar soporte u omitir el tráfico de multidifusión.

### Objetos necesarios para la multidifusión

La siguiente información ofrece una breve descripción general de los dos objetos necesarios para IBM MQ Multicast:

#### Objeto COMMINFO

El objeto COMMINFO contiene los atributos asociados con la transmisión multidifusión. Para obtener más información sobre los parámetros de objeto COMMINFO, consulte [DEFINE COMMINFO](#).

El único campo de COMMINFO que se DEBE establecer es el nombre del objeto COMMINFO. Este nombre se utiliza para identificar el objeto COMMINFO en un tema. El campo **GRPADDR** del objeto COMMINFO debe comprobarse para asegurarse de que el valor es una dirección de grupo de multidifusión válida.

## Objeto TOPIC

Un tema es el asunto de la información que se publica en un mensaje de publicación/suscripción y un tema se define creando un objeto TOPIC. Para obtener más información sobre los parámetros de objeto TOPIC, consulte [DEFINE TOPIC](#).

Se pueden utilizar temas existentes con la multidifusión cambiando los valores de los parámetros siguientes del objeto TOPIC: **COMMINFO** y **MCAST**.

- **COMMINFO** Este parámetro especifica el nombre del objeto de información de comunicación de multidifusión.
- **MCAST** Este parámetro especifica si la multidifusión está permitida en esta posición del árbol de temas. De forma predeterminada, **MCAST** se establece en **ASPARENT**, lo que significa que el atributo de multidifusión del tema se hereda del padre. Establecer **MCAST** en **HABILITADO** permite el tráfico de multidifusión en este nodo.

## Redes y temas de multidifusión

La siguiente información es una visión general de lo que sucede en las suscripciones con diferentes tipos de suscripción y definición de tema. En todos estos ejemplos se presupone que el parámetro **COMMINFO** del objeto TOPIC se ha establecido en el nombre de un objeto **COMMINFO** válido:

### Tema establecido en multidifusión habilitado

Si el parámetro **MCAST** de la serie de tema se establece en **ENABLED**, las suscripciones de los clientes con capacidad de multidifusión están permitidas y se realiza una suscripción de multidifusión a menos que:

- Sea una suscripción duradera desde un cliente con capacidad de multidifusión.
- Sea una suscripción no gestionada desde un cliente con capacidad de multidifusión.
- Sea una suscripción desde un cliente sin capacidad de multidifusión.

En estos casos, se realiza una suscripción sin capacidad de multidifusión y suscripciones se degradan a una publicación/suscripción normal.

### Tema establecido en multidifusión inhabilitado

Si el parámetro de la serie de tema **MCAST** se establece en **DISABLED**, siempre se realiza una suscripción sin multidifusión y suscripciones se degradan a una publicación/suscripción normal.

### Tema establecido sólo en multidifusión

Si el parámetro **MCAST** de la serie de tema se establece en **ONLY**, las suscripciones de los clientes con capacidad de multidifusión están permitidas y se realiza una suscripción de multidifusión a menos que:

- Es una suscripción duradera: las suscripciones duraderas se rechazan con el código de razón [2436 \(0984\) \(RC2436\): MQRC\\_DURABILITY\\_NOT\\_ALLOWED](#)
- Es una suscripción no gestionada: las suscripciones no gestionadas se rechazan con el código de razón [2046 \(07FE\) \(RC2046\): MQRC\\_OPTIONS\\_ERROR](#)
- Es una suscripción de un cliente sin capacidad multidifusión: estas suscripciones se rechazan con el código de razón [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#)
- Es una suscripción de una aplicación enlazada localmente: estas suscripciones se rechazan con el código de razón [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#)

Windows

Linux

AIX

## MQ Telemetry Visión general de

MQ Telemetry incluye un servicio de telemetría (MQXR) que forma parte de un gestor de colas, los clientes de telemetría que puede escribir usted mismo o descargar de forma gratuita y las interfaces administrativas de línea de mandatos y explorador. La telemetría se refiere a la recopilación de datos de una amplia gama de dispositivos remotos y a la administración de los mismos. Con MQ Telemetry puede integrar la recopilación de datos y el control de dispositivos con aplicaciones web.

El MQ Telemetry es un componente de IBM MQ. La actualización de estas versiones es básicamente la instalación de una versión posterior de IBM MQ.

Las aplicaciones de ejemplo siguen estando disponibles de forma gratuita en Eclipse Paho y MQTT.org. Consulte [Programas de ejemplo de IBM MQ Telemetry Transport](#).

Puesto que MQ Telemetry es un componente de IBM MQ, MQ Telemetry se puede instalar con el producto principal o después de que se haya instalado el producto principal. Para obtener información sobre la migración, consulte [Migración de MQ Telemetry en Windows](#) y [Migración de MQ Telemetry en Linux](#).

Los componentes incluidos en MQ Telemetry son los siguientes:

#### **Canales de telemetría**

Utilice los canales de telemetría para gestionar la conexión de clientes MQTT a IBM MQ. Los canales de telemetría utilizan nuevos objetos de IBM MQ, como `SYSTEM.MQTT.TRANSMIT.QUEUE`, para interactuar con IBM MQ.

#### **Servicio de telemetría (MQXR)**

Los clientes de MQTT utilizan el servicio de telemetría de `SYSTEM.MQXR.SERVICE` para conectarse a canales de telemetría.

#### **Soporte de IBM MQ Explorer para MQ Telemetry**

MQ Telemetry se puede administrar utilizando IBM MQ Explorer.

#### **Documentación**

La documentación de MQ Telemetry se incluye en la documentación estándar del producto IBM MQ. La documentación de SDK para Java y clientes C se suministra en la documentación del producto y como Javadoc y HTML.

## **Conceptos de telemetría**

Puede recopilar información de todo el entorno que le rodea para tomar decisiones sobre qué se debe hacer. Como consumidor puede comprobar qué tiene en el almacén antes de decidir qué comida comprar. Puede conocer cuánto tiempo dura un viaje si se marcha en ese momento, antes de reservar una conexión. Puede comprobar qué síntomas tiene antes de decidir si ir al médico. Puede comprobar cuándo va a llegar el autobús y decidir así si va esperarlo. La información con la que tomará estas decisiones se le presentará directamente en los medidores o dispositivos, de forma impresa o en pantalla. No importa dónde esté ni cuándo necesite esta información, puede recopilar estos datos, reunirlos, analizarlos y actuar según ellos.

Si las fuentes de información están muy dispersas o inaccesibles, resulta difícil y costoso reunir información precisa. Si desea hacer muchos cambios o le resulta difícil hacerlos, éstos no se realizarán o lo harán cuando ya no sean tan efectivos.

¿Qué ocurre si se reduce en gran parte los costes de la recopilación de la información y del control de dispositivos que estén muy dispersos, conectándolos con tecnología digital a Internet? Podrá analizar la información mediante recursos de Internet y de la empresa. Tendrá más oportunidades de tomar decisiones fundamentadas en la información que ha obtenido, y actuar según convenga.

Las tendencias tecnológicas y las presiones económicas y del entorno provocan que se den estos cambios:

1. Se reduce el coste de conexión y control de sensores y mecanismos de acceso gracias a la estandarización y conexión a procesadores digitales de bajo coste.
2. Internet y las tecnologías relacionadas se utilizan cada vez más para conectar dispositivos. En algunos países, los teléfonos móviles han superado a los ordenadores personales en el número de conexiones a aplicaciones de Internet. Otros dispositivos, seguramente, seguirán el mismo camino.
3. Internet y las tecnologías relacionadas facilitan mucho que una aplicación pueda obtener datos. El fácil acceso a los datos está haciendo que se utilicen los análisis de datos para transformar los datos de los sensores en información útil para muchas soluciones.
4. El uso inteligente de los recursos es a menudo una forma más rápida y barata de reducir las emisiones de dióxido de carbono y los costes. Las alternativas: encontrar nuevos recursos o desarrollar nuevas tecnologías para utilizar los recursos existentes pueden ser la solución a largo plazo. A corto plazo, el

desarrollo de nuevas tecnologías o la búsqueda de nuevos recursos resulta a menudo más arriesgado, lento y costoso que mejorar las soluciones con las que ya se cuenta.

## Ejemplo

A continuación se muestra un ejemplo de cómo estas tendencias crean nuevas oportunidades para interactuar de forma inteligente con el entorno.

El Convenio Internacional para la Seguridad de la Vida Humana en el Mar (SOLAS) necesita el Sistema de Identificación Automática (AIS) se despliegue en muchos buques. Es necesario en buques mercantes de más de 300 toneladas y buques de pasajeros. El sistema AIS es ante todo un sistema para evitar colisiones en el tráfico costero. Lo utilizan las autoridades marítimas para supervisar y controlar las aguas costeras.

Entusiastas de todo el mundo están desplegando estaciones de seguimiento AIS de bajo coste y subiendo información sobre el tráfico costero a Internet. Otros están creando aplicaciones que combinan la información de AIS con otra información de Internet. Los resultados los suben a sitios web y los publican a través de Twitter y SMS.

En una aplicación puede combinarse información procedente de estaciones AIS cerca de Southampton con información de propietarios de buques y geográfica. La aplicación suministra información en directo sobre llegadas y salidas de transbordadores a Twitter. Los usuarios habituales que utilizan los transbordadores entre Southampton y la Isla de Wight se suscriben al canal de noticias utilizando Twitter o SMS. Si el canal de información muestra que el transbordador llega tarde, los viajeros pueden retrasar su salida y tomar el transbordador cuando atraque más tarde que la hora de llegada programada.

Para obtener más ejemplos, consulte [“Casos de uso de telemetría”](#) en la página 113.

## Tareas relacionadas

[Instalación del MQ Telemetry](#)

[Administración de MQ Telemetry](#)

[Migración de MQ Telemetry en Windows](#)

[Migración de MQ Telemetry en Linux](#)

[Desarrollo de aplicaciones para MQ Telemetry](#)

[Resolución de problemas de MQ Telemetry](#)

## Referencia relacionada

[Referencia de MQ Telemetry](#)

Windows

Linux

AIX

## Introducción a MQ Telemetry

Cada vez es más común que las personas, las empresas y los gobiernos quieran utilizar MQ Telemetry para interactuar de forma más inteligente con el entorno en el que viven y trabajan. MQ Telemetry conecta todo tipo de dispositivos a Internet y a la empresa, y reduce los costes de la creación de aplicaciones para dispositivos inteligentes.

## ¿Qué es MQ Telemetry?

- Es una característica de IBM MQ que amplía la estructura de mensajería universal que proporciona IBM MQ a un amplio rango de sensores remotos, mecanismos de acceso y dispositivos de telemetría. MQ Telemetry amplía IBM MQ para que pueda interconectar aplicaciones empresariales inteligentes, servicios y responsables de decisiones con redes de dispositivos instrumentados.
- Las partes principales de MQ Telemetry son:

### El servicio de MQ Telemetry (MQXR).

Este servicio se ejecuta dentro del servidor IBM MQ y utiliza el protocolo IBM MQ Telemetry Transport (MQTT) para comunicarse con dispositivos de telemetría.

### **Las aplicaciones MQTT que se desarrollan.**

Estas aplicaciones controlan la información que se transmite entre los dispositivos de telemetría y el gestor de colas de IBM MQ y las acciones que se toman en respuesta a dicha información. Como ayuda para la creación de estas aplicaciones, se utilizan las bibliotecas cliente de MQTT.

### **¿Qué puede hacer por mí?**

- MQTT es un transporte de mensajería abierto que permite crear implementaciones de MQTT para una amplia gama de dispositivos.
- Los clientes MQTT se pueden ejecutar en pequeños dispositivos ligeros que tienen recursos limitados.
- MQTT funciona de forma eficaz en redes en las que el ancho de banda es bajo, donde el coste de enviar datos resulta caro o que pueden ser frágiles.
- La entrega de mensajes está asegurada y es independiente de la aplicación.
- Los programadores de aplicaciones no necesitan tener conocimientos de programación de comunicaciones.
- Los mensajes se pueden intercambiar con otras aplicaciones de mensajería. Estas pueden ser otra aplicación de telemetría, una MQI, JMS o una aplicación de mensajería empresarial.

### **¿Cómo lo utilizo?**

- Se proporcionan scripts de ejemplo que funcionan con una aplicación cliente IBM MQ Telemetry Transport v3 de ejemplo (`mqt-tv3app.jar`). Consulte [Programas de ejemplo de IBM MQ Telemetry Transport](#).
- Utilice IBM MQ Explorer y sus herramientas asociadas para administrar la característica de telemetría de IBM MQ.
- Utilice las bibliotecas como ayuda para la creación de aplicaciones MQTT que se conectan a un gestor de colas y que utilizan la mensajería de publicación/suscripción.
- Distribuya la aplicación y la biblioteca cliente al dispositivo en el que deba ejecutarse la aplicación.

### **¿Cómo funciona?**

- MQTT es un protocolo de publicación y suscripción. Una aplicación de cliente MQTT puede publicar mensajes en un servidor MQTT o bien suscribirse a mensajes enviados por aplicaciones que se conectan a un servidor MQTT.
- Las aplicaciones cliente MQTT utilizan bibliotecas cliente que implementan el transporte de mensajes MQTT.
- Una aplicación de cliente MQTT básica funciona como un cliente MQ estándar pero se puede ejecutar en una variedad mucho más amplia de plataformas y redes.
- El servicio MQ Telemetry (MQXR) convierte un gestor de colas IBM MQ en un servidor MQTT.
- Cuando un gestor de colas de IBM MQ actúa como servidor MQTT, otras aplicaciones que se conectan al gestor de colas pueden suscribirse y recibir los mensajes del cliente de MQTT.
- El gestor de colas actúa como direccionador, distribuyendo los mensajes procedentes de las aplicaciones de publicación a las aplicaciones suscriptoras.
- Los mensajes puede distribuirse entre los diferentes tipos de aplicaciones cliente. Por ejemplo, entre los clientes de telemetría y los clientes JMS.

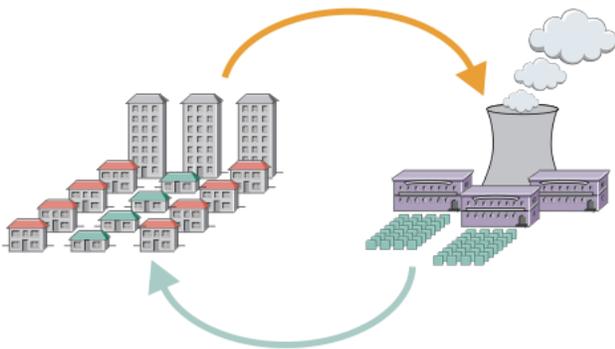
**Nota:** MQ Telemetry sustituye los nodos SCADA que se eliminaron de la versión 7 de WebSphere Message Broker (conocida ahora como IBM Integration Bus) y se ejecuta en Windows, Linux y AIX.

La telemetría es la detección automatizada, la medición de datos y el control de dispositivos remotos. Su principal ventaja es la transmisión de datos desde dispositivos a un punto de control central. La telemetría también incluye el envío de información de configuración y control a los dispositivos.

MQ Telemetry conecta pequeños dispositivos mediante el MQTT protocol y conecta los dispositivos a otras aplicaciones utilizando IBM MQ. MQ Telemetry llena el hueco existente entre los dispositivos e internet facilitando la creación de "soluciones inteligentes". Las soluciones inteligentes ponen a disposición de las aplicaciones que supervisan y controlan dispositivos la abundante información que puede encontrarse en Internet y en las aplicaciones de empresas.

En los diagramas siguientes muestran algunos usos típicos de MQ Telemetry:

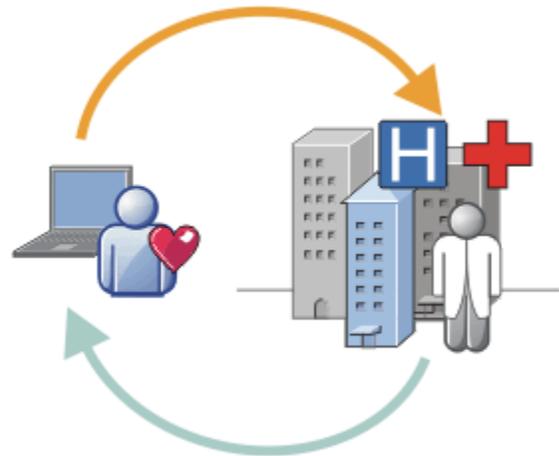
### Telemetría: Electricidad inteligente



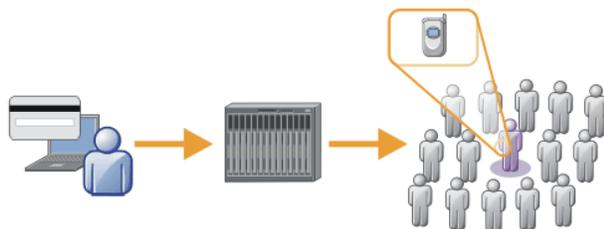
- Mensaje de MQTT que contiene los datos de uso energético enviados al proveedor de servicios.
- MQ Telemetry envía MANDATOS DE CONTROL basados en el análisis de datos de uso energético.
- Para obtener más información, consulte el caso siguiente: [“Caso de uso de telemetría: supervisión y control de la electricidad del hogar”](#) en la [página 115](#)

### Telemetría: Servicios de salud inteligente

- MQ Telemetry envía datos del estado de salud a su Hospital y Doctor.
- Se pueden enviar alertas o comentarios de mensajes MQTT basados en el análisis de datos sanitarios.
- Para obtener más información, consulte el caso siguiente: [“Caso de uso de telemetría: supervisión del paciente en el hogar”](#) en la [página 114](#)



### Telemetría: Uno entre la multitud



- Una simple tarjeta de transacciones se envía al servidor del banco.
- MQ Telemetry identifica a una persona entre miles, alertando al cliente de que se ha utilizado su tarjeta.
- MQ Telemetry puede utilizar la entrada más sencilla de información y localizar a esa persona.

Los casos de uso descritos en los subtemas son ejemplos reales. Ilustran algunas formas de utilizar la telemetría y algunos de los problemas comunes que esta tecnología debe resolver.

## Windows Linux AIX **Caso de uso de telemetría: supervisión del paciente en el hogar**

En la colaboración entre IBM y un proveedor de asistencia médica en un sistema de cuidados cardíacos para el paciente, un desfibrilador cardioversor se comunica con un hospital. Los datos sobre el paciente y el dispositivo implantado se transfieren utilizando la telemetría RF en el dispositivo MQTT situado en casa del paciente.

La transferencia suele realizarse durante la noche, a un transmisor situado junto a la cama. El transmisor transfiere los datos de forma segura a través del teléfono del sistema al hospital, donde se analizan los datos.

El sistema reduce el número de visitas que un paciente debe realizar al médico. Detecta cuándo el paciente o el dispositivo necesitan algún tipo de atención y, en caso de emergencia, avisa al médico de guardia.

La colaboración entre IBM y el proveedor de asistencia médica presenta características que son comunes a un número de casos de uso de telemetría:

### **Imperceptibilidad**

El dispositivo no necesita ninguna intervención del usuario aparte de proporcionarle electricidad, una línea de teléfono y pasar una parte del día cerca del mismo. Su funcionamiento es fiable y es fácil de utilizar.

Para eliminar la necesidad de que el paciente tenga que configurar el dispositivo, el proveedor del dispositivo preconfigura el mismo. El paciente sólo tiene que enchufarlo. El hecho de que el paciente no tenga que configurarlo, simplifica el funcionamiento del dispositivo y reduce la posibilidad de que se configure de forma incorrecta.

El cliente MQTT está integrado como parte del dispositivo. El desarrollador del dispositivo incorpora la implementación del cliente MQTT en el dispositivo y el desarrollador o proveedor configura el cliente MQTT como parte de la preconfiguración.

El cliente MQTT se proporciona como un archivo JAR de Java SE, que el desarrollador incluye en su aplicación Java. En entornos que no son Java, como por ejemplo, éste, el desarrollador del dispositivo puede implementar un cliente con un lenguaje diferente utilizando los formatos y el protocolo MQTT publicados. Como alternativa, el desarrollador puede utilizar uno de los clientes C suministrados como bibliotecas compartidas para plataformas Windows, Linux y ARM.

### **Conexión asimétrica**

La comunicación entre el desfibrilador y el hospital tiene características de red asimétrica. Se utilizan dos redes diferentes para solucionar los diferentes problemas de la recopilación de datos del paciente y su envío al hospital. Entre el paciente y el dispositivo MQTT, se utiliza una red RF de baja potencia y corto alcance. El transmisor se conecta con el hospital utilizando una conexión VPN con TCP/IP a través de una línea telefónica de ancho de banda bajo.

Es a menudo poco viable encontrar una forma de conectar cada dispositivo directamente a una red de protocolo de Internet. Una solución común es utilizar dos redes, conectadas mediante un concentrador. El dispositivo MQTT es un concentrador sencillo, que almacena información del paciente y la reenvía al hospital.

### **Seguridad**

El médico debe poder confiar en la autenticidad de los datos del paciente, y el paciente desea que se respete la privacidad de sus datos.

En algunos casos, basta con cifrar la conexión utilizando VPN o TLS. En otras situaciones, es recomendable mantener la seguridad de los datos incluso después de haberse almacenado.

En ocasiones, el dispositivo de telemetría no es seguro. Por ejemplo, si se comparte una casa. El usuario del dispositivo debe identificarse para garantizar así que los datos provengan del paciente correcto. El propio dispositivo debe autenticarse en el servidor utilizando TLS y el servidor debe hacer lo mismo con el dispositivo.

El canal de telemetría entre el dispositivo y el gestor de colas da soporte a JAAS para la autenticación de usuarios y a TLS para el cifrado de la comunicación y la autenticación del dispositivo. El acceso a una publicación está controlado por el gestor de autorizaciones sobre objetos en IBM MQ.

El identificador que se utiliza para autenticar al usuario puede correlacionarse con un identificador diferente como, por ejemplo, una identidad de paciente común. Un identificador común simplifica la configuración de la autorización en los temas de publicación de IBM MQ.

## Conectividad

La conexión entre el dispositivo MQTT y el hospital utiliza la línea de acceso telefónico y funciona con un ancho de banda bajo de 300 baudios.

Para funcionar correctamente a 300 baudios, el MQTT protocol solamente añade unos cuantos bytes adicionales a un mensaje además de las cabeceras TCP/IP.

El MQTT protocol proporciona una única mensajería de tipo *enviar y olvidar*, que mantiene las latencias bajas. También puede utilizar varias transmisiones para garantizar la entrega *al menos una vez y exactamente una vez* si la entrega garantizada es más importante que el tiempo de respuesta. Para garantizar la entrega, los mensajes se almacenan en el dispositivo hasta que se han entregado de forma satisfactoria. Si el dispositivo está conectado mediante una conexión inalámbrica, resulta especialmente útil la entrega garantizada.

## Escalabilidad

Los dispositivos de telemetría se despliegan normalmente en grandes cantidades, de decenas de miles a millones.

La conexión de tantos dispositivos a un sistema requiere grandes exigencias de una solución. Existen exigencias empresariales como el coste de los dispositivos y del software y exigencias administrativas para la gestión de licencias, dispositivos y usuarios. Entre las exigencias técnicas se incluyen la carga en la red y en los servidores.

La apertura de conexiones consume más recursos de red que el mantenimiento de conexiones abiertas. Sin embargo, en un caso como éste que utiliza líneas de teléfono, el gasto de conexiones reside en no dejar las conexiones abiertas más de lo necesario. Los datos de las transferencias son, principalmente, de tipo de proceso por lotes. Se puede planificar que las conexiones se realicen durante la noche para evitar una franja de hora punta repentina de conexiones a la hora de acostarse.

En el lado del cliente, la mínima configuración necesaria de éste resulta útil para la escalabilidad de los clientes. El cliente MQTT está incorporado en el dispositivo. No existen requisitos de configuración ni debe integrarse ningún paso de aceptación de la licencia cliente MQTT en el despliegue de los dispositivos a los pacientes.

En el servidor, MQ Telemetry tiene un objetivo inicial de 50.000 conexiones abiertas por cada gestor de colas.

Las conexiones se gestionan utilizando IBM MQ Explorer. IBM MQ Explorer filtra las conexiones para que se visualicen en un número gestionable. Si se elige correctamente una planificación de asignaciones de identificadores a clientes, es posible filtrar las conexiones según la ubicación geográfica o alfabéticamente por el nombre del paciente.

## Caso de uso de telemetría: supervisión y control de la electricidad del hogar

Los medidores inteligentes recopilan más detalles sobre el consumo de electricidad que los tradicionales.

Los medidores inteligentes a menudo van unidos a una red local de telemetría para poder supervisar y controlar los diferentes aparatos que hay en la casa. Algunos están conectados de forma remota para que puedan supervisarse y controlarse a distancia.

La conexión remota puede establecerse de forma individual, la puede establecer una compañía de suministro eléctrico, o se puede establecer a través de un punto de control central. El punto de control central puede leer la utilización que se ha hecho de la electricidad y proporcionar datos del uso. Puede proporcionar datos que afectan a la utilización como, por ejemplo, información continuada sobre los costes y las condiciones meteorológicas. Puede limitar la carga para mejorar la eficacia global del suministro de electricidad.

Los medidores inteligentes están empezando a estar muy extendidos. Por ejemplo, el gobierno del Reino Unido está considerando el despliegue de medidores inteligentes en cada hogar del país para 2020.

Los casos de uso sobre mediciones en los hogares tienen una serie de características comunes:

### **Imperceptibilidad**

A menos que el usuario desee expresamente intervenir en el ahorro de electricidad utilizando el medidor, éste no necesita que el usuario intervenga. No debe reducir la fiabilidad del suministro de electricidad a aparatos individuales.

Un cliente MQTT puede estar incorporado en el software desplegado con el medidor y no requiere una instalación o configuración aparte.

### **Conexión asimétrica**

La comunicación entre los aparatos y el medidor inteligente exige diferentes estándares de conectividad entre el medidor y el punto de conexión remoto.

La conexión del medidor inteligente al aparato debe ser de alta disponibilidad y cumplir con estándares de red de área doméstica.

Es probable que la red remota utilice varias conexiones físicas. Algunas de ellas, como un teléfono móvil, tienen un alto coste de transmisión y pueden ser intermitentes. La especificación MQTT v3 está destinada a las conexiones remotas y las conexiones entre adaptadores locales y el medidor inteligente.

La conexión entre los enchufes y los aparatos y el medidor utiliza una red de área doméstica, por ejemplo Zigbee. MQTT para redes de sensores (MQTT-S) se ha diseñado para funcionar con Zigbee y otros protocolos de red con un ancho de banda reducido. MQ Telemetry no ofrece soporte directamente a MQTT-S. Requiere una pasarela para conectar MQTT-S a MQTT v3.

Al igual que ocurre con la supervisión del paciente en el hogar, las soluciones para supervisar y controlar la electricidad del hogar necesitan múltiples redes conectadas que utilizan el medidor inteligente como un concentrador.

### **Seguridad**

Existe una serie de cuestiones sobre seguridad relacionadas con los medidores inteligentes. Entre estas cuestiones se incluyen la no repudiación de la transmisión, la autorización de cualquier acción de control que se inicie y la privacidad de los datos del consumo de electricidad.

Para garantizar la privacidad, los datos transferidos entre el medidor y el punto de control remoto mediante MQTT se pueden cifrar mediante TLS. Para garantizar la autorización de las acciones de control, la conexión de MQTT entre el medidor y el punto de control remoto se puede autenticar mediante TLS.

### **Conectividad**

La naturaleza física de la red remota puede variar considerablemente. Puede utilizar una conexión con ancho de banda existente o una red de móviles con alto coste de llamada y disponibilidad intermitente. Para un coste elevado, intermitente, las conexiones MQTT son un protocolo eficaz y fiable; consulte [“Caso de uso de telemetría: supervisión del paciente en el hogar” en la página 114.](#)

## Escalabilidad

Las compañías eléctricas y puntos de control central planean, con el tiempo, desplegar decenas de millones de medidores inteligentes. En un principio, la cantidad de medidores por despliegue se cifra en decenas de cientos de miles. Este número es comparable con el destino inicial de MQTT de 50.000 conexiones cliente abiertas por cada gestor de colas.

Un aspecto importante de la arquitectura de la supervisión y control de la electricidad en el hogar es utilizar el medidor inteligente como un concentrador de redes. Cada adaptador de cada aparato es un sensor diferente. Al conectarlos a un concentrador local utilizando MQTT, el concentrador puede concentrar los flujos de datos a una única sesión TCP/IP con el punto de control central y almacenar también mensajes durante un corto periodo para superar las interrupciones de la sesión.

Las conexiones remotas deben dejarse abiertas en los casos de uso de electricidad en el hogar por dos razones. En primer lugar, porque se tarda mucho tiempo en abrir las conexiones, en relación al tiempo que se tarda en enviar las solicitudes. El tiempo que se tarda en abrir muchas conexiones para enviar solicitudes de "limitación de carga" en un intervalo corto es muy largo. En segundo lugar, porque para recibir solicitudes de limitación de carga de una compañía de suministro de electricidad, el cliente debe haber abierto la conexión antes. Con MQTT, las conexiones siempre las inicia el cliente y para recibir solicitudes de limitación de carga de la compañía de suministro de electricidad, la conexión debe dejarse abierta.

Si el índice de conexiones abiertas es crítico, o el servidor inicia solicitudes de tiempo crítico, la solución suele ser mantener muchas conexiones abiertas.

## Windows Linux AIX Casos de uso de telemetría: identificación por radiofrecuencia (RFID)

RFID es la utilización de un código RFID incorporado para identificar y realizar seguimiento de un objeto de forma inalámbrica. Los códigos RFID pueden leerse hasta en un rango de varios metros y fuera de la línea de visión del lector RFID. Los códigos pasivo se activan mediante un lector RFID. Los códigos activos transmiten sin activación externa. Los códigos activos deben tener una fuente de alimentación eléctrica. Los códigos pasivos pueden utilizar una fuente de alimentación eléctrica para aumentar su rango de alcance.

RFID se utiliza en muchas aplicaciones, y los tipos de casos de uso varían enormemente. Los casos de uso de RFID y los de supervisión del paciente en el hogar y el de supervisión y control de electricidad en el hogar tienen algunas similitudes y diferencias.

### Imperceptibilidad

En muchos casos de uso, se despliega el lector RFID en grandes cantidades y debe funcionar sin intervención del usuario. El lector incluye un cliente MQTT incorporado para comunicarse con un punto de control central.

Por ejemplo, en un almacén de distribución, un lector utiliza un sensor de movimiento para detectar los palés. Activa los códigos RFID de los elementos que hay en el palé y envía datos y solicitudes a las aplicaciones centrales. Los datos se utilizan para actualizar la ubicación de la mercancía. Las solicitudes controlan lo que ocurre con un palé cercano, por ejemplo, si se está moviendo hacia una bahía determinada. Los sistemas de equipajes de aerolíneas y aeropuertos utilizan de esta forma el RFID.

En algunos casos de uso de RFID, el lector tiene un entorno informático estándar, como Java Platform, Micro Edition (Java ME). En estos casos, el cliente MQTT se puede desplegar en un paso de configuración distinto, después de la fabricación.

### Conexión asimétrica

Los lectores RFID pueden estar separados del dispositivo de control local que contiene un cliente MQTT o bien cada lector puede incorporar un cliente MQTT. La elección de cada tipología depende de factores generales, geográficos y relativos a las comunicaciones.

## Seguridad

La privacidad y autenticidad son asuntos que van unidos a los códigos RFID. Los códigos RFID son discretos y pueden controlarse, falsificarse o ser forzadas de forma secreta.

La solución a los temas de seguridad de RFID aumenta la posibilidad de despliegue de nuevas soluciones RFID. Aunque la seguridad está expuesta en el código RFID y el lector local, la utilización del proceso de información central puede contrarrestar diferentes amenazas. Por ejemplo, puede detectarse un intento de forzar el código relacionando directamente los niveles de mercancía con las entregas y los envíos.

## Conectividad

Las aplicaciones RFID utilizan normalmente el almacenamiento y envío por lotes de información recopilada a partir de lectores RFID y las consultas inmediatas. En el caso de uso del almacén de distribución, el lector RFID está conectado todo el tiempo. Cuando se lee un código, éste queda publicado junto con la información sobre el lector. La aplicación del almacén publica una respuesta y la devuelve al lector.

En la aplicación de almacén, la red suele ser fiable y las solicitudes inmediatas pueden utilizar mensajes de tipo *enviar y olvidar* para conseguir un rendimiento de latencia baja. Los datos que se almacenan y envían por lotes pueden utilizar la mensajería de tipo *exactamente una vez* para minimizar los costes de administración asociados a la pérdida de datos.

## Escalabilidad

Si la aplicación de RFID necesita respuestas inmediatas, en uno o dos segundos, los lectores RFID deben permanecer conectados.

## Casos de uso de telemetría: detección del entorno

La detección del entorno utiliza la telemetría para recoger información sobre los niveles y la calidad de las aguas de los ríos, la contaminación atmosférica y otros datos ambientales.

Los sensores se ubican normalmente en lugares remotos sin acceso a comunicación por cable. Los anchos de banda inalámbricos son caros y poco seguros. Normalmente, se conectan una serie de sensores en una zona geográfica pequeña a un dispositivo de supervisión local que se encuentra en una ubicación segura. Las conexiones locales pueden ser por cable o inalámbricas.

### Imperceptibilidad

Lo normal es que los dispositivos de sensores estén en lugares poco accesibles, tengan poca potencia y estén repartidos en mayores cantidades que los dispositivos de supervisión central. Los sensores son a veces dispositivos "no inteligentes" y el dispositivo de supervisión local incluye adaptadores para transformar y almacenar los datos del sensor. Es probable que el dispositivo de supervisión incorpore un sistema de ámbito general que dé soporte a Java Platform, Standard Edition (Java SE) o Java Platform, Micro Edition (Java ME). Es poco probable que la invisibilidad sea un requisito importante a la hora de configurar el cliente de MQTT.

### Conexión asimétrica

Con las posibilidades de los sensores y el coste y ancho de banda de la conexión remota, se obtiene normalmente un concentrador de supervisión local conectado a un servidor central.

### Seguridad

A menos que la solución sea para uso militar o defensivo, la seguridad no es un requisito muy importante.

### Conectividad

En muchos de los usos no es necesario una supervisión continua o una disponibilidad inmediata de los datos. Los datos de excepción como, por ejemplo, una alerta en el nivel de crecida de un río, sí necesitan enviarse de forma inmediata. Los datos de los sensores se agregan al supervisor local para

reducir los costes de conexión y comunicación, y se transfieren utilizando conexiones planificadas. Los datos de excepción se reenvían al supervisor tan pronto como se detecten.

### Escalabilidad

Los sensores se concentran en torno a concentradores locales y los datos de los sensores se agregan en paquetes que se transmiten según se haya planificado. Ambos factores reducen la carga en el servidor central que se impondría si se utilizaran los sensores conectados directamente.

## Windows Linux AIX **Casos de uso de telemetría: aplicaciones móviles**

Las aplicaciones móviles son aplicaciones que se ejecutan en dispositivos inalámbricos. Los dispositivos pueden ser plataformas de aplicaciones genéricas o dispositivos personalizados.

Las plataformas genéricas incluyen dispositivos de mano, como teléfonos y PDA, y dispositivos portátiles como, por ejemplo, los ordenadores portátiles. Los dispositivos personalizados utilizan hardware especial adaptado a aplicaciones específicas. Un ejemplo de dispositivo móvil personalizado es aquél que se utiliza para entregas de paquetes "firmados". Las aplicaciones para dispositivos móviles personalizados a menudo se crean para una plataforma de software genérica.

### Imperceptibilidad

El despliegue de aplicaciones móviles personalizadas se gestiona y puede incluir la configuración de la aplicación de cliente MQTT. La imperceptibilidad no suele ser un requisito importante al configurar el cliente MQTT.

### Conexión asimétrica

A diferencia de la topología de concentrador local de los casos de uso anteriores, los clientes móviles se conectan de forma remota. La capa de aplicación de cliente se conecta directamente a una aplicación en el concentrador central.

### Seguridad

Cuando existe seguridad física reducida, el dispositivo móvil y su usuario deben autenticarse. Se utiliza TLS para confirmar la identidad del dispositivo, y JAAS para el usuario.

### Conectividad

Si la aplicación móvil depende de la cobertura inalámbrica, debe poder funcionar sin conexión y debe poder hacer frente de forma eficaz a una conexión que se interrumpa. En este entorno, el objetivo es permanecer conectado, pero la aplicación debe poder almacenar y enviar mensajes. A menudo, los mensajes son pedidos o confirmaciones de entregas, por lo que son importantes desde el punto de vista empresarial. Tienen que almacenarse y reenviarse de forma fiable.

### Escalabilidad

La escalabilidad no es un punto importante. La cantidad de clientes de aplicaciones no suelen ser más de varios miles o decenas de miles en los casos de uso de aplicaciones móviles personalizadas.

## Windows Linux AIX **Conexión de dispositivos de telemetría a un gestor de colas**

Los dispositivos de telemetría se conectan a un gestor de colas utilizando un cliente MQTT v3. El cliente MQTT v3 utiliza TCP/IP para conectarse a un escucha TCP/IP denominado servicio de telemetría (MQXR).

Cuando se conecta un dispositivo de telemetría con un gestor de colas, el cliente MQTT inicia una conexión TCP/IP utilizando el método `MqttClient.connect`. Al igual que los clientes IBM MQ un cliente MQTT debe estar conectado al gestor de colas para enviar y recibir mensajes. La conexión se realiza en el servidor utilizando un escucha TCP/IP, instalado con MQ Telemetry, denominado servicio de telemetría (MQXR). Cada gestor de colas ejecuta, como máximo, un servicio de telemetría (MQXR).

El servicio de telemetría (MQXR) utiliza la dirección de socket remoto que establece cada cliente en el método `MqttClient.connect` para asignar la conexión a un canal de telemetría. Una dirección de socket es la combinación de nombre de host TCP/IP y número de puerto. Si varios clientes utilizan la misma dirección de socket remoto, todos estarán conectados al mismo canal de telemetría, a través del servicio de telemetría (MQXR).

Si hay varios gestores de colas en un servidor, divida los canales de telemetría entre los gestores de colas. Asigne las direcciones de socket remoto entre los gestores de colas. Defina cada canal de telemetría con una dirección de socket remoto exclusiva. Dos canales de telemetría no deben utilizar la misma dirección de socket.

Si se configura la misma dirección de socket remoto para canales de telemetría que estén en varios gestores de colas, el primer canal de telemetría en conectarse, gana. Los canales subsiguientes que se conectan en la misma dirección fallan.

Si hay varios adaptadores de red en el servidor, divida las direcciones de socket remoto entre los canales de telemetría. La asignación de direcciones de socket es totalmente arbitraria, siempre que cualquier dirección de socket específica se haya configurado en sólo un canal de telemetría.

Configure IBM MQ para conectar clientes MQTT utilizando los asistentes proporcionados en el suplemento de MQ Telemetry para IBM MQ Explorer. De forma alternativa, siga las instrucciones de [Configuración de un gestor de colas para telemetría en Linux y AIX](#) y [Configuración de un gestor de colas para telemetría en Windows](#) para configurar la telemetría manualmente.

### Referencia relacionada

[Propiedades de MQXR](#)

Windows

Linux

AIX

## Protocolos de conexión de telemetría

MQ Telemetry da soporte a TCP/IP IPv4 y IPv6 y TLS.

Windows

Linux

AIX

## Servicio de telemetría (MQXR)

El servicio de telemetría (MQXR) es un escucha TCP/IP, que se gestiona como un servicio de IBM MQ. Cree el servicio utilizando un asistente de IBM MQ Explorer o con un mandato `runmqsc`.

El servicio de MQ Telemetry (MQXR) se denomina `SYSTEM.MQXR.SERVICE`.

El asistente de configuración de ejemplo de telemetría, que se proporciona en la función de MQ Telemetry para IBM MQ Explorer, crea el servicio de telemetría y un canal de telemetría de ejemplo; consulte [Verificación de la instalación de MQ Telemetry utilizando IBM MQ Explorer](#).

Cree la configuración de ejemplo desde la línea de mandatos; consulte [Verificación de la instalación de MQ Telemetry utilizando la línea de mandatos](#).

El servicio de telemetría (MQXR) se inicia y se detiene automáticamente con el gestor de colas. Controle el servicio utilizando la carpeta de servicios en IBM MQ Explorer. Para ver el servicio, debe pulsar el icono para impedir que IBM MQ Explorer filtre objetos SYSTEM en la pantalla.

Para ver un ejemplo de cómo crear el servicio manualmente, consulte

- [Linux](#) [AIX](#) [Creación de SYSTEM.MQXR.SERVICE en Linux](#).
- [Windows](#) [Creación de SYSTEM.MQXR.SERVICE en Windows](#).

V 9.3.0

A partir de IBM MQ 9.3.0, [Creación de SYSTEM.MQXR.SERVICE en Linux](#) y [Creación de SYSTEM.MQXR.SERVICE en Windows](#) se actualizan para especificar la clave predeterminada para requerir que se cifren las frases de contraseña para los canales TLS MQTT. Para obtener más información, consulte [Cifrado de las frases de contraseña TLS de MQTT](#).

## Canales de telemetría

Cree canales de telemetría para crear conexiones con diferentes propiedades, por ejemplo, JAAS (Java Authentication and Authorization Service) o la autenticación TLS, o para gestionar grupos de clientes.

Cree canales de telemetría utilizando el asistente de **New Telemetry Channel**, proporcionado en la función MQ Telemetry para IBM MQ Explorer. Configure un canal, utilizando el asistente, para aceptar conexiones de clientes MQTT o un puerto TCP/IP específico. Desde IBM WebSphere MQ 7.1, puede configurar MQ Telemetry utilizando el programa de línea de mandatos, **runmqsc**.

Cree varios canales de telemetría en diferentes puertos para gestionar más fácilmente grandes cantidades de conexiones de cliente, dividiendo los clientes en grupos. Cada canal de telemetría tiene un nombre diferente.

Puede configurar los canales de telemetría con atributos de seguridad diferentes para crear tipos de conexiones diferentes. Cree varios canales para aceptar conexiones de cliente en diferentes direcciones TCP/IP. Utilice TLS para cifrar mensajes y autenticar el canal de telemetría y el cliente; consulte [Configuración de TLS de clientes y canales de telemetría de MQTT](#). Especifique el ID de usuario para simplificar el acceso a los objetos de IBM MQ. Especifique una configuración JAAS para autenticar el usuario de MQTT con JAAS; consulte [Identificación, autorización y autenticación de clientes MQTT](#).

## Protocolo IBM MQ Telemetry Transport

El protocolo IBM MQ Telemetry Transport (MQTT) v3 se ha diseñado para intercambiar mensajes entre pequeños dispositivos con un ancho de banda reducido o conexiones caras y para enviar mensajes de forma segura. Utiliza TCP/IP.

El MQTT protocol está publicado; consulte [Formato y protocolo de IBM MQ Telemetry Transport](#). La versión 3 del protocolo utiliza la opción de publicar/suscribir y soporta tres calidades de servicio: *enviar y olvidar*, *al menos una vez* y *exactamente una vez*.

Los mensajes son reducidos gracias al pequeño tamaño de las cabeceras del protocolo y a la carga del mensaje en matriz de bytes. Las cabeceras constan de una cabecera fija de 2 bytes y de hasta 12 bytes de otras cabeceras variables adicionales. El protocolo utiliza 12 bytes de cabeceras variables para suscribirse y conectarse y sólo 2 bytes para cabeceras variables para la mayoría de las publicaciones.

Con tres calidades de servicio, se puede compensar la baja latencia con la fiabilidad; consulte [Calidades de servicio proporcionadas por un cliente del MQTT](#). *Enviar y olvidar* no utiliza ningún almacenamiento de dispositivo permanente y únicamente una transmisión para enviar o recibir una publicación. *Al menos una vez* y *exactamente una vez* requieren el almacenamiento permanente en el dispositivo para mantener el estado del protocolo y guardar un mensaje hasta que se acuse recibo de él.

## Clientes MQTT

Una aplicación de cliente MQTT se encarga de recopilar información del dispositivo de telemetría, conectar con el servidor y publicar la información en el servidor. También puede suscribirse a temas, recibir publicaciones y controlar el dispositivo de telemetría.

A diferencia de las aplicaciones cliente de IBM MQ, las aplicaciones cliente MQTT no son aplicaciones de IBM MQ. No especifican ningún gestor de colas al que conectarse. No se limitan a utilizar interfaces de programación de IBM MQ específicas. En su lugar, los clientes MQTT implementan el protocolo MQTT 3. Puede escribir su propia biblioteca de clientes para intercambiar información con el MQTT protocol en el lenguaje de programación y en la plataforma que desee. Consulte [Formato y protocolo de IBM MQ Telemetry Transport](#).

Para simplificar la escritura de aplicaciones cliente de MQTT, utilice las bibliotecas cliente C, Java y JavaScript que integran el MQTT protocol para diversas plataformas. Si incorpora estas bibliotecas en las apps de MQTT, un cliente MQTT totalmente funcional puede ser tan corto como 15 líneas de código. MQTTLas bibliotecas cliente de están disponibles gratuitamente en Eclipse Paho y MQTT.org. Consulte [Programas de ejemplo de IBM MQ Telemetry Transport](#).

La aplicación de cliente MQTT se encarga siempre de iniciar una conexión con un canal de telemetría. Una vez conectado, la aplicación cliente de MQTT o una aplicación de IBM MQ puede iniciar un intercambio de mensajes.

Las aplicaciones cliente de MQTT y las aplicaciones de IBM MQ se publican y se suscriben al mismo conjunto de temas. Una aplicación de IBM MQ también puede enviar un mensaje directamente a una aplicación cliente de MQTT sin que la aplicación cliente cree primero una suscripción. Consulte [Configurar gestión de colas distribuidas para enviar mensajes a clientes MQTT](#).

Las aplicaciones cliente de MQTT están conectadas a IBM MQ utilizando un canal de telemetría. El canal de telemetría actúa como un puente entre los distintos tipos de mensajes utilizados por MQTT y IBM MQ. Crea publicaciones y suscripciones en el gestor de colas, en nombre de la aplicación de cliente MQTT. El canal de telemetría envía publicaciones que coinciden con las suscripciones de una aplicación de cliente MQTT del gestor de colas a la aplicación de cliente MQTT.

Windows

Linux

AIX

## Envío de un mensaje a un cliente MQTT

Las aplicaciones de IBM MQ pueden enviar mensajes de clientes MQTT v3 publicando en las suscripciones creadas por los clientes o enviando mensajes directamente. Los clientes MQTT pueden enviarse mensajes entre sí publicando en los temas suscritos por otros clientes.

### Un cliente MQTT se suscribe a una publicación, que recibe de IBM MQ

Realice la tarea, [“Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer”](#) en la [página 124](#) para enviar una publicación de IBM MQ a un cliente MQTT.

El modo estándar para que un cliente MQTT v3 reciba mensajes es que cree una suscripción a un tema o conjunto de temas. En el fragmento de código de ejemplo, [Figura 44 en la página 123](#), el cliente de MQTT se suscribe utilizando la serie de tema "MQTT Examples". Una aplicación IBM MQ C, [Figura 45 en la página 123](#), publica en el tema utilizando la serie de tema "MQTT Examples". En el fragmento de código de la [Figura 46 en la página 123](#), el cliente MQTT recibe la publicación en el método de devolución de llamada, `messageArrived`.

Para obtener más información sobre cómo configurar IBM MQ para enviar publicaciones en respuesta a suscripciones de clientes MQTT, consulte [Publicación de un mensaje en respuesta a una suscripción de cliente MQTT](#).

### Una aplicación IBM MQ envía un mensaje directamente a un cliente MQTT

Realice la tarea, [“Envío de un mensaje a un cliente MQTT utilizando IBM MQ Explorer”](#) en la [página 128](#) para enviar un mensaje directamente desde IBM MQ a un cliente MQTT.

Un mensaje enviado de esta manera a un cliente MQTT se denomina un mensaje no solicitado. Los clientes MQTT v3 reciben mensajes no solicitados como publicaciones, con un conjunto de nombres de temas. El servicio de telemetría (MQXR) establece el nombre del tema en el nombre de la cola remota.

Para obtener más información sobre cómo configurar IBM MQ para enviar mensajes directamente a clientes MQTT, consulte [Envío de un mensaje a un cliente directamente](#).

### Un cliente MQTT publica un mensaje

Un cliente MQTT v3 puede publicar un mensaje que recibe otro cliente MQTT v3, pero no puede enviar un mensaje no solicitado. El fragmento de código de la [Figura 47 en la página 124](#) muestra cómo un cliente MQTT v3, escrito en Java, publica un mensaje.

El patrón típico para enviar un mensaje a un cliente MQTT v3 específico, es que cada cliente cree una suscripción a su propio `ClientIdentifier`. Efectúe la tarea [“Publicación de un mensaje en un cliente MQTT v3 específico”](#) en la [página 130](#) para publicar un mensaje de un cliente MQTT a otro cliente MQTT utilizando `ClientIdentifier` como serie de tema.

## Fragmentos de código de ejemplo

En el fragmento de código de la [Figura 44](#) en la [página 123](#) se muestra cómo un cliente MQTT escrito en Java crea una suscripción. También necesita un método de devolución de llamada, `messageArrived` para poder recibir publicaciones para la suscripción.

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int       QoS = 1;
client.subscribe(topicString, QoS);
```

*Figura 44. Suscriptor de cliente MQTT v3*

El fragmento de código de [Figura 45](#) en la [página 123](#) muestra cómo una aplicación IBM MQ escrita en C envía una publicación. El fragmento de código se extrae de la tarea, [Crear un publicador en un tema variable](#).

```
/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);
```

*Figura 45. Publicador de IBM MQ*

Cuando llega la publicación, el cliente MQTT llama al método `messageArrived` de la clase `MqttCallback` del cliente de aplicaciones MQTT.

```
public class Callback implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods
```

*Figura 46. Método de `messageArrived`*

En la [Figura 47](#) en la [página 124](#) se muestra MQTT v3 publicando un mensaje en la suscripción creada en la [Figura 44](#) en la [página 123](#).

```

String      address = "localhost";
String      clientId = String.format("%-23.23s",
                                   System.getProperty("user.name") + "-" +
                                   (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient  client = new MqttClient(address, clientId);
String      topicString = "MQTT Examples";
MqttTopic   topic = client.getTopic(Example.topicString);
String      publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

Figura 47. Publicador de cliente MQTT v3

## Windows Linux AIX **Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer**

Siga los pasos de esta tarea para publicar un mensaje utilizando IBM MQ Explorer y suscríbase a él con el programa de utilidad de cliente de MQTT. En una tarea adicional se le muestra cómo configurar un alias de gestor de colas, en lugar de establecer la cola de transmisión predeterminada en `SYSTEM.MQTT.TRANSMIT.QUEUE`.

### Antes de empezar

La tarea presupone que está familiarizado con IBM MQ y IBM MQ Explorer y que se han instalado IBM MQ y la característica de MQ Telemetry.

El usuario que cree los recursos del gestor de colas para esta tarea deberá tener permisos suficientes para poder hacerlo. A efectos de demostración, se supone que el ID de usuario de IBM MQ Explorer es miembro del grupo `mqm`.

### Acerca de esta tarea

En la tarea, cree un tema en IBM MQ y suscríbase a él utilizando el programa de utilidad de cliente de MQTT. Cuando se publica en el tema utilizando IBM MQ Explorer, el cliente MQTT recibe la publicación.

### Procedimiento

Efectúe una de las tareas siguientes:

- Ha instalado MQ Telemetry, pero todavía no lo ha iniciado. Efectúe la tarea: [“Inicio de la tarea sin haber definido todavía el servicio de telemetría \(MQXR\)”](#) en la página 125.
- Ha ejecutado la telemetría de IBM MQ telemetría antes, pero desea utilizar un nuevo gestor de colas para realizar la demostración. Efectúe la tarea: [“Inicio de la tarea sin haber definido todavía el servicio de telemetría \(MQXR\)”](#) en la página 125.
- Desea efectuar la tarea utilizando un gestor de colas existente que no tiene definidos los recursos de telemetría. No desea ejecutar el asistente **Definir configuración de ejemplo**.
  - a. Efectúe una de las tareas siguientes para configurar la telemetría:
    - [Configuración de un gestor de colas para la telemetría en Linux y AIX](#)
    - [Configuración de un gestor de colas para la telemetría en Windows](#)
  - b. Efectúe la tarea: [“Inicio de la tarea después ejecutar el servicio de telemetría \(MQXR\)”](#) en la página 126
- Si desea efectuar la tarea utilizando un gestor de colas existente que ya tenga definidos los recursos, realice la tarea: [“Inicio de la tarea después ejecutar el servicio de telemetría \(MQXR\)”](#) en la página 126.

## Qué hacer a continuación

Efectúe los pasos indicados en [“Envío de un mensaje a un cliente MQTT utilizando IBM MQ Explorer”](#) en la [página 128](#) para enviar un mensaje directamente al programa de utilidad cliente.

### ***Inicio de la tarea sin haber definido todavía el servicio de telemetría (MQXR)***

Cree un gestor de colas y ejecute el asistente **Definir configuración de ejemplo** para definir recursos de telemetría de ejemplo para el gestor de colas. Publique un mensaje utilizando IBM MQ Explorer y suscríbase a él con el programa de utilidad cliente de MQTT.

## Acerca de esta tarea

Al configurar los recursos de ejemplo mediante el asistente **Definir configuración de ejemplo**, se establecen los permisos del ID de usuario invitado. Considere detenidamente si desea autorizar al ID de usuario invitado de esta forma. A los usuarios guest en Windows y nobody en Linux se les concede permiso para publicar y suscribirse al directorio raíz del árbol de temas y transferir mensajes a SYSTEM.MQTT.TRANSMIT.QUEUE.

El asistente también establece la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE, que puede interferir con las aplicaciones que se ejecuten en un gestor de colas existente. Es posible, pero laborioso, configurar la telemetría y no utilizar la cola de transmisión predeterminada; efectúe lo siguiente en la tarea: [“Utilización de un alias de gestor de colas”](#) en la [página 127](#). En esta tarea, creará un gestor de colas para evitar la posibilidad de interferir en las colas de transmisión predeterminadas existentes.

## Procedimiento

1. Utilizando IBM MQ Explorer, cree e inicie un nuevo gestor de colas.
  - a) Pulse el botón derecho del ratón en la carpeta Queue Managers > **Nuevo > Gestor de colas...**  
Escriba un nombre de gestor de colas > **Finalizar**.  
  
Piense en un nombre de gestor de colas; por ejemplo, MQTTQMGR.
2. Cree e inicie el servicio de telemetría (MQXR), y cree un canal de telemetría de ejemplo.
  - a) Abra la carpeta Queue Managers\QmgrName\Telemetry.
  - b) Pulse **Definir configuración de ejemplo... > Finalizar**  
  
Marque el recuadro de selección **Iniciar programa de utilidad cliente MQTT**.
3. Cree una suscripción para MQTT Example utilizando el programa de utilidad de cliente de MQTT.
  - a) Pulse **Conectar**.  
  
El **Historial del cliente** registra un suceso de Connected .
  - b) Escriba MQTT Example en el campo **Suscripción \ Tema > Suscribirse**.  
  
El **Historial del cliente** registra un suceso de Subscribed .
4. Cree MQTTExampleTopic en IBM MQ.
  - a) Pulse con el botón derecho del ratón en la carpeta Queue Managers\QmgrName\Topics de **MQ Explorer > Nuevo > Tema**.
  - b) Escriba MQTTExampleTopic como **Nombre > Siguiente**.
  - c) Escriba MQTT Example como **Serie de tema > Finalizar**.
  - d) Pulse **Aceptar** para cerrar la ventana de acuse de recibo.
5. Publique Hello World! en el tema MQTT Example utilizando IBM MQ Explorer.
  - a) Pulse la carpeta Queue Managers\QmgrName\Topics en IBM MQ Explorer.
  - b) Pulse el botón derecho del ratón MQTTExampleTopic > **Probar publicación...**
  - c) Escriba Hello World! en el campo **Datos de mensaje > Publicar mensaje > Vaya la ventana de programa de utilidad cliente de MQTT**.

El **Historial del cliente** registra un suceso de Received .

## ***Inicio de la tarea después ejecutar el servicio de telemetría (MQXR)***

Cree un canal de telemetría y un tema. Autorice al usuario a utilizar el tema y la cola de transmisión de telemetría. Publique un mensaje utilizando IBM MQ Explorer y suscríbase a él con el programa de utilidad cliente de MQTT.

### **Antes de empezar**

En esta versión de la tarea, un gestor de colas, *QmgrName*, está definido y en ejecución. Hay un servicio de telemetría (MQXR) definido y se está ejecutando. Es posible que el servicio de telemetría (MQXR) se haya creado manualmente, o al ejecutar el asistente **Definir configuración de ejemplo**.

### **Acerca de esta tarea**

En esta tarea debe configurar un gestor de colas existente para enviar una publicación al programa de utilidad cliente MQTT.

En el paso “1” en la página 126 de la tarea se establece la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE, que puede interferir con las aplicaciones que se ejecuten en un gestor de colas existente. Es posible, pero laborioso, configurar la telemetría y no utilizar la cola de transmisión predeterminada; efectúe lo siguiente en la tarea: “Utilización de un alias de gestor de colas” en la página 127.

### **Procedimiento**

1. Establezca SYSTEM.MQTT.TRANSMIT.QUEUE como la cola de transmisión predeterminada.
  - a) Pulse con el botón derecho del ratón en Queue Managers\*QmgrName* folder > **Propiedades...**
  - b) Pulse **Comunicación** en el navegador.
  - c) Pulse **Seleccionar ...** > Seleccionar SYSTEM.MQTT.TRANSMIT.QUEUE > **Aceptar** > **Aceptar**.
2. Cree un canal de telemetría MQTTExampleChannel para conectar el programa de utilidad cliente de MQTT a IBM MQ e inicie el programa de utilidad cliente de MQTT.
  - a) Pulse con el botón derecho del ratón en la carpeta Queue Managers\*QmgrName* \Telemetry\Channels de **MQ Explorer**> **Nuevo** > **Canal de telemetría...**
  - b) Escriba MQTTExampleChannel en el campo **Nombre de canal** > **Siguiente** > **Siguiente**.
  - c) Cambie el **ID de usuario fijo** en el panel de autorización del cliente al ID de usuario que va a publicar y suscribirse a MQTTExample > **Siguiente**.
  - d) Deje seleccionado **Iniciar programa de utilidad cliente** > **Finalizar**.
3. Cree una suscripción para MQTT Example utilizando el programa de utilidad de cliente de MQTT.
  - a) Pulse **Conectar**.

El **Historial del cliente** registra un suceso de Connected .
  - b) Escriba MQTT Example en el campo **Suscripción \ Tema** > **Suscribir**.

El **Historial del cliente** registra un suceso de Subscribed .
4. Cree MQTTExampleTopic en IBM MQ.
  - a) Pulse con el botón derecho del ratón en la carpeta Queue Managers\*QmgrName*\Topics de **MQ Explorer**> **Nuevo** > **Tema**.
  - b) Escriba MQTTExampleTopic como **Nombre** > **Siguiente**.
  - c) Escriba MQTT Example como **Serie de tema** > **Finalizar**.
  - d) Pulse **Aceptar** para cerrar la ventana de acuse de recibo.
5. Si desea que un usuario, que no está en el grupo mqm, publique y se suscriba al tema MQTTExample, haga lo siguiente:

a) Autorice al usuario a publicar y suscribirse al tema MQTTExampleTopic:

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

b) Autorice al usuario a transferir un mensaje a la cola SYSTEM.MQTT.TRANSMIT.QUEUE:

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. Publique Hello World! en el tema MQTT Example utilizando IBM MQ Explorer.

a) Pulse la carpeta Queue Managers\QmgrName\Topics en IBM MQ Explorer.

b) Pulse el botón derecho del ratón MQTTExampleTopic > **Probar publicación...**

c) Escriba Hello World! en el campo **Datos de mensaje** > **Publicar mensaje** > Vaya la ventana de programa de utilidad cliente de MQTT.

El **Historial del cliente** registra un suceso de Received .

### **Utilización de un alias de gestor de colas**

Publique un mensaje en el programa de utilidad cliente de MQTT utilizando IBM MQ Explorer sin establecer la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE.

La tarea es una continuación de la tarea anterior, y en ella se utiliza un alias de gestor de colas para evitar establecer la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE.

### **Antes de empezar**

Complete la tarea [“Inicio de la tarea sin haber definido todavía el servicio de telemetría \(MQXR\)”](#) en la [página 125](#) o la tarea [“Inicio de la tarea después ejecutar el servicio de telemetría \(MQXR\)”](#) en la [página 126](#).

### **Acerca de esta tarea**

Cuando un cliente de MQTT crea una suscripción, IBM MQ envía su respuesta utilizando ClientIdentifier, como nombre del gestor de colas remoto. En esta tarea, utiliza ClientIdentifier, MyClient.

Si no hay ninguna cola de transmisión o alias de gestor de colas llamado MyClient, la respuesta se coloca en la cola de transmisión predeterminada. Cuando se establece la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE, el cliente de MQTT obtiene la respuesta.

Puede evitar establecer la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE utilizando los alias de gestor de colas. Debe configurar un alias de gestor de colas para cada ClientIdentifier. Normalmente, hay demasiados clientes para que resulte práctico utilizar los alias de gestor de colas. A menudo ClientIdentifier es imprevisible, haciendo que sea imposible configurar la telemetría esta forma.

Sin embargo, en algunas circunstancias puede que tenga que configurar la cola de transmisión predeterminada en un valor distinto de SYSTEM.MQTT.TRANSMIT.QUEUE. Los pasos de [Procedimiento configuran un alias de gestor de colas en lugar de establecer la cola de transmisión predeterminada en SYSTEM.MQTT.TRANSMIT.QUEUE](#).

### **Procedimiento**

1. Elimine SYSTEM.MQTT.TRANSMIT.QUEUE como la cola de transmisión predeterminada.

a) Pulse con el botón derecho del ratón en Queue Managers\QmgrName folder > **Propiedades...**

b) Pulse **Comunicación** en el navegador.

c) Quite SYSTEM.MQTT.TRANSMIT.QUEUE del campo **Cola de transmisión predeterminado** > **Aceptar**.

2. Compruebe que ya no pueda crear una suscripción con el programa de utilidad cliente MQTT.

- a) Pulse **Conectar**.  
El **Historial del cliente** registra un suceso de Connected .
- b) Escriba MQTT Example en el campo **Suscripción \ Tema > Suscribirse**.  
El **Historial de cliente** registra un suceso Subscribe failed y un suceso Connection lost .
3. Cree un alias de gestor de colas para el ClientIdentifier, MyClient.
  - a) Pulse con el botón derecho del ratón en la carpeta Queue Managers\QmgrName\Queues > **Nuevo > Definición de cola remota**.
  - b) Llame a la definición, MyClient > **Siguiente**.
  - c) Escriba MyClient en el campo **Gestor de colas remoto**.
  - d) Escriba SYSTEM.MQTT.TRANSMIT.QUEUE en el campo **Cola de transmisión > Finalizar**.
4. Vuelva a conectar el programa de utilidad cliente MQTT.
  - a) Compruebe que el **ClientIdentifier** se haya establecido en MyClient.
  - b) **Conectar**  
El **Historial del cliente** registra un suceso de Connected .
5. Cree una suscripción para MQTT Example utilizando el programa de utilidad de cliente de MQTT.
  - a) Pulse **Conectar**.  
El **Historial del cliente** registra un suceso de Connected .
  - b) Escriba MQTT Example en el campo **Suscripción \ Tema > Suscribirse**.  
El **Historial del cliente** registra un suceso de Subscribed .
6. Publique Hello World! en el tema MQTT Example utilizando IBM MQ Explorer.
  - a) Pulse la carpeta Queue Managers\QmgrName\Topics en IBM MQ Explorer.
  - b) Pulse el botón derecho del ratón MQTTExampleTopic > **Probar publicación...**
  - c) Escriba Hello World! en el campo **Datos de mensaje > Publicar mensaje > Vaya la ventana de programa de utilidad cliente de MQTT**.  
El **Historial del cliente** registra un suceso de Received .

## Windows Linux AIX Envío de un mensaje a un cliente MQTT utilizando IBM MQ Explorer

Envíe un mensaje al programa de utilidad cliente de MQTT transfiriendo un mensaje a una cola de IBM MQ utilizando IBM MQ Explorer. En la tarea se muestra cómo configurar una definición de cola remota para enviar un mensaje directamente a un cliente MQTT.

### Antes de empezar

Lleve a cabo la tarea [“Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer”](#) en la [página 124](#). Deje el programa de utilidad cliente MQTT conectado.

### Acerca de esta tarea

En la tarea se muestra cómo enviar un mensaje a un cliente MQTT utilizando la cola en lugar de publicar en un tema. No se crea ninguna suscripción en el cliente. En el paso [“2”](#) en la [página 129](#) de la tarea se muestra que se ha suprimido la suscripción anterior.

### Procedimiento

1. Elimine las suscripciones existentes desconectando y volviendo a conectar el programa de utilidad cliente MQTT.

La suscripción se elimina porque, a menos que se cambien los valores predeterminados, el programa de utilidad cliente MQTT se conecta con una sesión limpia; consulte [Limpiar sesiones](#).

Para que le resulte más fácil realizar la tarea, escriba su propio `ClientIdentifier`, en lugar de utilizar el `ClientIdentifier` generado creado por el programa de utilidad cliente MQTT.

- a) Pulse **Desconectar** para desconectar el programa de utilidad cliente MQTT del canal de telemetría.

El **Historial del cliente** registra un suceso de `Disconnected`

- b) Cambie el valor de **Identificador de cliente** por `MyClient`.
- c) Pulse **Conectar**.

El **Historial del cliente** registra un suceso de `Connected`

2. Compruebe que el programa de utilidad cliente MQTT ya no recibe la publicación para `MQTTExampleTopic`.

- a) Pulse la carpeta `Queue Managers\QmgrName\Topics` en IBM MQ Explorer.
- b) Pulse el botón derecho del ratón `MQTTExampleTopic` > **Probar publicación...**
- c) Escriba `Hello World!` en el campo **Datos de mensaje** > **Publicar mensaje** > Vaya la ventana de programa de utilidad cliente de MQTT.

No se registra ningún suceso en el **Historial del cliente**.

3. Cree una definición de cola remota para el cliente.

Establezca `ClientIdentifier`, `MyClient`, como el nombre del gestor de colas remoto, en la definición de la cola remota. Utilice el nombre que desee como nombre de la cola remota. El nombre de la cola remota se pasa a un cliente MQTT como nombre de tema.

- a) Pulse con el botón derecho del ratón en la carpeta `Queue Managers\QmgrName\Queues` > **Nuevo** > **Definición de cola remota**.
- b) Llame a la definición, `MyClientRemoteQueue` > **Siguiente**.
- c) Escriba `MQTTExampleQueue` en el campo **Cola remota**.
- d) Escriba `MyClient` en el campo **Gestor de colas remoto**.
- e) Escriba `SYSTEM.MQTT.TRANSMIT.QUEUE` en el campo **Cola de transmisión** > **Finalizar**.

4. Transfiera un mensaje de prueba a `MyClientRemoteQueue`.

- a) Pulse el botón derecho del ratón en **MyClientRemoteQueue** > **Transferir mensaje de prueba...**
- b) Escriba `Hello queue!` en el campo **Datos de mensaje** > **Transferir mensaje** > **Cerrar**

El **Historial del cliente** registra un suceso de `Received`.

5. Elimine `SYSTEM.MQTT.TRANSMIT.QUEUE` como la cola de transmisión predeterminada.

- a) Pulse con el botón derecho del ratón en `Queue Managers\QmgrName folder` > **Propiedades...**
- b) Pulse **Comunicación** en el navegador.
- c) Quite `SYSTEM.MQTT.TRANSMIT.QUEUE` del campo **Cola de transmisión predeterminado** > **Aceptar**.

6. Vuelva a efectuar el paso “4” en la [página 129](#).

`MyClientRemoteQueue` es una definición de cola remota que nombra explícitamente la cola de transmisión. No necesita una para definir la cola de transmisión predeterminada para enviar un mensaje a `MyClient`.

## Qué hacer a continuación

Con la cola de transmisión predeterminada ya no establecida en `SYSTEM.MQTT.TRANSMIT.QUEUE`, el programa de utilidad cliente de MQTT no puede crear una nueva suscripción a menos que se haya definido un alias de gestor de colas para el `ClientIdentifier`, `MyClient`. Restaure la cola de transmisión predeterminada en `SYSTEM.MQTT.TRANSMIT.QUEUE`.

### v3 específico

Publique un mensaje de un cliente MQTT v3 a otro, utilizando `ClientIdentifier` como nombre de tema y IBM MQ como intermediario de publicación/suscripción.

#### Antes de empezar

Lleve a cabo la tarea “Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer” en la [página 124](#). Deje el programa de utilidad cliente MQTT conectado.

#### Acerca de esta tarea

La tarea demuestra dos cosas:

1. La suscripción a un tema de un cliente MQTT y la recepción de una publicación de otro cliente MQTT.
2. La configuración de suscripciones "punto a punto" utilizando el `ClientIdentifier` como la serie de tema.

#### Procedimiento

1. Elimine las suscripciones existentes desconectando y volviendo a conectar el programa de utilidad cliente MQTT.

La suscripción se elimina porque, a menos que se cambien los valores predeterminados, el programa de utilidad cliente MQTT se conecta con una sesión limpia; consulte [Limpiar sesiones](#).

Para que le resulte más fácil realizar la tarea, escriba su propio `ClientIdentifier`, en lugar de utilizar el `ClientIdentifier` generado creado por el programa de utilidad cliente MQTT.

- a) Pulse **Desconectar** para desconectar el programa de utilidad cliente MQTT del canal de telemetría.

El **Historial del cliente** registra un suceso de `Disconnected`

- b) Cambie el valor de **Identificador de cliente** por `MyClient`.
- c) Pulse **Conectar**.

El **Historial del cliente** registra un suceso de `Connected`

2. Cree una suscripción al tema `MyClient`

`MyClient` es el `ClientIdentifier` de este cliente.

- a) Escriba `MyClient` en el campo **Suscripción\tema** > **Suscribir**.

El **Historial del cliente** registra un suceso de `Subscribed`.

3. Inicie otro programa de utilidad MQTT.

- a) Abra la carpeta `Queue Managers\QmgrName\Telemetry\channels`.

- b) Pulse el botón derecho del ratón en el canal **PlainText** > **Ejecutar programa de utilidad cliente de MQTT...**

- c) Pulse **Conectar**.

El **Historial del cliente** registra un suceso de `Connected`

4. Publique `Hello MyClient!` en el tema `MyClient`.

- a) Copie el tema de suscripción, `MyClient`, del programa de utilidad cliente de MQTT que se ejecuta con `ClientIdentifier`, `MyClient`.

- b) Pegue `MyClient` en el campo **Publicación\Tema** de cada una de las instancias del programa de utilidad cliente de MQTT.

- c) Escriba `Hello MyClient!` en el campo **Publicación\ mensaje**.

- d) Pulse **Publicar** en ambas instancias.

## Resultados

El **Historial del cliente** en el programa de utilidad cliente de MQTT con `ClientIdentifier`, `MyClient`, registra dos sucesos **Recibidos** y un suceso **Publicado**. La otra instancia del programa de utilidad cliente MQTT registra un suceso **Publicado**.

Si solo ve un suceso del tipo **Recibido**, compruebe las posibles causas siguientes:

1. ¿Se ha establecido la cola de transmisión predeterminada del gestor de colas en `SYSTEM.MQTT.TRANSMIT.QUEUE`?
2. ¿Ha creado alias de gestor de colas o definiciones de colas remotas que hagan referencia a `MyClient` al realizar los otros ejercicios? En caso de que tenga un problema de configuración, suprima los recursos que hagan referencia a `MyClient`, como, por ejemplo, un alias de gestor de colas o colas de transmisión. Desconecte los programas de utilidad de cliente, detenga y reinicie el servicio de telemetría (MQXR).

## Windows Linux AIX Envío de un mensaje a una aplicación de IBM MQ desde un cliente MQTT

Una aplicación de IBM MQ puede recibir un mensaje de un cliente MQTT v3 suscribiéndose a un tema. El cliente MQTT se conecta a IBM MQ utilizando un canal de telemetría y envía un mensaje a la aplicación de IBM MQ publicándolo en el mismo tema.

Realice la tarea, [“Publicación de un mensaje en IBM MQ desde un cliente MQTT”](#) en la [página 131](#), para aprender a enviar una publicación de un cliente MQTT a una suscripción definida en IBM MQ.

Si el tema está en un clúster o se ha distribuido mediante la jerarquía de publicación/suscripción, la suscripción puede estar en un gestor de colas diferente del gestor de colas al que el cliente MQTT está conectado.

## Windows Linux AIX Publicación de un mensaje en IBM MQ desde un cliente MQTT

Cree una suscripción a un tema utilizando IBM MQ Explorer y publique en el tema utilizando el programa de utilidad cliente de MQTT..

### Antes de empezar

Lleve a cabo la tarea [“Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer”](#) en la [página 124](#). Deje el programa de utilidad cliente MQTT conectado.

### Acerca de esta tarea

En la tarea se muestra la publicación de un mensaje con un cliente MQTT y la recepción de la publicación utilizando una suscripción duradera no gestionada que se ha creado utilizando IBM MQ Explorer.

### Procedimiento

1. Cree una suscripción duradera para la serie de tema MQTT `Example`.  
Realice los pasos siguientes para crear la cola y la suscripción utilizando IBM MQ Explorer.
  - a) Pulse con el botón derecho del ratón en la carpeta `Queue Managers\QmgrName\Queues` en IBM MQ Explorer > **Nuevo** > **Cola local...**
  - b) Escriba `MQTTExampleQueue` como nombre de cola > **Finalizar**.
  - c) Pulse con el botón derecho del ratón en la carpeta `Queue Managers\QmgrName\Subscriptions` en IBM MQ Explorer > **Nuevo** > **Suscripción....**
  - d) Escriba `MQTTExampleSubscription` como nombre de cola > **Siguiente**.
  - e) Pulse **Seleccionar...** > `MQTTExampleTopic` > **Aceptar**.

Ya ha creado el tema `MQTTExampleTopic` en el paso “4” en la página 125 de “[Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer](#)” en la página 124.

f) Escribo `MQTTExampleQueue` como nombre de destino > **Finalizar**.

2. Como paso opcional, establezca la cola para que la utilice un usuario distinto, sin autorización de `mqm`.

Si está realizando la configuración para usuarios con menos autorización que `mqm`, debe otorgar autorización `put` y `get` a `MQTTExampleQueue`. El acceso al tema y a la cola de transmisión se ha configurado en “[Publicación de un mensaje en el programa de utilidad cliente MQTT de IBM MQ Explorer](#)” en la página 124.

a) Autorice a un usuario a colocar (`put`) y acceder a la cola (`get`) `MQTTExampleQueue`:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Publique `Hello IBM MQ!` en el tema `MQTT Example` utilizando el programa de utilidad cliente `MQTT`.

Si no ha dejado el programa de utilidad cliente `MQTT` conectado, pulse el botón derecho (del ratón) en el canal **PlainText** > **Ejecutar el programa de utilidad cliente MQTT...** > **Conectar**.

a) Escriba `MQTT Example` en el campo **Publicación\Tema**.

b) Escriba `Hello IBM MQ!` en el campo **Publicación\Mensaje** > **Publicar**.

4. Abra la carpeta `Queue Managers\QmgrName\Queues` y busque `MQTTExampleQueue`.

El valor del campo **Profundidad de cola actual** es 1.

5. Pulse con el botón derecho `MQTTExampleQueue` > **Examinar mensajes ...** y examine la publicación.

Windows

Linux

AIX

## Aplicaciones de publicación/suscripción MQTT

Utilice la publicación/suscripción basada en temas para escribir aplicaciones MQTT.

Cuando el cliente MQTT está conectado, las publicaciones fluyen en ambas direcciones entre el cliente y el servidor. Cuando la información se publica en el cliente, entonces se envían las publicaciones desde éste. El cliente recibe las publicaciones cuando se publica un mensaje en un tema que coincide con una suscripción que haya creado el cliente.

El intermediario de publicación/suscripción de IBM MQ gestiona los temas y suscripciones creados por los clientes MQTT. Los temas creados por los clientes MQTT comparten el mismo espacio de temas que los temas creados por las aplicaciones de IBM MQ.

Las publicaciones que coinciden con la serie de tema en una suscripción de cliente MQTT se colocan en `SYSTEM.MQTT.TRANSMIT.QUEUE` con el nombre del gestor de colas remoto establecido en el `ClientIdentifier` del cliente. El servicio de telemetría (`MQXR`) reenvía las publicaciones al cliente que ha creado la suscripción. Utiliza `ClientIdentifier`, que se ha establecido como el nombre del gestor de colas remoto para identificar al cliente.

Normalmente, se tiene que haber definido `SYSTEM.MQTT.TRANSMIT.QUEUE` como la cola de transmisión predeterminada. Es posible, aunque laborioso, para configurar MQTT no utilizar la cola de transmisión predeterminada; consulte [Configurar gestión de colas distribuidas para enviar mensajes a clientes MQTT](#).

Un cliente MQTT puede crear una sesión persistente; consulte “[Sesiones de MQTT con estado y sin estado](#)” en la página 136. Las suscripciones que se crean en una sesión persistente son duraderas. Las publicaciones que llegan para un cliente que tiene una sesión persistente se almacenan en `SYSTEM.MQTT.TRANSMIT.QUEUE` y se reenvían al cliente cuando éste se vuelve a conectar.

Un cliente MQTT también puede publicar y suscribirse a las publicaciones retenidas; consulte [Publicaciones retenidas y clientes MQTT](#). Un suscriptor de un tema de publicación retenida recibe la publicación más reciente al tema. El suscriptor recibe la publicación retenida cuando crea una suscripción, o cuando se vuelve a conectar a su sesión anterior.

Escriba aplicaciones de telemetría utilizando flujos de mensajes de IBM MQ o IBM Integration Bus.

Utilice JMS MQI u otras interfaces de programación de IBM MQ para programar aplicaciones de telemetría en IBM MQ.

El servicio de telemetría (MQXR) convierte entre mensajes MQTT v3 y mensajes IBM MQ. Crea suscripciones y publicaciones en nombre de los clientes MQTT y reenvía publicaciones a los clientes MQTT. Una publicación es la carga útil de un mensaje MQTT v3. La carga útil consta de las cabeceras del mensaje y una matriz de bytes en formato `jms-bytes`. El servidor de telemetría correlaciona las cabeceras entre un mensaje MQTTv3 y un mensaje IBM MQ; consulte [“Integración de MQ Telemetry con gestores de colas”](#) en la página 133.

Utilice la publicación, MQInput y los nodos Nodos de entrada de JMS para enviar y recibir publicaciones entre clientes de IBM Integration Bus y MQTT.

La utilización de flujos de mensajes puede integrar la telemetría con los sitios web utilizando HTTP y con otras aplicaciones utilizando los adaptadores IBM MQ y WebSphere.

## Integración de MQ Telemetry con gestores de colas

El cliente MQTT está integrado con IBM MQ como aplicación de publicación/suscripción. Puede publicar o suscribirse a temas en IBM MQ, crear nuevos temas o utilizar temas existentes. Recibe publicaciones de IBM MQ como resultado de los clientes MQTT, incluido él mismo u otras aplicaciones de IBM MQ que se publican en los temas de sus suscripciones. Las reglas se aplican para decidir los atributos de una publicación.

Muchos de los atributos asociados a temas, publicaciones, suscripciones y mensajes que proporciona IBM MQ no están soportados. En [“Cliente de MQTT al intermediario de publicación/suscripción de IBM MQ”](#) en la página 133 y en [“IBM MQ a un cliente MQTT”](#) en la página 135 se describe cómo se configuran los atributos de publicaciones. Los valores dependen de si la publicación va hacia o desde el intermediario de publicación/suscripción de IBM MQ.

En IBM MQ los temas de publicación/suscripción están asociados con objetos de tema administrativo. Los temas creados por clientes MQTT no son diferentes. Cuando un cliente MQTT crea una serie de tema para una publicación, el intermediario de publicación/suscripción de IBM MQ lo asocia a un objeto de tema administrativo. El intermediario correlaciona la serie de tema en la publicación con el padre de objeto de tema administrativo más cercano. La correlación es la misma que para las aplicaciones IBM MQ. Si no hay ningún tema que haya creado el usuario, la publicación de temas se correlaciona a SYSTEM.BASE.TOPIC. Los atributos que se aplican a la publicación se derivan del objeto de tema.

Cuando una aplicación de IBM MQ o un administrador crea una suscripción, la suscripción tiene un nombre. Enumere las suscripciones mediante IBM MQ Explorer, o mediante los comandos `runmqsc` o PCF. A todas las suscripciones de cliente MQTT se les asigna un nombre. Se les da un nombre de la forma: `ClientIdentifier:Topic name`

### Cliente de MQTT al intermediario de publicación/suscripción de IBM MQ

Un cliente MQTT ha enviado una publicación a IBM MQ. El servicio de telemetría (MQXR) convierte la publicación en un mensaje de IBM MQ. El mensaje de IBM MQ contiene tres partes:

1. MQMD
2. RFH2
3. Mensaje

Las propiedades de MQMD se establecen en sus valores predeterminados, excepto cuando se indique lo contrario en la [Tabla 9 en la página 134](#).

Campo MQMD	Tipo	Valor
<b>Format</b>	MQCHAR8	MQFMT_RF_HEADER_2
<b>UserIdentifier</b>	MQCHAR12	Establezca uno de los valores siguientes: MqttClient.ClientIdentifier MqttConnectOptions.UserName Un ID de usuario establecido por el administrador de IBM MQ para el canal de telemetría.
<b>Priority</b>	MLONG	MQPRI_PRIORITY_AS_Q_DEF (Valor predeterminado para IBM MQ, que es diferente de JMS que tiene un valor predeterminado de 4).
<b>Persistence</b>	MLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

La cabecera RFH2 no contiene una carpeta <msd> para definir el tipo del mensaje de JMS. El servicio de telemetría (MQXR) crea el mensaje de IBM MQ como un mensaje JMS predeterminado. El tipo de mensaje predeterminado de JMS es un mensaje `jms-bytes`. Una aplicación puede acceder a información de cabecera adicional como propiedades de mensaje; consulte [Propiedades del mensaje](#).

Los valores de RFH2 se establecen tal como se muestra en la [Tabla 10 en la página 134](#). El formato de la propiedad se establece en la cabecera fija RFH2, y los demás valores se establecen en las carpetas de RFH2.

Propiedad de RFH2	Tipo/carpeta	Cabecera
Formato	MQCHAR8	MQFMT_NONE
ClientIdentifier	mqtt/clientId	Copiar MqttClient.ClientIdentifier con una longitud de 1...23 bytes.
QoS	mqtt/qos	Copiar QoS del mensaje MQTT entrante.
ID de mensaje	mqtt/msgid	Copiar ID de mensaje del mensaje entrante MQTT, si QoS es 1 o 2.
MQIsRetained	mqs/Ret	Establecer si la publicación MQTT original se ha enviado con el conjunto de propiedades RETAIN y el mensaje se recibe como una publicación retenida.
MQTopicString	mqs/Top	El tema en el que se ha publicado el mensaje MQTT.

La carga útil de una publicación MQTT se correlaciona con el contenido de un mensaje de IBM MQ:

Tabla 11. Cómo la carga útil en una publicación de MQTT se correlaciona con el contenido del mensaje de IBM MQ

Contenido del mensaje	Tipo	Contenido del mensaje de IBM MQ
Buffer	MQBYTE <i>n</i>	Copia de bytes del mensaje MQTT entrante. La longitud puede ser cero.

## IBM MQ a un cliente MQTT

Un cliente se ha suscrito a un tema de publicación. Una aplicación de IBM MQ ha publicado en el tema, y en consecuencia, el intermediario de publicación/suscripción de MQTT envía una publicación al suscriptor de IBM MQ. De forma alternativa, una aplicación de IBM MQ ha enviado un mensaje no solicitado directamente a un cliente MQTT. En la Tabla 12 en la página 135 se describe cómo se establecen las cabeceras de mensajes fijos en el mensaje que se envía al cliente MQTT. Los demás datos de la cabecera de mensaje de IBM MQ o de cualquier otra cabecera, se eliminan. Los datos de mensaje del mensaje de IBM MQ se envían como carga útil del mensaje en el mensaje MQTT, sin ninguna alteración. El mensaje MQTT se envía al cliente MQTT a través del servicio de telemetría (MQXR).

Tabla 12. Cómo se establecen las cabeceras de mensajes fijas en un mensaje de IBM MQ enviado al cliente de MQTT

Campo de MQTT	Tipo	Valor
<b>DUP</b>	Boolean	Establézcalo si QoS = 1 o 2 y el mensaje se han enviado a este cliente en una transmisión anterior, y el mensaje no se ha reconocido transcurrido un tiempo.
<b>QoS</b>	entero	<p>La forma en que se establece el valor de QoS en una publicación saliente desde el intermediario de publicación/suscripción en IBM MQ depende de la publicación entrante. Depende de si la publicación entrante se ha enviado desde un cliente MQTT o desde una aplicación IBM MQ.</p> <p><b>MQTT</b> Valor inferior del QoS en la publicación entrante, y en el de QoS que ha solicitado el suscriptor.</p> <p><b>IBM MQ</b> Valor inferior del QoS derivado de la publicación entrante:                      MQPER_NOT_PERSISTENT→QoS=0                      MQPER_PERSISTENT→QoS=2</p> <p>y el QoS que ha solicitado el suscriptor. Si el mensaje se envía al cliente sin una suscripción, QoS se establece, de forma predeterminada, en 2. Un cliente puede modificar este valor suscribiéndose a DEFAULT . QoS con un QoS diferente.</p>
<b>RETAIN</b>	Boolean	Establecer si la publicación de entrada tiene establecida la propiedad retenida.

En la Tabla 13 en la página 136 se describe cómo las cabeceras de mensajes variables se establecen en el mensaje MQTT que se envía al cliente MQTT.

Tabla 13. Cómo se establecen las propiedades de cabecera variables de MQTT en un mensaje de MQTT que se envía al cliente de MQTT

Campo de MQTT	Tipo	Valor
Topic name	Serie	La serie de tema con la que se ha publicado el mensaje MQTT.
Message ID	Serie	Los 2 últimos bytes de la propiedad MQMD .MsgId de la publicación, cuando se coloca en la cola SYSTEM .MQTT . TRANSMIT . QUEUE.
Payload	byte []	Copia directa de los bytes procedentes de la publicación entrante al intermediario de publicación/suscripción. La longitud puede ser cero.

Windows

Linux

AIX

## Sesiones de MQTT con estado y sin estado

Los clientes MQTT pueden crear una sesión con estado con el gestor de colas. Cuando un cliente MQTT con estado se desconecta, el gestor de colas mantiene las suscripciones creadas por el cliente y en los mensajes en curso. Cuando el cliente se vuelve a conectar, resuelve los mensajes en curso. Envía los mensajes que están en la cola de entrega para realizar su entrega, y recibe los mensajes publicados para sus suscripciones mientras estaba desconectado.

Cuando un cliente MQTT se conecta a un canal de telemetría, inicia una nueva sesión o bien reanuda una sesión antigua. Una sesión nueva no tiene ningún mensaje pendiente sin acuse de recibo, ninguna suscripción, ni publicación a la espera de ser entregada. Cuando un cliente se conecta, especifica si se empieza con una sesión limpia o si se reanuda una sesión existente; consulte [Limpiar sesiones](#).

Si el cliente reanuda una sesión existente, continúa como si la conexión no se hubiera interrumpido. Las publicaciones que están esperando a ser entregadas se envían al cliente y las transferencias de mensajes no confirmadas se completan. Cuando un cliente de una sesión persistente se desconecta del servicio de telemetría (MQXR), todas las suscripciones que el cliente ha creado permanecen. Las publicaciones de las suscripciones se envían al cliente cuando éste vuelve a conectarse. Si se vuelve a conectar sin reanudar la sesión anterior, el servicio de telemetría (MQXR) descarta las publicaciones.

El gestor de colas guarda la información del estado de la sesión en la cola SYSTEM .MQTT . PERSISTENT . STATE.

El administrador de IBM MQ puede desconectar y depurar una sesión.

Windows

Linux

AIX

## Cuando un cliente MQTT no está conectado

Cuando un cliente no se conecta al gestor de colas, puede continuar recibiendo publicaciones en su nombre. Se reenvían al cliente cuando éste se vuelve a conectar. Un cliente puede crear una "Última voluntad y testamento" para que el gestor de colas realice publicaciones en su nombre cuando éste se desconecte de forma inesperada.

Si desea recibir notificaciones cuando el cliente se desconecte de forma inesperada, puede registrar una publicación de "Última voluntad y testamento"; consulte la publicación [Última voluntad y testamento](#). Se envía a través del servicio de telemetría (MQXR), si éste detecta que la conexión con el cliente se ha interrumpido sin que este último lo haya solicitado.

Un cliente puede publicar una publicación retenida en cualquier momento; consulte [Publicaciones retenidas y clientes MQTT](#). Puede solicitarse que una suscripción nueva a un tema se envíe con cualquier publicación retenida asociada al mismo. Si crea la última voluntad y testamento como una publicación retenida, puede utilizarla para supervisar el estado de un cliente.

Por ejemplo, el cliente publica una publicación retenida, cuando se conecta, informando de su disponibilidad. Al mismo tiempo, crea una publicación retenida de última voluntad y testamento que anuncia que no está disponible. Además, justo antes de realizar una desconexión planificada, publica

su no disponibilidad como una publicación retenida. Para averiguar si el cliente está disponible, debe suscribirse al tema de la publicación retenida. Siempre recibirá una de las tres publicaciones.

Si el cliente va a recibir mensajes publicados cuando se desconecta, vuelva a conectarlo a su sesión anterior; consulte [“Sesiones de MQTT con estado y sin estado”](#) en la página 136. Sus suscripciones quedan activas hasta que se supriman o hasta que el cliente cree una sesión limpia.

## Windows Linux AIX Acoplamiento dinámico entre clientes MQTT y aplicaciones de IBM MQ

El flujo de publicaciones entre los clientes MQTT y las aplicaciones IBM MQ se acopla de forma dinámica. Las publicaciones pueden originarse en un cliente MQTT o en una aplicación de IBM MQ y sin ningún orden establecido. Los publicadores y los suscriptores son de acoplamiento dinámico. Interactúan unos con otros de forma indirecta a través de publicaciones y suscripciones. También puede enviar mensajes directamente a un cliente MQTT desde una aplicación de IBM MQ.

Los clientes MQTT y las aplicaciones de IBM MQ se pueden acoplar dinámicamente en dos sentidos:

1. Los publicadores y suscriptores tienen un acoplamiento dinámico por la asociación de una publicación y una suscripción con un tema. Los publicadores y suscriptores normalmente no tienen en cuenta ni la dirección ni la identidad del otro origen de una publicación o suscripción.
2. Los clientes MQTT publican, suscriben, reciben publicaciones y procesan reconocimientos de entrega en hebras separadas.

Una aplicación de cliente MQTT no espera hasta que se haya entregado una publicación. La aplicación transfiere un mensaje al cliente MQTT y a continuación, la aplicación continúa en su propia hebra. Para sincronizar la aplicación con la entrega de una publicación, se utiliza una señal de entrega; consulte [Entrega de señales](#).

Después de pasar un mensaje al cliente MQTT, la aplicación tiene la opción de esperar en la señal de entrega. En lugar de esperar, el cliente puede proporcionar un método de devolución de llamada que se invoca cuando la publicación se entrega a IBM MQ. También Puede ignorar la señal de entrega.

En función de la calidad de servicio asociada al mensaje, se devuelve la señal de entrega de forma inmediata al método de devolución de llamada, o puede hacerse una vez transcurrido cierto tiempo. La señal de entrega puede devolverse incluso después de que el cliente se desconecte y vuelva a conectarse. Si la calidad de servicio es *enviar y olvidar*, la señal de entrega se devuelve inmediatamente. En los otros dos casos, la señal de entrega sólo se devuelve cuando el cliente recibe el acuse de recibo indicando que la publicación se ha enviado a los suscriptores.

Las publicaciones que se envían a un cliente MQTT como resultado de una suscripción cliente se entregan al método de devolución de llamada `messageArrived`. `messageArrived` se ejecuta en una hebra diferente a la de la aplicación principal.

### Envío de mensajes directamente a un cliente MQTT

Puede enviar un mensaje a un cliente MQTT determinado de una de estas dos formas.

1. Una aplicación de IBM MQ puede enviar un mensaje directamente a un cliente MQTT sin una suscripción; consulte [Envío de un mensaje a un cliente directamente](#).
2. Un enfoque alternativo es utilizar el convenio de denominación `ClientIdentifier`. Haga que todos los suscriptores MQTT creen suscripciones utilizando como tema el `ClientIdentifier` exclusivo. Publique en `Identificador_cliente`. La publicación se envía al cliente que esté suscrito al tema `Identificador_cliente`. Mediante esta técnica puede enviar una publicación a un suscriptor MQTT determinado.

## Windows Linux AIX Seguridad de MQ Telemetry

La seguridad de los dispositivos de telemetría puede ser importante, ya que suelen ser portátiles y suelen utilizarse en lugares que no pueden controlarse de forma precisa. Puede utilizar VPN para proteger

la conexión del dispositivo MQTT en el servicio de telemetría (MQXR). MQ Telemetry proporciona dos mecanismos de seguridad adicionales, TLS y JAAS.

TLS se utiliza principalmente para cifrar las comunicaciones entre el dispositivo y el canal de telemetría y para autenticar que el dispositivo se está conectando al servidor correcto; consulte [Autenticación del canal de telemetría utilizando TLS](#). También puede utilizar TLS para comprobar si el dispositivo cliente tiene permiso para conectarse al servidor; consulte [Autenticación de cliente MQTT utilizando TLS](#).

JAAS se utiliza principalmente para comprobar si el usuario del dispositivo tiene permiso para utilizar una aplicación de servidor; consulte [Autenticación de cliente MQTT utilizando una contraseña](#). JAAS puede utilizarse con LDAP para comprobar una contraseña que utiliza un directorio de inicio de sesión único.

TLS y JAAS pueden utilizarse conjuntamente para proporcionar dos factores de autenticación. Puede restringir los cifrados que utiliza TLS a cifrados que cumplan los estándares FIPS.

Con decenas de miles de usuarios, no siempre resulta práctico proporcionar perfiles individuales de seguridad. Tampoco resulta práctico utilizar los perfiles para autorizar a usuarios individuales a acceder a objetos de IBM MQ. En su lugar, es mejor agrupar los usuarios en clases para la autorización de publicaciones y suscripciones a temas y enviar publicaciones a los clientes.

Configure cada canal de telemetría para que se correlacione a clientes con ID de usuario de cliente comunes. Utilice un ID de usuario común para cada cliente que se conecta en un canal específico; consulte [Identidad y autorización de cliente MQTT](#).

La autorización de grupos de usuarios no compromete la autenticación de cada individuo. Cada usuario puede autenticarse, en el servidor o en el cliente, con su valor de Username y de Password, y luego autorizarse en el servidor mediante un ID de usuario común.

Windows

Linux

AIX

## Globalización de MQ Telemetry

La carga útil del mensaje en el protocolo MQTT v3 se codifica como una matriz de bytes. Por lo general, las aplicaciones que manejan texto crean la carga útil del mensaje en formato UTF-8. El canal de telemetría describe la carga útil del mensaje como UTF-8, pero no realiza ninguna conversión de páginas de códigos. La serie del tema de publicación debe estar en UTF-8.

La aplicación se encarga de convertir los datos alfanuméricos a la página de códigos correcta y los datos numéricos a la codificación de números correcta.

El cliente MQTT Java tiene un cómodo método `MqttMessage.toString`. El método trata la carga útil del mensaje como si estuviera codificada en el conjunto de caracteres predeterminado de la plataforma que, por lo general, es UTF-8. Convierte la carga útil en una String de Java. Java cuenta con un método `String.getBytes` que convierte una serie en una matriz de bytes codificada utilizando el conjunto de caracteres predeterminado de la plataforma local. Dos programas MQTT Java que intercambien texto en la carga útil del mensaje entre plataformas con el mismo conjunto de caracteres predeterminado lo hacen de una forma fácil y eficaz en UTF-8.

Si el conjunto de caracteres predeterminado de una de las plataformas no es UTF-8, las aplicaciones deben establecer un convenio para el intercambio de mensajes. Por ejemplo, el publicador especifica la conversión de una serie a UTF-8 mediante el método `getBytes("UTF8")`. Para recibir el texto de un mensaje, el suscriptor presupone que el mensaje se ha codificado en el juego de caracteres UTF-8.

El servicio de telemetría (MQXR) describe la codificación de todas las publicaciones entrantes de mensajes de clientes MQTT como si fuera UTF-8. Establece `MQMD.CodedCharSetId` a UTF-8, y `RFH2.CodedCharSetId` a `MQCCSI_INHERIT`; consulte [“Integración de MQ Telemetry con gestores de colas”](#) en la página 133. El formato de la publicación se establece en `MQFMT_NONE`, por lo que ni los canales, ni `MQGET`, pueden realizar la conversión.

Windows

Linux

AIX

## Rendimiento y escalabilidad de MQ Telemetry

Tenga en cuenta los siguientes factores al gestionar un gran número de clientes y mejorar la escalabilidad de MQ Telemetry.

## Planificación de la capacidad

Para obtener información sobre los informes de rendimiento de MQ Telemetry, consulte los documentos de [MQ Performance](#).

### Conexiones

Los costes que conllevan las conexiones incluyen:

- El gasto de configuración de la misma conexión en términos de uso del procesador y de tiempo.
- Costes de la red.
- Memoria usada cuando se mantiene una conexión abierta que no se utilizada.

Existe una carga adicional que se produce cuando los clientes permanecen conectados. Si una conexión se mantiene abierta, los flujos de TCP/IP y los mensajes de MQTT utilizan la red para comprobar que la conexión aún está activa. Además, se utiliza memoria del servidor cada vez que la conexión de cliente se mantiene abierta.

Si se envía más de un mensaje por minuto, debe mantener la conexión abierta para evitar el coste que supone iniciar una nueva conexión. Si envía menos de un mensaje cada 10 ó 15 minutos, considere la posibilidad de interrumpir la conexión para evitar el gasto que supone mantenerla abierta. Es posible que desee mantener una conexión TLS abierta, pero desocupada, durante períodos de tiempo más largos, porque es más cara de configurar.

Tenga en cuenta, además, la capacidad del cliente. Si el cliente cuenta con una recurso de almacén y reenvío, puede configurar los mensajes por lotes, e interrumpir la conexión entre los envíos de los procesos por lotes. Sin embargo, si el cliente se desconecta, el cliente no podrá recibir ningún mensaje del servidor. Por lo tanto, el objetivo de la aplicación está relacionado con la decisión.

Si el sistema cuenta con un cliente que envía muchos mensajes, por ejemplo, transfiere archivos, es mejor no esperar la respuesta del servidor por mensaje. En su lugar, envíe todos los mensajes y compruebe al final que se han recibido. De forma alternativa, utilice [Calidad de servicio \(QoS\)](#).

Puede variar la QoS según el mensaje, suministrando mensajes sin importancia con QoS 0 y mensajes importantes utilizando una QoS de 2. La capacidad de proceso de mensajes puede ser aproximadamente el doble con una QoS de 0 que con una QoS de 2.

### Convenios de denominación

Si diseña la aplicación para muchos clientes, implemente un convenio de denominación eficaz. Para correlacionar cada cliente al `ClientIdentifier` correcto, haga que el `ClientIdentifier` sea lo más significativo posible. Un buen convenio de denominación facilita al administrador el poder averiguar qué clientes se están ejecutando. Un convenio de denominación ayuda al administrador a filtrar una larga lista de clientes en IBM MQ Explorer y ayuda en la determinación de problemas; consulte [Identificador de cliente](#).

### Rendimiento

La longitud de los nombres de tema afecta al número de bytes que se transmiten por la red. Cuando se publica o se realiza una suscripción, el número de bytes de un mensaje puede ser importante. Por tanto, es recomendable que limite el número de caracteres del nombre de un tema. Cuando un cliente MQTT se suscribe a un tema IBM MQ le da un nombre del formulario:

```
ClientIdentifier: TopicName
```

Para ver todas las suscripciones para un cliente MQTT, puede utilizar el mandato IBM MQ MQSC **DISPLAY:**

```
DISPLAY SUB(' ClientID1:*')
```

## Definición de recursos en IBM MQ para que los utilicen los clientes MQTT

Un cliente MQTT se conecta a un gestor de colas remoto de IBM MQ. Hay dos métodos básicos para que una aplicación de IBM MQ envíe mensajes a un cliente MQTT establecer la cola de transmisión predeterminada en `SYSTEM.MQTT.TRANSMIT.QUEUE` o utilizar alias de gestor de colas. Defina la cola de transmisión predeterminada de un gestor de colas; si existe un gran número de clientes MQTT. Si se utiliza el valor de cola de transmisión predeterminado se simplifica el esfuerzo de administración, consulte [Configurar gestión de colas distribuidas para enviar mensajes a clientes MQTT](#).

## Mejora de la escalabilidad evitando suscripciones.

Cuando un cliente de MQTT V3 se suscribe a un tema, el servicio de telemetría (MQXR) crea una suscripción en IBM MQ. La suscripción dirige las publicaciones del cliente a `SYSTEM.MQTT.TRANSMIT.QUEUE`. El nombre de gestor de colas remoto que aparece en la cabecera de transmisión de cada publicación se establece en el `ClientIdentifier` del cliente MQTT que realizó la suscripción. Si hay muchos clientes, cada uno de ellos realizando sus propias suscripciones, esto da como resultado que se mantengan muchas suscripciones de proxy en toda la jerarquía o clúster de publicación/suscripción de IBM MQ. Para obtener información sobre cómo no utilizar la publicación/suscripción pero, cómo utilizar una solución de tipo punto a punto, consulte [Envío de un mensaje a un cliente directamente](#).

## Gestión de grandes cantidades de clientes

Para dar soporte a muchos clientes conectados al mismo tiempo, aumente la memoria disponible para el servicio de telemetría (MQXR) definiendo los parámetros de JVM `-Xms` y `-Xmx`. Siga estos pasos:

1. Busque el archivo `java.properties` en el directorio de configuración del servicio de telemetría; consulte el directorio de configuración del servicio de telemetría (MQXR) en [Windows](#) o el [directorio de configuración del servicio de telemetría en Linux](#).
2. Siga las instrucciones del archivo; un almacenamiento dinámico de 1 GB es suficiente para que haya 50.000 clientes conectados simultáneamente.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Añada otros argumentos de línea de mandatos para pasar a la JVM que ejecuta el servicio de telemetría (MQXR) en el archivo `java.properties`; consulte [Cómo pasar parámetros de JVM al servicio de telemetría \(MQXR\)](#).

Para aumentar el número de descriptores de archivo abiertos en Linux, añada las líneas siguientes a `/etc/security/limits.conf/` y vuelva a iniciar sesión.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Cada socket necesita un descriptor de archivo. El servicio de telemetría necesita algunos descriptores de archivo adicionales, por lo que este número debe ser mayor que el número de sockets abiertos necesarios.

El gestor de colas utiliza un manejador de objeto para cada suscripción no duradera. Para dar soporte a muchas suscripciones no duraderas activas, aumente el número máximo de manejadores activos en el gestor de colas; por ejemplo:

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

Figura 48. Modificar el número máximo de manejadores en Windows

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

Figura 49. Modificar el número máximo de manejadores en Linux

## Otras consideraciones

Cuando planifique los requisitos del sistema, tenga en cuenta el tiempo que necesitará para reiniciar el sistema. El tiempo de inactividad planificado puede tener implicaciones para el número de mensajes que se acumulen en la cola, a la espera de ser procesados. Configure el sistema de modo que los mensajes se puedan procesar de forma satisfactoria en un tiempo aceptable. Revise el almacenamiento de disco, la memoria y la potencia de proceso. Con algunas aplicaciones cliente, es posible descartar mensajes cuando el cliente se vuelve a conectar. Para descartar mensajes, establezca `CleanSession` en los parámetros de conexión del cliente; consulte [Borrado de sesiones](#). O bien, publique y suscriba utilizando el nivel máximo de calidad de servicio, 0, en un cliente MQTT; consulte [Calidad de servicio](#). Utilice mensajes no persistentes cuando envíe mensajes desde IBM MQ. Los mensajes que tienen estas calidades de servicio no se recuperan cuando se reinicia el sistema o la conexión.

Windows

Linux

AIX

## Dispositivos soportados por MQ Telemetry

Los clientes MQTT se pueden ejecutar en una gama de dispositivos, desde sensores y actuadores hasta dispositivos portátiles y sistemas para vehículos.

Los clientes MQTT son pequeños y se ejecutan en dispositivos con poca memoria y una potencia de proceso restringida. El MQTT protocol es fiable y tiene cabeceras pequeñas, que son ideales para redes limitadas por un ancho de banda reducido, un alto coste y una disponibilidad intermitente.

MQ Telemetry se comunica con dispositivos de telemetría a través de aplicaciones cliente MQTT. Estas aplicaciones utilizan los siguientes recursos, que implementan todos ellos el protocolo MQTT v3:

- Las siguientes bibliotecas cliente:
  - *MQTT client for Java*, que se utiliza para compilar aplicaciones nativas para (por ejemplo), dispositivos Android, OS X, Linux o Windows. Las aplicaciones que utilizan esta biblioteca de cliente se pueden ejecutar en todas las variaciones de Java desde CLDC (Connected Limited Device Configuration)/MIDP (Mobile Information Device Profile), a través de CDC (Connected Device Configuration)/Foundation, J2SE (Java Platform, Standard Edition) y J2EE (Java Platform, Enterprise Edition). La biblioteca de clases personalizadas `jclRM` de IBM también está soportada. La plataforma Java ME se suele utilizar en pequeños dispositivos, como por ejemplo, actuadores, sensores, teléfonos móviles y otros dispositivos integrados. La plataforma Java SE se instala por lo general en dispositivos de gama superior e integrados, como por ejemplo, sistemas y servidores.
  - *MQTT client for C*, que se utiliza para crear aplicaciones nativas para (por ejemplo) dispositivos iOS, OS X, Linux o Windows. Esta biblioteca de cliente proporciona una implementación de referencia C junto con el cliente nativo precompilado para sistemas Windows y Linux. La implementación de referencia C permite la portabilidad de MQTT a una amplia gama de dispositivos y plataformas. Algunos sistemas Windows en Intel, incluidos Windows 7, RedHat, Ubuntu y algunos sistemas Linux en plataformas ARM como Eurotech Viper, implementan versiones de Linux que ejecutan el cliente C, pero IBM no proporciona soporte de servicio para las plataformas. Deberá reducir los problemas con el cliente en una plataforma soportada si tiene previsto llamar al centro de soporte de IBM.
  - *MQTT client for Java*, que se utiliza para compilar aplicaciones web basadas en navegador.

MQTTLas bibliotecas cliente de están disponibles gratuitamente en Eclipse Paho y MQTT.org. Consulte [Programas de ejemplo de IBM MQ Telemetry Transport](#).

## Seguridad en IBM MQ

En IBM MQ, existen varios métodos para proporcionar seguridad: la interfaz del servicio de autorización; rutinas de salida de canal escritas por el usuario o por terceros; seguridad de canal utilizando la seguridad de la capa de transporte (TLS), registros de autenticación de canal y seguridad de mensajes.

### Interfaz del servicio de autorización

La autorización para utilizar llamadas MQI, mandatos y acceso a objetos la proporciona el **gestor de autorizaciones sobre objetos** (OAM), que está habilitado de forma predeterminada. El acceso a las entidades de IBM MQ se controla a través de grupos de usuarios de IBM MQ y el OAM.

Los administradores pueden utilizar una interfaz de línea de mandatos para otorgar o revocar las autorizaciones según sea necesario.

Para obtener más información sobre la creación de componentes del servicio de autorización, consulte [Configuración de la seguridad en sistemas AIX, Linux, and Windows](#).

## Salidas de canal escritas por el usuario o de otros fabricantes

Los canales pueden utilizar salidas de canal escritas por el usuario o de otros fabricantes. Para obtener más información, consulte [Programas de salida de canal para canales de mensajería](#).

## Seguridad de canal mediante TLS

El protocolo TLS (Transport Layer Security) proporciona la seguridad de canal estándar de la industria, con protección contra escuchas y manipulaciones no autorizadas y contra falsas identidades.

TLS utiliza técnicas de clave pública y simétricas para proporcionar confidencialidad e integridad así como autenticación mutua.

Para obtener una revisión completa de la seguridad en IBM MQ, incluida información detallada sobre TLS, consulte [Seguridad](#). Para obtener una visión general de TLS, incluidos los punteros a los mandatos que se describen en este apartado, consulte [Protocolos de seguridad de cifrado: TLS](#).

## Registros de autenticación de canal

Utilice registros de autenticación de canal para ejercer un control preciso sobre el acceso otorgado a conectar sistemas en un nivel de canal. Para obtener más información, consulte [Registros de autenticación de canal](#).

## Seguridad de mensajes

Utilice Advanced Message Security, que es un componente con licencia y instalado por separado de IBM MQ, para proporcionar protección criptográfica a los mensajes enviados y recibidos utilizando IBM MQ. Consulte [Advanced Message Security](#).

### Tareas relacionadas

[Seguridad](#)

[Planificación de los requisitos de seguridad](#)

## Soporte TLS de cliente gestionado de IBM MQ.NET

El cliente IBM MQ.NET completamente gestionado proporciona soporte de seguridad de la capa de transporte (TLS) que se basa en el kit Microsoft.NET SSLStreams. Esto es diferente de los otros clientes IBM MQ , que se basan en IBM Global Security Kit (GSKit).

Puede desarrollar aplicaciones IBM MQ.NET para que se ejecuten en modalidad gestionada o en modalidad no gestionada.

- En modalidad gestionada, las aplicaciones .NET funcionan en CLR (Common Language Runtime) .NET sin ninguna invocación entre plataformas al invocar la MQI C.
- En modalidad no gestionada, C MQI se invoca para las operaciones de MQI subyacentes. Básicamente, la interfaz en modalidad no gestionada consta de las clases de derivador .NET en la parte superior de C MQI.

El cliente de IBM MQ.NET gestionado utiliza las bibliotecas de Microsoft.NET Framework para implementar protocolos de sockets seguros TLS. El sistema.NET.Security.La clase SSLStream de Microsoft se utiliza para implementar la seguridad (TLS) en IBM MQ.NET.

La modalidad de cliente de IBM MQ.NET no gestionada ya da soporte a la característica TLS, que se basa en C MQI (y GSKit). Es decir, C MQI maneja las operaciones TLS. En este caso, GSKit implementa los protocolos de socket seguro TLS.

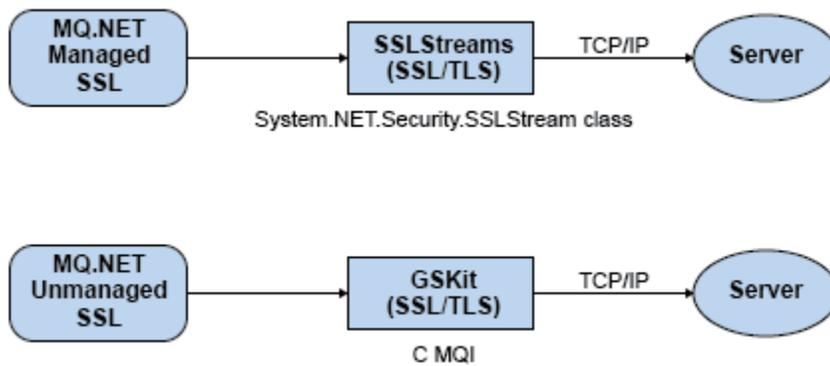


Figura 50. Comparación de TLS gestionada y no gestionada de IBM MQ.NET

En la tabla siguiente se resumen las diferencias entre las implementaciones gestionadas y no gestionadas:

Tabla 14. Diferencias entre implementaciones gestionadas y no gestionadas

Modalidad	Protocolos	Implementación	Comentarios
SSL gestionado de IBM MQ.NET	TLS	Clase System.NET.Security.SSLStream  La clase SSLStream funciona como una corriente a través de un socket TCP conectado.	TLS 1.0  TLS 1.2 (solo con Microsoft.NET Framework v4.5)
SSL no gestionado de IBM MQ.NET	TLS	GSKIT y C-MQI	Protocolos de sockets seguros TLS

### Conceptos relacionados

Soporte de SSL (Secure Sockets Layer) y TLS (Transport Layer Security) para .NET

## IBM MQ MQI clients

Un IBM MQ MQI client es un componente del producto IBM MQ que puede instalarse en un sistema en el que no se ejecuta ningún gestor de colas.

Un IBM MQMQI *cliente* es un componente que permite que una aplicación que se ejecuta en un sistema emita llamadas MQI a un gestor de colas que se ejecuta en otro sistema. La salida de la llamada se devuelve al cliente, que la devuelve a su vez a la aplicación.

Utilizando un IBM MQ MQI client, una aplicación que se ejecuta en el mismo sistema que el cliente puede conectarse a un gestor de colas que se ejecuta en otro sistema. La aplicación puede emitir llamadas MQI a ese gestor de colas. Dicha aplicación se denomina aplicación IBM MQ MQI client y el gestor de colas se denomina *gestor de colas servidor*.

Un IBM MQ *servidor* es un gestor de colas que proporciona servicios de gestión de colas a uno o más clientes. Todos los objetos de IBM MQ, por ejemplo colas, solamente existen en la máquina del gestor de colas (la máquina del servidor de IBM MQ) y no en el cliente. Un servidor de IBM MQ también puede dar soporte a aplicaciones de IBM MQ locales.

La diferencia entre un servidor de IBM MQ y un gestor de colas ordinario es que un servidor tiene un enlace de comunicaciones dedicado con cada cliente. Puede obtener información adicional sobre la creación de canales para clientes y servidores consultando [Configuración de colas distribuidas](#).

Una aplicación IBM MQ MQI client y un gestor de colas de servidor se comunican entre sí utilizando un *canal MQI*. Un canal MQI se inicia cuando la aplicación de cliente emite una llamada **MQCONN** o **MQCONNX** para conectarse al gestor de colas y finaliza cuando la aplicación de cliente emite una llamada **MQDISC** para desconectarse del gestor de colas. Los parámetros de entrada de una llamada MQI fluyen en una dirección de un canal MQI y los parámetros de salida fluyen en la dirección opuesta.

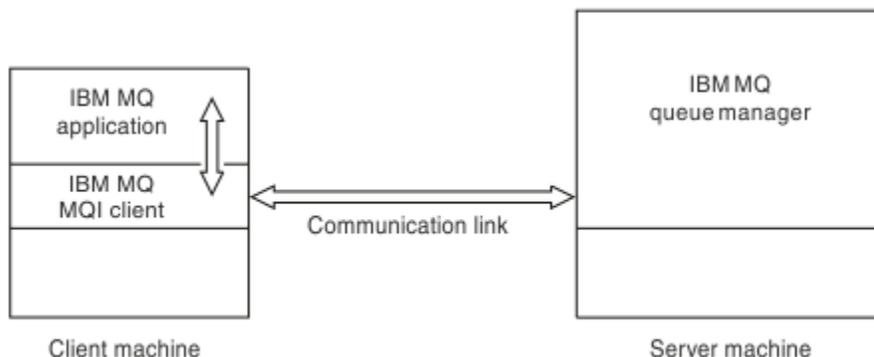


Figura 51. Enlace entre un cliente y un servidor

Se pueden utilizar las plataformas siguientes. Las combinaciones dependen del producto IBM MQ que esté utilizando y se describen en [“Soporte de plataforma para clientes de IBM MQ”](#) en la página 145.

#### IBM MQ MQI client

AIX and Linux  
Windows  
IBM i

#### Servidor de IBM MQ

AIX and Linux  
Windows  
IBM i  
z/OS

MQI está disponible para las aplicaciones que se ejecutan en la plataforma cliente; las colas y otros objetos de IBM MQ se mantienen en un gestor de colas que ha instalado en un servidor.

Una aplicación que se desea ejecutar en el entorno IBM MQ MQI client debe enlazarse primero con la biblioteca de cliente relevante. Cuando la aplicación emite una llamada MQI, el IBM MQ MQI client dirige la solicitud a un gestor de colas, donde se procesa y desde donde se envía una respuesta al IBM MQ MQI client.

El enlace entre la aplicación y el IBM MQ MQI client se establece dinámicamente en tiempo de ejecución.

También puede desarrollar aplicaciones cliente utilizando las IBM MQ classes for .NET, IBM MQ classes for Java o IBM MQ classes for Java Message Service (JMS). Puede utilizar clientes Java y JMS en las plataformas siguientes:

-  IBM i
-  AIX
-  Linux
-  Windows

El uso de Java y JMS no se describe aquí. Para obtener detalles completos acerca de la instalación, configuración y uso de IBM MQ classes for Java y IBM MQ classes for JMS, consulte [Utilización de IBM MQ classes for Java](#) y [Utilización de IBM MQ classes for JMS](#).

## Aplicaciones IBM MQ en un entorno cliente-servidor

Cuando se enlazan con un servidor, las aplicaciones cliente de IBM MQ pueden emitir la mayoría de llamadas MQI de la misma forma que las aplicaciones locales. La aplicación cliente emite una llamada MQCONN para conectarse a un gestor de colas especificado. Todas las llamadas MQI adicionales que especifiquen el manejador de conexión devuelto desde la petición de conexión serán procesadas entonces por este gestor de colas.

Debe enlazar sus aplicaciones con las bibliotecas de cliente adecuadas. Consulte [Creación de aplicaciones para IBM MQ MQI clients](#).

### Conceptos relacionados

[“¿Por qué utilizar clientes IBM MQ” en la página 145](#)

La utilización de clientes IBM MQ es una forma eficaz de implementar la mensajería y la transferencia a colas de IBM MQ.

[“¿Qué es un cliente transaccional extendido?” en la página 147](#)

Un cliente transaccional extendido de IBM MQ puede actualizar recursos gestionados por otro gestor de recursos, bajo el control de un gestor de transacciones externo.

[“Cómo se conecta el cliente al servidor” en la página 148](#)

Un cliente se conecta a un servidor utilizando MQCONN o MQCONNX y se comunica a través de un canal.

[“Gestión y soporte de transacciones” en la página 149](#)

Introducción a la gestión de transacciones y cómo IBM MQ da soporte a las transacciones.

[“Extensión de recursos del gestor de colas” en la página 151](#)

Puede ampliar los recursos del gestor de colas utilizando las salidas de usuario, las salidas de API, o los servicios instalables.

### Información relacionada

[Cómo configurar un IBM MQ MQI client](#)

## ¿Por qué utilizar clientes IBM MQ

La utilización de clientes IBM MQ es una forma eficaz de implementar la mensajería y la transferencia a colas de IBM MQ.

Puede tener una aplicación que utilice la interfaz MQI en una máquina y el gestor de colas en ejecución en una máquina diferente (física o virtual). Las ventajas son las siguientes:

- No es necesario realizar una implementación completa de IBM MQ en la máquina cliente.
- Los requisitos de hardware en el sistema cliente se reducen.
- Los requisitos de administración del sistema se reducen.
- Una aplicación de IBM MQ que se ejecuta en un cliente puede conectarse a varios gestores de colas en distintos sistemas.
- Se pueden utilizar canales alternativos que utilicen protocolos de transmisión diferentes.

## Soporte de plataforma para clientes de IBM MQ

IBM MQ en todas las plataformas de servidor soportadas acepta conexiones de cliente de IBM MQ MQI clients en un número de plataformas.

IBM MQ instalado como *Producto base y servidor* en todas las plataformas de servidor soportadas puede aceptar conexiones de IBM MQ MQI clients en las siguientes plataformas:

-  IBM i
-  AIX
-  Linux
-  Windows

Las conexiones de cliente están sujetas a diferencias en el identificador de juego de caracteres codificados (CCSID) y el protocolo de comunicaciones.

## ¿Qué aplicaciones se ejecutan en un IBM MQ MQI client?

En el entorno cliente, la interfaz MQI completa está soportada. Esto permite que casi cualquier aplicación de IBM MQ se configure para ejecutarse en un sistema IBM MQ MQI client enlazando la aplicación en el IBM MQ MQI client a la biblioteca MQIC, en lugar de hacerlo a la biblioteca MQI. Existen las siguientes excepciones:

- MQGET con señal
- Una aplicación que necesita la coordinación de punto de sincronismo con otros gestores de recursos utilizan un cliente transaccional extendido.

Si se habilita la lectura anticipada, para mejorar el rendimiento de la mensajería no persistente no todas las opciones de MQGET están disponibles. La tabla muestra las opciones que están permitidas y si se pueden alterar entre llamadas MQGET.

Tabla 15. Opciones MQGET permitidas cuando la lectura anticipada está habilitada			
Valores	Permitidas cuando la lectura anticipada está habilitada y se pueden modificar entre llamadas MQGET	Permitidas cuando la lectura anticipada está habilitada, pero no se pueden modificar entre llamadas MQGET <sup>1</sup>	Opciones MQGET que no están permitidas cuando la lectura anticipada está habilitada <sup>2</sup>
Valores MD de MQGET	MsgId <sup>3</sup> CorrelId <sup>3</sup>	Encoding CodedCharSetId	
Opciones MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST <sup>4</sup> MQGMO_BROWSE_NEXT <sup>4</sup> MQGMO_BROWSE_MESSAGE_UNDER_CURSOR <sup>4</sup>	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR <sup>4</sup> MQGMO_LOCK MQGMO_UNLOCK
Valores MQGMO		MsgHandle	

1. Si estas opciones se modifican entre llamadas MQGET se devuelve el código de razón MQRC\_OPTIONS\_CHANGED.
2. Si estas opciones se especifican en la primera llamada MQGET, la lectura anticipada está inhabilitada. Si estas opciones se especifican en una llamada MQGET posterior, se devuelve el código de razón MQRC\_OPTIONS\_ERROR.
3. Las aplicaciones cliente han de tener presente que si los valores MsgId y CorrelId se modifican entre llamadas MQGET, es posible que los mensajes con los valores anteriores ya se hayan enviado al cliente y permanezcan en el almacenamiento intermedio de lectura anticipada del cliente hasta que se consuman (o se depuren automáticamente).

4. La primera llamada MQGET determina si los mensajes se han de examinar u obtener de una cola cuando la lectura anticipada está habilitada. Si la aplicación intenta utilizar una combinación de opciones de examen y de obtención, se devuelve el código de razón MQRC\_OPTIONS\_CHANGED.
5. MQGMO\_MSG\_UNDER\_CURSOR no es posible con la lectura anticipada. Los mensajes se pueden examinar u obtener cuando la lectura anticipada está habilitada, pero no se puede hacer ambas cosas a la vez.

Una aplicación que se ejecuta en un IBM MQ MQI client se puede conectarse a más de un gestor de colas simultáneamente, o utilizar un nombre de gestor de colas con un asterisco (\*) en una llamada MQCONN o MQCONNX (consulte los ejemplos en [Conexión de aplicaciones IBM MQ MQI client a gestores de colas](#)).

## ¿Qué es un cliente transaccional extendido?

Un cliente transaccional extendido de IBM MQ puede actualizar recursos gestionados por otro gestor de recursos, bajo el control de un gestor de transacciones externo.

Si no está familiarizado con los conceptos de gestión de transacciones, consulte [“Gestión y soporte de transacciones”](#) en la página 149.

Tenga en cuenta que el cliente transaccional XA ahora se suministra como parte de IBM MQ.

Una aplicación de cliente puede participar en una unidad de trabajo que está gestionada por un gestor de colas al que está conectado. Dentro de la unidad de trabajo, la aplicación de cliente puede transferir mensajes a las colas y obtener mensajes de las colas que son propiedad de ese gestor de colas. La aplicación de cliente puede entonces utilizar la llamada **MQCMIT** para confirmar la unidad de trabajo o la llamada **MQBACK** para restituir la unidad de trabajo. Sin embargo, dentro de la misma unidad de trabajo, la aplicación de cliente no puede actualizar los recursos de otro gestor de recursos, las tablas de una base de datos de Db2, por ejemplo. La utilización de un cliente transaccional extendido de IBM MQ elimina esta restricción.

Un cliente transaccional extendido de IBM MQ es un IBM MQ MQI client con alguna función adicional. Mediante esta función, una aplicación de cliente, dentro de la misma unidad de trabajo puede realizar las tareas siguientes:

- Transferir mensajes a las colas y obtener mensajes de las colas que son propiedad del gestor de colas al que está conectado
- Actualizar los recursos de un gestor de recursos que no sea un gestor de colas de IBM MQ

Esta unidad de trabajo la debe gestionar un gestor de transacciones externo que se ejecuta en el mismo sistema que la aplicación de cliente. La unidad de trabajo no la puede gestionar el gestor de colas al que está conectada la aplicación de cliente. Esto significa que el gestor de colas sólo puede actuar como un gestor de recursos y no como un gestor de transacciones. También indica que la aplicación de cliente puede confirmar o restituir la unidad de trabajo utilizando sólo la interfaz de programación de aplicaciones (API) que proporciona el gestor de transacciones externo. La aplicación de cliente no puede por lo tanto utilizar las llamadas de MQI, **MQBEGIN**, **MQCMIT** y **MQBACK**.

El gestor de transacciones externo se comunica con el gestor de colas como un gestor de recursos utilizando el mismo canal MQI que el que utiliza la aplicación de cliente que está conectada al gestor de colas. Pero en una situación de recuperación tras un error, cuando no hay ninguna aplicación en ejecución, el gestor de transacciones puede utilizar un canal MQI dedicado para recuperar las unidades incompletas de trabajo en las que el gestor de colas estaba participando en el momento de producirse la anomalía.

En esta sección, un IBM MQ MQI client que no tiene la función transaccional extendida se conoce como cliente base de IBM MQ. Por lo tanto, puede considerar que un cliente transaccional extendido de IBM MQ consta de un cliente base de IBM MQ y además incorpora la función transaccional extendida.

**Nota:**  IBM MQ MQI client en IBM i no da soporte a la función transaccional extendida de IBM MQ.

## Soporte de plataforma para clientes transaccionales extendidos

### Multi

Los clientes transacciones extendidos están disponibles en todas las Multiplatforms que soportan un cliente base. Los clientes no están disponibles para z/OS.

Una aplicación cliente que utiliza un cliente transaccional extendido sólo puede conectarse a un gestor de colas de los siguientes productos de IBM MQ 9.0o posterior:

-  IBM MQ for AIX
-  IBM MQ for IBM i
-  IBM MQ para Linux
-  IBM MQ for Windows

 Aunque no hay clientes transaccionales extendidos que se ejecutan en z/OS, una aplicación de cliente que utiliza un cliente transaccional extendido se puede conectar a un gestor de colas que se ejecuta en z/OS.

Para cada plataforma, los requisitos de hardware y software para el cliente transaccional extendido son los mismos que los requisitos para el cliente base de IBM MQ. Un lenguaje de programación está soportado por un cliente transaccional extendido si está soportado por el cliente base de IBM MQ y por el gestor de transacciones que se está utilizando.

Para obtener información sobre los gestores de transacciones externas para todas las plataformas, consulte [Requisitos del sistema para IBM MQ](#).

## Cómo se conecta el cliente al servidor

Un cliente se conecta a un servidor utilizando MQCONN o MQCONNX y se comunica a través de un canal.

Una aplicación que se ejecuta en el entorno de cliente de IBM MQ debe mantener una conexión activa entre las máquinas cliente y servidor.

La conexión la establece una aplicación al emitir una llamada MQCONN o MQCONNX. Los clientes y los servidores se comunican mediante *canales MQI* o bien, cuando se utiliza el compartición de conversaciones, las conversaciones comparten cada una instancia de canal MQI. Cuando la llamada se realiza satisfactoriamente, la instancia de canal MQI o la conversación permanece conectada hasta que la aplicación emita una llamada MQDISC. Este es el caso de cada gestor de colas al que una aplicación necesita conectarse.

## Un cliente y un gestor de colas en la misma máquina

También puede ejecutar una aplicación en el entorno IBM MQ MQI client cuando la máquina también tiene instalado un gestor de colas.

En esta situación, tiene la opción de enlazarse a las bibliotecas del gestor de colas o a las bibliotecas del cliente, pero recuerde que si se enlaza a las bibliotecas del cliente, sigue siendo necesario definir las conexiones de canal. Esto puede ser útil durante la fase de desarrollo de una aplicación. Puede probar el programa en su propia máquina, sin depender de otros, y tener confianza en que seguirá funcionando cuando lo traslade a un entorno IBM MQ MQI client independiente.

## Cientes en plataformas diferentes

En este ejemplo, la máquina servidor se comunica con tres IBM MQ MQI clients en distintas plataformas.

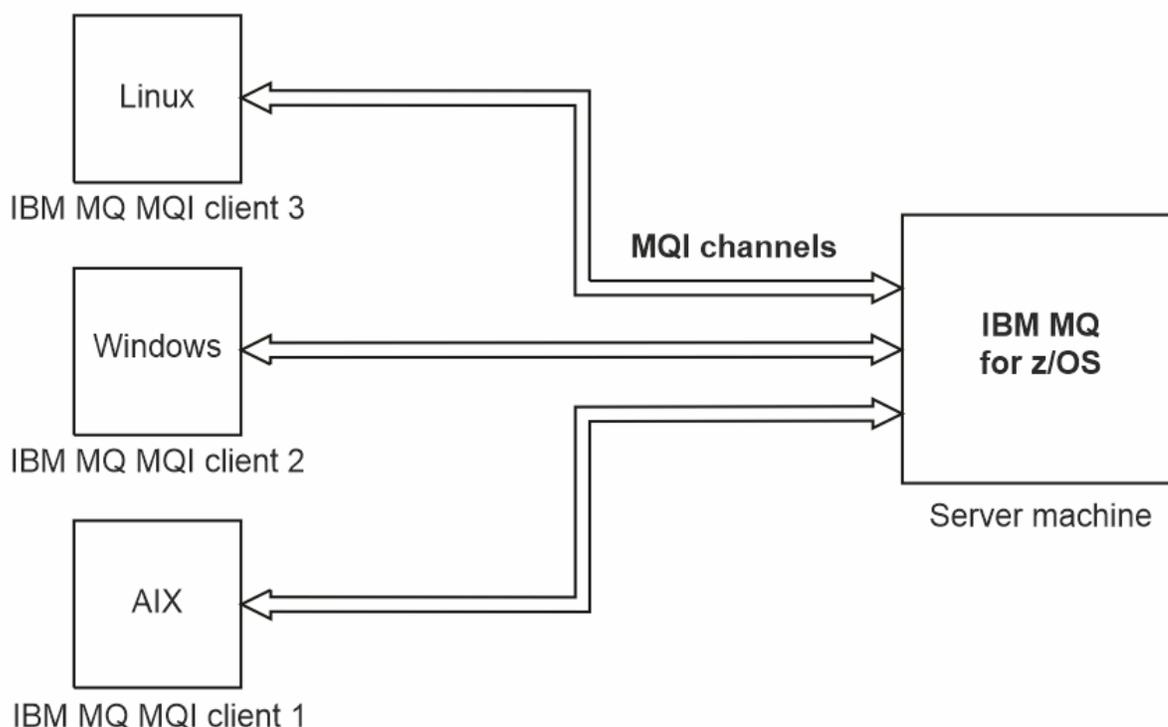


Figura 52. Servidor de IBM MQ conectado a clientes en distintas plataformas

Son posibles otros entornos más complejos. Por ejemplo, un cliente de IBM MQ se puede conectar a más de un gestor de colas o a cualquier número de gestores de colas conectados como parte de un grupo de compartición de colas.

### Utilización de versiones diferentes de software de cliente y de servidor

Si utiliza versiones anteriores de productos IBM MQ, asegúrese de que el servidor da soporte a la conversión de código del CCSID del cliente.

Un cliente IBM MQ puede conectarse a todas las versiones soportadas del gestor de colas. Si se está conectando a un gestor de colas de versión anterior, no puede utilizar las características y estructuras de una versión posterior del producto en la aplicación IBM MQ en el cliente.

Un gestor de colas de IBM MQ puede comunicarse con clientes en diferentes versiones consigo mismo negociando hasta el nivel de protocolo más alto soportado mutuamente. Esto significa que los clientes más antiguos se pueden utilizar con niveles de gestor de colas posteriores. Se recomienda que tanto el cliente como el servidor estén en versiones de IBM MQ que estén actualmente en soporte para facilitar el diagnóstico de problemas y habilitar el soporte por parte de IBM.

Para obtener más información, consulte los lenguajes de programación soportados en [Desarrollo de aplicaciones](#).

## Gestión y soporte de transacciones

Introducción a la gestión de transacciones y cómo IBM MQ da soporte a las transacciones.

Un *gestor de recursos* es un subsistema informático que posee y gestiona recursos a los que las aplicaciones pueden acceder y que pueden actualizar. A continuación, se muestran algunos ejemplos de gestores de recursos:

- Un gestor de colas de IBM MQ, con recursos que son sus colas
- Una base de datos de Db2, con recursos que son las tablas

Cuando una aplicación actualiza los recursos de uno o varios gestores de recursos, es posible que haya un requisito empresarial para garantizar que determinadas actualizaciones se completan todas satisfactoriamente como un grupo o no se completará ninguna de ellas. El motivo de este tipo de requisito es que los datos empresariales podrían quedarse en un estado incoherente si sólo se completaran correctamente algunas de estas actualizaciones y otras no.

Las actualizaciones de los recursos que se gestionan de este modo se dice que ocurren dentro de una *unidad de trabajo* o una *transacción*. Un programa de aplicación puede agrupar un conjunto de actualizaciones en una unidad de trabajo.

Durante una unidad de trabajo, una aplicación emite peticiones a los gestores de recursos para actualizar sus recursos. La unidad de trabajo finaliza cuando la aplicación emite una petición para confirmar todas las actualizaciones. Hasta que las actualizaciones se confirmen, ninguna de ellas estará visible para las otras aplicaciones que acceden a los mismos recursos. De forma alternativa, si la aplicación decide que no puede completar la unidad de trabajo por algún motivo, puede emitir una petición para restituir todas las actualizaciones que haya solicitado hasta ese momento. En este caso, ninguna de las actualizaciones estará nunca visible para otras aplicaciones. Normalmente, estas actualizaciones están relacionadas lógicamente y deben ejecutarse todas satisfactoriamente para mantener la integridad de los datos. Si una actualización se ejecuta satisfactoriamente y otra no, se pierde la integridad de los datos.

Cuando una unidad de trabajo se ejecuta satisfactoriamente, se dice que se *confirma*. Una vez confirmada, todas las actualizaciones realizadas dentro de esa unidad de trabajo se convierten en persistentes e irreversibles. Sin embargo, si la unidad de trabajo no se ejecuta correctamente, todas las actualizaciones se *restituyen*. Este proceso, en el que las unidades de trabajo se confirman o se restituyen con integridad, se conoce como *coordinación del punto de sincronismo*.

El momento específico en el que todas las actualizaciones en una unidad de trabajo se han confirmado o se han restituido se denomina *punto de sincronismo*. Una actualización en una unidad de trabajo se dice que ocurre *dentro del control de punto de sincronismo*. Si una aplicación solicita una actualización que está *fuera del control de punto de sincronismo*, el gestor de recursos confirma inmediatamente la actualización, aunque haya una unidad de trabajo en curso, y la actualización no se puede restituir más tarde.

El subsistema informático que gestiona unidades de trabajo se denomina *gestión de transacciones* o *coordinador de puntos*.

Una unidad de trabajo de *local* es aquella en la que los únicos recursos actualizados son los del gestor de colas de IBM MQ. En ese caso, la coordinación del punto de sincronismo la proporciona el propio gestor de colas utilizando el proceso de confirmación en una sola fase.

Una unidad de trabajo *global* es aquella en la que los recursos que pertenecen a otros gestores de recursos, como bases de datos conformes a las normas XA, también se actualizan. En este caso, debe utilizarse un procedimiento de confirmación en dos fases y la unidad de trabajo la puede coordinar el propio gestor de colas, o bien externamente otro gestor de transacciones que cumpla las normas XA, como por ejemplo IBM TXSeries o BEA Tuxedo.

Un gestor de transacciones es el responsable de garantizar que todas las actualizaciones de recursos en una unidad de trabajo se completen correctamente o bien ninguna de ellas se completa. Es a un gestor de transacciones a quien la aplicación emite una petición para confirmar o restituir una unidad de trabajo. Entre los ejemplos de gestores de transacciones están CICS y WebSphere Application Server, aunque ambos poseen también otra función.

Algunos gestores de recursos tienen su propia función de gestión de transacciones. Por ejemplo, un gestor de colas de IBM MQ puede gestionar unidades de trabajo que impliquen actualizaciones de sus propios recursos y actualizaciones en las tablas de Db2. El gestor de colas no necesita un gestor de transacciones independiente para realizar esta función, aunque se puede utilizar uno, en caso de que sea requisito del usuario. Si se utiliza un gestor de transacciones independiente, se denominará *gestor de transacciones externo*.

Para que un gestor de transacciones externo gestione una unidad de trabajo, debe haber una interfaz estándar entre el gestor de transacciones y cada gestor de recursos que participa en la unidad de trabajo. Esta interfaz permite que el gestor de transacciones y un gestor de recursos se comuniquen entre sí. Una de estas interfaces es la *Interfaz XA*, que es una interfaz estándar soportada por una serie de gestores de

transacciones y gestores de recursos. La Interfaz XA está publicada por The Open Group en *Distributed Transaction Processing: The XA Specification*.

Cuando más de un gestor de recursos participa en una unidad de trabajo, un gestor de transacciones debe utilizar un protocolo de *confirmación en dos fases* para asegurarse de que todas las actualizaciones dentro de la unidad de trabajo se completan satisfactoriamente o bien ninguna de ellas se completará, aunque haya una anomalía en el sistema. Cuando una aplicación emite una petición a un gestor de transacciones para confirmar una unidad de trabajo, el gestor de transacciones realiza lo siguiente:

#### **Fase 1 (preparar la confirmación)**

El gestor de transacciones pregunta a cada gestor de recursos que participa en la unidad de trabajo para asegurarse de que toda la información sobre las actualizaciones previstas de sus recursos esté en un estado recuperable. Normalmente, un gestor de recursos es quien lo hace grabando la información en un registro y asegurándose de que la información se graba a través del disco duro. La Fase 1 se completa cuando el gestor de transacciones recibe notificación de cada gestor de recursos de que la información sobre las actualizaciones previstas de sus recursos se encuentran en un estado recuperable.

#### **Fase 2 (confirmar)**

Cuando la Fase 1 se ha completado, el gestor de transacciones toma la decisión irrevocable de confirmar la unidad de trabajo. Solicita a cada gestor de recursos que participa en la unidad de trabajo que confirme las actualizaciones de sus recursos. Cuando un gestor de recursos recibe esta petición, debe confirmar las actualizaciones. No tiene la opción de restituirlas en esta fase. La Fase 2 se completa cuando el gestor de transacciones recibe notificación de cada gestor de recursos de que ha confirmado las actualizaciones de los recursos.

La Interfaz XA utiliza un protocolo de confirmación en dos fases.

Para obtener más información, consulte [Escenarios de soporte transaccional](#).

IBM MQ también proporciona soporte para Microsoft Transaction Server (COM+). [Utilización de Microsoft Transaction Server \(COM+\)](#) proporciona información sobre cómo configurar IBM MQ para aprovechar el soporte de COM+.

## **Extensión de recursos del gestor de colas**

---

Puede ampliar los recursos del gestor de colas utilizando las salidas de usuario, las salidas de API, o los servicios instalables.

### **Salidas de usuario**

Las salidas de usuario proporcionan un mecanismo para que los usuarios inserten su propio código en una función del gestor de colas. Las salidas de usuario soportadas son:

#### **Salidas de canal**

Estas salidas cambian el modo en que funcionan los canales. Las salidas de canal se describen en [Programas de salida de canal para canales de mensajes](#).

#### **Salidas de conversión de datos**

Estas salidas crean fragmentos de código fuente que pueden transferirse a programas de aplicación para convertir datos de un formato a otro. Las salidas de conversiones de datos se describen en [Escritura de salidas de conversión de datos](#).

#### **La salida de carga de trabajo del clúster**

La función realizada por esta salida la define el proveedor de la salida. Se proporciona información de definición de llamada en [MQ\\_CLUSTER\\_WORKLOAD\\_EXIT - Descripción de llamada](#).

### **Salidas de API**

Las salidas de API permiten escribir código que cambia el comportamiento de las llamadas de API de IBM MQ, como, por ejemplo, MQPUT y MQGET, e insertar ese código inmediatamente antes o inmediatamente después de esas llamadas. La inserción se realiza de forma automática. El gestor de colas dirige el código

de salida a los puntos registrados. Para obtener más información sobre las salidas de la API, consulte [Utilización y escritura de salidas de API](#).

## Servicios instalables

Los servicios instalables tienen interfaces formalizadas (una API) con varios puntos de entrada.

Una implementación de un servicio instalable se denomina *componente de servicio*. Puede utilizar los componentes suministrados con IBM MQ o puede escribir su propio componente para ejecutar las funciones que necesita.

Los servicios instalables que se proporcionan actualmente son los siguientes:

### Servicio de autorización

El servicio de autorización le permite crear su propio recurso de seguridad.

El componente de servicio predeterminado que implementa el servicio es el gestor de autorizaciones sobre objetos (OAM). De forma predeterminada el OAM está activo, es decir, no tiene que hacer nada para configurarlo. Puede utilizar la interfaz del servicio de autorización para crear otros componentes que sustituyan o aumenten el OAM. Para obtener más información sobre el OAM, consulte [Configuración de la seguridad en sistemas AIX, Linux, and Windows](#).

### Servicio de nombres

El servicio de nombres permite a las aplicaciones compartir colas ya que se identifican las colas remotas como si fuesen colas locales.

Puede escribir su propio componente de servicio de nombres. Es posible que desee hacerlo si tiene la intención de utilizar el servicio de nombres con IBM MQ, por ejemplo. Para utilizar el servicio de nombres debe tener un componente escrito por el usuario o un componente proporcionado por un proveedor de software distinto. De forma predeterminada, el servicio de nombres está inactivo.

### Conceptos relacionados

[Salidas de usuario, salidas de API y servicios instalables de IBM MQ](#)

## Interfaces de lenguaje de IBM MQ Java

IBM MQ proporciona tres interfaces de programación de aplicaciones (API) para su uso en aplicaciones Java : IBM MQ classes for Jakarta Messaging, IBM MQ classes for JMSy IBM MQ classes for Java.

IBM soporta, y es un participante activo en, estándares abiertos.

- A partir de IBM MQ 8.0, el producto implementa el estándar JMS 2.0 , que introdujo una nueva API simplificada junto con características como suscripciones compartidas.
-   A partir de IBM MQ 9.3.0, también se da soporte a [Jakarta Messaging 3.0](#) .
- Además, WebSphere Liberty da soporte a JMS 2.0 y Jakarta Messaging 3.0 con IBM MQ.

Dentro de IBM MQ hay tres API para su uso en aplicaciones Java :

### **IBM MQ classes for Jakarta Messaging**

IBM MQ classes for Jakarta Messaging es un proveedor de Jakarta Messaging que implementa las interfaces de Jakarta Messaging para IBM MQ como el sistema de mensajería. Jakarta Connectors Architecture proporciona una forma estándar de conectar aplicaciones que se ejecutan en un entorno de Jakarta EE a un sistema de información empresarial (EIS) como, por ejemplo, IBM MQ o Db2.

### **IBM MQ classes for JMS**

IBM MQ classes for JMS es un proveedor de JMS que implementa las interfaces de JMS para IBM MQ como el sistema de mensajería. Java Platform, Enterprise Edition Connector Architecture (JCA) proporciona una forma estándar de conectar aplicaciones que se ejecutan en un entorno Java EE con un EIS (Enterprise Information System) como IBM MQ o Db2.

## IBM MQ classes for Java

IBM MQ classes for Java permite utilizar IBM MQ en un entorno Java. IBM MQ classes for Java permite que una aplicación Java se conecte con IBM MQ como un cliente IBM MQ o se conecte directamente a un gestor de colas de IBM MQ.

### Nota:

- ▶ **V 9.3.0** ▶ **V 9.3.0** JMS 2.0 ha sido reemplazado por Jakarta Messaging. IBM MQ classes for JMS sigue dando soporte al estándar JMS 2.0, pero las futuras mejoras en la mensajería de Java solo surgirán en Jakarta Messaging, por lo tanto en IBM MQ classes for Jakarta Messaging. Los IBM MQ classes for JMS sólo se recomiendan para mantener y ampliar las aplicaciones JMS 2.0 existentes. IBM MQ classes for Jakarta Messaging debe ser la tecnología preferida para el nuevo desarrollo.
- ▶ **Stabilized** IBM MQ classes for Java se estabilizan funcionalmente en el nivel suministrado en IBM MQ 8.0. Las aplicaciones existentes que utilizan IBM MQ classes for Java seguirán estando totalmente soportadas, pero esta API se estabiliza, por lo que no se añadirán nuevas características y se rechazarán las solicitudes de mejoras. Soporte completo significa que los defectos se solucionarán junto con los cambios necesarios por los cambios en los requisitos del sistema IBM MQ.

▶ **V 9.3.0** ▶ **JM 3.0** ▶ **V 9.3.0** A partir de IBM MQ 9.3, IBM MQ classes for Java, IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging se crean con Java 8. Los entornos de ejecución de Java en o por encima de estos niveles deben utilizarse para ejecutar aplicaciones utilizando estas interfaces.

### Conceptos relacionados

Acceso a IBM MQ desde Java -Opción de API

▶ **V 9.3.0** ▶ **V 9.3.0** [¿Por qué debo utilizar clases de IBM MQ para Jakarta Messaging?](#)

[¿Por qué debo utilizar IBM MQ classes for JMS?](#)

[¿Por qué debo utilizar IBM MQ classes for Java?](#)

## IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging son los proveedores de mensajería que se proporcionan con IBM MQ. Cada uno de estos proveedores también proporciona dos conjuntos de extensiones a la API de mensajería. Las aplicaciones Java Platform, Standard Edition (Java SE) y Java Platform, Enterprise Edition (Java EE) pueden utilizar estos proveedores de mensajería.

▶ **V 9.3.0** ▶ **JM 3.0** ▶ **V 9.3.0** IBM MQ 9.3.0 introduce soporte para [Jakarta Messaging 3.0](#). JMS 2.0 sigue siendo totalmente compatible.

Las especificaciones JMS y Jakarta Messaging definen un conjunto de interfaces que las aplicaciones pueden utilizar para realizar operaciones de mensajería. Desde IBM MQ 8.0, el producto da soporte a la versión JMS 2.0 del estándar JMS. Esta implementación ofrece todas las funcionalidades de la API clásica, pero requiere menos interfaces y es más sencilla de utilizar. Para obtener más información, consulte “Modelo JMS y Jakarta Messaging” en la [página 157](#) y la especificación JMS 2.0 en [Java .net](#).

▶ **JM 3.0** A partir de IBM MQ 9.3.0, Jakarta Messaging también está soportado.

El paquete `jakarta.jms` (Jakarta Messaging 3.0) o `javax.jms` (JMS 2.0) especifica los detalles de las interfaces de mensajería y un proveedor de mensajería implementa estas interfaces para un producto de mensajería específico. Por ejemplo:

- IBM MQ classes for JMS son proveedores de JMS e implementan las interfaces JMS para IBM MQ y también proporcionan los dos conjuntos de extensiones siguiente a la API de JMS:
  - Extensiones de IBM MQJMS
  - Extensiones de IBMJMS
- Una fábrica de conexiones, cola u objeto de tema creado utilizando `javax.dimso jakarta.jms`, las interfaces o cualquier conjunto de extensiones de JMS se pueden dirigir utilizando cualquiera de estas

API; es decir, se pueden convertir a cualquiera de las interfaces. Para mantener la portabilidad de las aplicaciones al máximo nivel, utilice las API más genéricas que se ajusten a sus requisitos.

Dado que JMS y Jakarta Messaging tienen mucho en común, las referencias posteriores a JMS en este tema pueden entenderse como referencias a ambos. En caso necesario, se señalan las diferencias.

## Extensiones de IBM MQJMS

IBM MQ classes for JMS también proporciona extensiones a la API de JMS. IBM MQ classes for JMS contiene extensiones que se implementan en objetos MQConnectionFactory, MQQueue y MQTopic. Estos objetos tienen propiedades y métodos que son específicos de IBM MQ. Los objetos pueden ser objetos administrados, o una aplicación puede crearlos dinámicamente en tiempo de ejecución. Estas extensiones se denominan extensiones de IBM MQ JMS. Tenga en cuenta que, en esta documentación, los objetos creados dinámicamente por una aplicación en tiempo de ejecución no se consideran objetos administrados.

## Extensiones de IBMJMS

Además de las extensiones de IBM MQ JMS, IBM MQ classes for JMS proporciona un conjunto más genérico de extensiones a la API de JMS o Java como el lenguaje de programación utilizado. Estas extensiones se denominan extensiones de IBM JMS y tienen los siguientes objetivos generales:

- Para proporcionar un mayor nivel de coherencia entre los proveedores de IBM JMS.
- Para facilitar la escritura de una aplicación puente entre dos sistemas de mensajería de IBM.
- Para facilitar el puerto de una aplicación de un proveedor de IBM JMS a otro.

El punto central de interés de estas extensiones es crear y configurar fábricas de conexiones y destinos de forma dinámica durante la ejecución, pero las extensiones también proporcionan una función que no está directamente relacionada con la mensajería como, por ejemplo, una función para la determinación de problemas.

### Tareas relacionadas

[Utilización de clases IBM MQ para JMS/Jakarta Messaging](#)

[Configuración de recursos JMS y Jakarta Messaging](#)

## IBM MQ classes for Jakarta Messaging: una visión general

IBM MQ 9.3.0 introduce soporte para Jakarta Messaging. Para Jakarta Messaging 3.0, el control de la especificación JMS se ha movido de Oracle al proceso de comunidad de Java. Sin embargo, Oracle conserva el control del nombre "javax", que se utiliza en otras tecnologías de la web (Java). Así que, aunque Jakarta Messaging 3.0 es funcionalmente equivalente a JMS 2.0, hay algunas diferencias en la nomenclatura. El nombre oficial de la versión 3.0 es Jakarta Messaging en lugar de Java Message Service, y los nombres de paquete y constantes tienen el prefijo `jakarta` en lugar de `javax`.

## En segundo plano

Durante muchos años, la plataforma Java se ha presentado en dos formatos: Standard Edition y Enterprise Edition.

Java Platform, Standard Edition (a veces abreviado como Java SE) es el lenguaje principal y las bibliotecas de clases, que pueden ejecutarse en un contexto autónomo. La mayoría de los paquetes de Java en Java SE tienen nombres que empiezan por "java".

Java Platform, Enterprise Edition (Java EE) amplía esto, añadiendo funciones como Mensajería, varios beans, transaccionalidad, etc. Algunas de estas tecnologías también se pueden utilizar en un contexto Java SE. La mayoría de los paquetes de Java en Java EE históricamente tienen nombres que empiezan por "javax."-sin embargo, hay algún cruce, por lo que algunos paquetes de Java SE tienen "javax." como prefijo de su nombre.

Java Message Service (JMS) forma parte de Java Platform, Enterprise Edition. Java EE 7 incorpora JMS 2.0.

Hasta Java EE 7, las tecnologías estaban bajo la dirección de Oracle.

Las tecnologías Java EE han pasado recientemente de la administración de Oracle a un proceso comunitario supervisado por la Fundación Eclipse .

Como el "javax". no se ha podido mover el nombre al nuevo proyecto, se ha adoptado un nuevo nombre- todos los paquetes y nombres de propiedad tienen ahora el prefijo "jakarta". y Java Platform, Enterprise Edition se denominará "Jakarta EE" en el futuro. La numeración de la versión ha continuado: la versión 8 era una versión provisional que se puede ignorar en gran medida, y Jakarta EE 9 es el punto en el que el "jakarta". el prefijo entra en vigor.

La tecnología Jakarta EE principal que se aplica en el contexto IBM MQ es Jakarta Messaging 3.0 -el sucesor de Java Message Service (JMS) 2.0. Por lo tanto, Jakarta EE 9 incorpora Jakarta Messaging 3.0.

IBM MQ sigue dando soporte a Java EE 7 y JMS 2.0, al tiempo que introduce soporte para Jakarta EE 9 y Jakarta Messaging 3.0.

## Qué se entrega: Java SE

Para Java Platform, Standard Edition, además de IBM MQ classes for JMS (que da soporte a las operaciones de JMS 2.0 con IBM MQ), IBM MQ 9.3.0 proporciona IBM MQ classes for Jakarta Messaging. Estas clases proporcionan un proveedor de Jakarta Messaging 3.0 que se integra con IBM MQ, permitiendo el uso de gestores de colas de IBM MQ para facilitar las operaciones de Jakarta Messaging .

Estos se proporcionan como un archivo JAR estándar, `com.ibm.mq.jakarta.client.jar`, en el subdirectorio `java/lib` de la instalación de IBM MQ .

Para su uso en contenedores OSGi, como Apache Felix o Eclipse Equinox, IBM MQ también proporciona un par de paquetes OSGi:

- `com.ibm.mq.osgi.jms30.clientprereqs_V.R.M.F.jar`
- `com.ibm.mq.osgi.jms30.client_V.R.M.F.jar`

donde *V.R.M.F* representa la versión de IBM MQ, por ejemplo 9.3.0.0. Estos paquetes se pueden encontrar en el subdirectorio `java/lib/OSGi` de la instalación de IBM MQ .

## ¿Qué se entrega?: Jakarta EE 9 y Jakarta EE 10

Para admitir la mensajería basada en IBM MQ en servidores de aplicaciones compatibles con Jakarta EE 9 y posteriores, IBM MQ proporciona un adaptador de recursos para Jakarta Messaging : `wmq.jakarta.jmsra.rar`. Esto se puede encontrar en el subdirectorio `java/lib/jca` de la instalación IBM MQ.

IBM MQ sigue proporcionando un adaptador de recursos compatible con Java EE 7, `wmq.jmsra.rar`, en el subdirectorio `java/lib/jca` de la instalación IBM MQ.

## Cómo se entregan estos artefactos

Estos archivos JAR y el archivo RAR para el adaptador de recursos se empaquetan con los artefactos preexistentes en el soporte de instalación habitual de IBM MQ, tanto el soporte de instalación específico de la plataforma, como los archivos « .rpm », como el soporte redistribuible, como los archivos JAR de cliente redistribuibles autoextraíbles.

## Qué ha cambiado entre JMS 2.0 y Jakarta Messaging 3.0

Jakarta EE 9 y Jakarta Messaging 3.0 no presentan ninguna funcionalidad nueva. Todo lo que cambia son nombres. Por ejemplo, si utiliza "javax.jms.Connection" en JMS 2.0, utilice "jakarta.jms.Connection" en Jakarta Messaging 3.0.

A medida que Eclipse Foundation reenvíe la plataforma Jakarta EE , se construirá sobre esta base y este convenio de denominación se utilizará para la nueva funcionalidad introducida en el futuro.

## Qué ha cambiado entre IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging

### Resumen

IBM MQ classes for JMS, que proporcionan soporte para JMS 2.0, permanecen disponibles y se recomiendan principalmente para mantener y ampliar las aplicaciones existentes. Están totalmente soportados.

IBM MQ classes for Jakarta Messaging, que proporcionan soporte para Jakarta Messaging 3.0, se recomiendan para el nuevo desarrollo.

En IBM MQ 9.3.0, estas dos ofertas son funcionalmente equivalentes. Sólo la denominación difiere. Sin embargo, es más probable que surjan nuevas funciones de mensajería en IBM MQ classes for Jakarta Messaging que en IBM MQ classes for JMS.

Las dos ofertas son interoperables. Los mensajes producidos por IBM MQ classes for JMS pueden ser consumidos por IBM MQ classes for Jakarta Messaging y viceversa. Pero las dos ofertas no deben coexistir en una sola aplicación.

### Cambios en la nomenclatura

<i>Tabla 16. Cambios en los nombres de paquete</i>	
<b>IBM MQ classes for JMS nombre de paquete</b>	<b>IBM MQ classes for Jakarta Messaging nombre de paquete</b>
com.ibm.mq.jms[*]	com.ibm.mq.jakarta.jms[*]
com.ibm.jms	com.ibm.jakarta.jms
com.ibm.msg.client.jms.*	com.ibm.msg.client.jakarta.jms.*
com.ibm.msg.client.wmq.*	com.ibm.msg.client.jakarta.wmq.*

Los paquetes relacionados con servicios comunes (rastreo, registro, soporte de idioma nacional, etc.) y las implementaciones de JMQUI (local y remota) son comunes a IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging, por lo que no es necesario realizar cambios en estas áreas.

Tenga en cuenta que los nombres de propiedad también han cambiado. Por ejemplo, la propiedad para habilitar las extensiones IBM MQ en IBM MQ classes for Jakarta Messaging es **com.ibm.mq.jakarta.jms.SupportMQExtensions**.

Los nombres de propiedad que son independientes de IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging, como las distintas propiedades de **com.ibm.msg.client.commonservices.trace.\*** , se aplican por igual a ambas ofertas.

### Programas de utilidad administrativos

Los programas de utilidad **crtmqenv** y **setmqenv** ahora aceptan una opción para especificar si la vía de acceso de clases debe configurarse para IBM MQ classes for JMS (-j 2.0) o IBM MQ classes for Jakarta Messaging (-j 3.0), y hay variantes IBM MQ classes for Jakarta Messaging de los programas de utilidad **runjms** , denominadas **runjms30** y nombres similares.

El programa de utilidad **dspmqr** , cuando se le solicite que informe sobre los componentes de Java , incluye IBM MQ classes for Jakarta Messaging en su salida.

Para configurar objetos IBM MQ classes for Jakarta Messaging para que se recuperen a través de JNDI, el nuevo programa de utilidad **JMS30Admin** es equivalente al programa de utilidad **JMSAdmin** para IBM MQ classes for JMS.

Tenga en cuenta que como los objetos subyacentes son de paquetes diferentes. Las definiciones JNDI creadas por **JMSAdmin** no pueden ser utilizadas por IBM MQ classes for Jakarta Messaging, ni tampoco las creadas por **JMS30Admin** pueden ser utilizadas por IBM MQ classes for JMS.

**Nota:** No hay soporte para objetos IBM MQ classes for Jakarta Messaging proporcionados por IBM MQ Explorer; su integración JNDI es solo para IBM MQ classes for JMS .

## Conceptos relacionados

[¿Por qué debo utilizar clases de IBM MQ para Jakarta Messaging?](#)

## Modelo JMS y Jakarta Messaging

El modelo JMS y Jakarta Messaging define un conjunto de interfaces que las aplicaciones Java pueden utilizar para realizar operaciones de mensajería. IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging son proveedores de mensajería que definen cómo están relacionados los objetos de mensajería de Java con los conceptos de IBM MQ . Las especificaciones JMS y Jakarta Messaging esperan que determinados objetos de mensajería sean objetos administrados.

A partir de IBM MQ 8.0, el producto da soporte a la versión JMS 2.0 del estándar JMS, que introdujo una API simplificada, al mismo tiempo que retiene la API clásica, de JMS 1.1.

 IBM MQ 9.3.0 introduce soporte para [Jakarta Messaging 3.0](#). JMS 2.0 sigue siendo totalmente compatible. Dado que JMS y Jakarta Messaging tienen mucho en común, las referencias posteriores a JMS en este tema pueden entenderse como referencias a ambos. En caso necesario, se señalan las diferencias.

## API simplificada

JMS 2.0 ha introducido la API simplificada, al tiempo que retiene las interfaces específicas del dominio e independientes del dominio de JMS 1.1. La API simplificada disminuye el número de objetos necesarios para enviar y recibir mensajes y consta de las interfaces siguientes:

### ConnectionFactory

ConnectionFactory es un objeto administrado que utiliza un cliente de JMS para crear una conexión. Esta interfaz también se utiliza en la API clásica.

### Contexto de JMS

Este objeto combina los objetos Conexión y Sesión de la API clásica. Se pueden crear objetos JMSContext a partir de otros objetos JMSContext, duplicando la conexión subyacente.

### JMSProducer

Se crea un JMSProducer mediante un JMSContext y se utiliza para enviar mensajes a una cola o tema. El objeto JMSProducer genera la creación de los objetos necesarios para enviar el mensaje.

### JMSConsumidor

Se crea un JMSConsumer mediante un JMSContext y se utiliza para recibir mensajes de un tema o una cola.

La API simplificada tiene varios efectos:

- El objeto JMSContext siempre inicia automáticamente la conexión subyacente.
- Ahora los JMSProducers y los JMSConsumers pueden trabajar directamente con los cuerpos de los mensajes, sin tener que obtener el objeto de mensaje completo, utilizando el método `getBody` del mensaje.
- Las propiedades de los mensajes se pueden establecer en el objeto JMSProducer mediante el encadenamiento de métodos, antes de enviar un 'body', un contenido de mensajes. El objeto JMSProducer manejará la creación de todos los objetos necesarios para enviar el mensaje. Utilizando JMS 2.0, se pueden establecer las propiedades y un envío de mensajes, como se indica a continuación:

```
context.createProducer().
setProperty("foo", "bar").
setTimeToLive(10000).
setDeliveryMode(NON_PERSISTENT).
```

```
setDisableMessageTimestamp(true).  
send(dataQueue, body);
```

JMS 2.0 también introdujo suscripciones compartidas donde los mensajes se pueden compartir entre varios consumidores. Todas las suscripciones de JMS 1.1 se tratan como suscripciones no compartidas.

## API clásica

La lista siguiente resume las interfaces principales de JMS de la API clásica:

### Destino

Un objeto Destination es la ubicación a la que una aplicación envía mensajes, o es el origen desde el que una aplicación recibe mensajes, o ambas cosas.

### ConnectionFactory

Un objeto ConnectionFactory encapsula un conjunto de propiedades de configuración de una conexión. Una aplicación utiliza una fábrica de conexiones para crear una conexión.

### Conexión

Un objeto Connection encapsula una conexión activa de una aplicación en un servidor de mensajería. Una aplicación utiliza una conexión para crear sesiones.

### Session

Una sesión es un contexto de hebra única para enviar y recibir mensajes. Una aplicación utiliza una sesión para crear mensajes, productores de mensajes y consumidores de mensajes. Una sesión es transaccional o no transaccional.

### Mensaje

Un objeto Message encapsula un mensaje que una aplicación envía o recibe.

### MessageProducer

Una aplicación utiliza un productor de mensajes para enviar mensajes a un destino.

### MessageConsumer

Una aplicación utiliza un consumidor de mensajes para recibir mensajes enviados a un destino.

Figura 53 en la página 158 muestra estos objetos y sus relaciones.

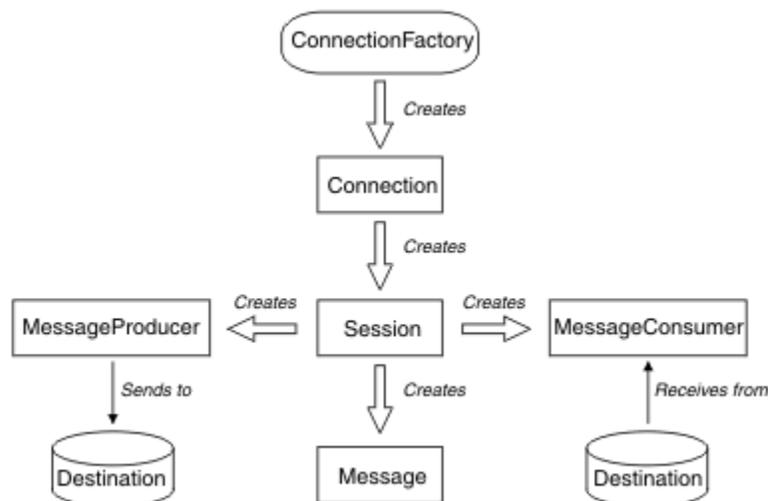


Figura 53. Objetos JMS y sus relaciones

El diagrama muestra las interfaces principales: ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message y Destination. Una aplicación utiliza una fábrica de conexiones para crear una conexión y utiliza una conexión para crear sesiones. La aplicación puede utilizar una sesión para crear mensajes, productores de mensajes y consumidores de mensajes. La aplicación utiliza un productor de mensajes para enviar mensajes a un destino, y utiliza un consumidor de mensajes para recibir mensajes enviados a un destino.

Un objeto Destination, ConnectionFactory o Connection lo pueden utilizar varias hebras al mismo tiempo, pero no pueden utilizar al mismo tiempo un objeto Session, MessageProducer o MessageConsumer. La forma más simple de asegurarse de que un objeto Session, MessageProducer o MessageConsumer no se utiliza simultáneamente es crear un objeto Session separado para cada hebra.

JMS da soporte a dos tipos de mensajería:

- Mensajería punto a punto
- Mensajería de publicación/suscripción

También se hace referencia a estos tipos de mensajería como *dominios de mensajería* y puede combinar ambos tipos de mensajería en una aplicación. En el dominio punto a punto, un destino es una cola y, en el dominio de publicación/suscripción, un destino es un tema.

En las versiones de JMS anteriores a JMS 1.1, la programación para los dominios punto a punto utiliza un conjunto de interfaces y métodos y la programación para los dominios de publicación/suscripción utiliza otro conjunto. Los dos conjuntos son similares, pero independientes. A partir de JMS 1.1, puede utilizar un conjunto común de interfaces y métodos que dan soporte a ambos dominios de mensajería. Las interfaces comunes proporcionan una vista independiente del dominio para cada dominio de mensajería. La Tabla 17 en la página 159 muestra las interfaces independientes del dominio de JMS y sus interfaces específicas del dominio correspondientes.

*Tabla 17. Las interfaces independientes del dominio de JMS y sus interfaces específicas del dominio correspondientes.*

<b>Interfaces independientes del dominio</b>	<b>Interfaces específicas del dominio para el dominio punto a punto</b>	<b>Interfaces específicas del dominio para el dominio de publicación/suscripción</b>
ConnectionFactory	QueueConnectionFactory	TopicConnectionFactory
Conexión	QueueConnection	TopicConnection
Destino	Cola	Tema
Session	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver QueueBrowser	TopicSubscriber

**JMS 2.0** IBM MQ classes for JMS 2.0 da soporte a las interfaces específicas de dominio de JMS 1.1 anteriores y a la API simplificada de JMS 2.0. Por lo tanto, IBM MQ classes for JMS 2.0 se puede utilizar para mantener aplicaciones existentes, incluido el desarrollo de nuevas funciones en aplicaciones existentes.

**JM 3.0** IBM MQ classes for Jakarta Messaging 3.0 da soporte a las versiones de Jakarta Messaging de las mismas interfaces y se recomienda para el desarrollo de nuevas aplicaciones.

En IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging, los objetos JMS están relacionados con los conceptos de IBM MQ de las maneras siguientes:

- Un objeto Connection tiene propiedades derivadas de las propiedades de la fábrica de conexiones utilizada para crear la conexión. Estas propiedades controlar cómo se conecta una aplicación a un gestor de colas. Los ejemplos de estas propiedades son el nombre del gestor de colas y, en el caso de una aplicación que se conecta al gestor de colas en modo cliente, el nombre de host o dirección IP del sistema en el que se ejecuta el gestor de colas.
- Un objeto Session encapsula un identificador de conexión IBM MQ, que por lo tanto define el ámbito transaccional de la sesión.

- Un objeto MessageProducer y un objeto MessageConsumer encapsula cada uno un manejador de objetos de IBM MQ.

Cuando se utiliza IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging, se aplican todas las reglas normales de IBM MQ . En concreto, tenga en cuenta que una aplicación puede enviar un mensaje a una cola remota pero solo puede recibir un mensaje de una cola propiedad del gestor de colas al que está conectada la aplicación.

La especificación JMS espera que los objetos ConnectionFactory y Destination sean objetos administrados. Un administrador crea y mantiene los objetos administrados en un repositorio central y una aplicación JMS recupera estos objetos utilizando la interfaz JNDI (Naming and Directory Interface) de Java.

En IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging, la implementación de la interfaz Destination es una superclase abstracta de Queue y Topic, por lo que una instancia de Destination es un objeto Queue o un objeto Topic. La interfaz independiente del dominio trata una cola o un tema como un destino. El dominio de mensajería para un objeto MessageProducer o MessageConsumer queda determinado por si el destino es una cola o un tema.

Por lo tanto, en IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging , los objetos de los tipos siguientes pueden ser objetos administrados:

- ConnectionFactory
- QueueConnectionFactory
- TopicConnectionFactory
- Cola
- Tema
- XAConnectionFactory
- XAQueueConnectionFactory
- XATopicConnectionFactory

## Arquitectura de IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging tienen una arquitectura en capas. La capa superior de código es una capa común que cualquier proveedor de mensajería de IBM Java puede utilizar.

 IBM MQ 9.3.0 introduce soporte para [Jakarta Messaging 3.0](#). JMS 2.0 sigue siendo totalmente compatible.

IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging tienen una arquitectura en capas tal como se muestra en el diagrama [Figura 54](#) en la [página 161](#). La capa superior de código es una capa común que puede utilizar cualquier proveedor de IBM JMS o Jakarta Messaging. Cuando una aplicación llama a un método JMS o Jakarta Messaging, cualquier proceso de la llamada que no sea específico de un sistema de mensajería lo realiza la capa común, que también proporciona una respuesta coherente a la llamada. Todo procesamiento de la llamada que sea específico de un sistema de mensajería se delega en una capa inferior. En el diagrama siguiente, el proveedor de mensajería de IBM MQ se muestra en la capa inferior, junto con otros dos proveedores de mensajería (proveedor de mensajería A y proveedor de mensajería B.)

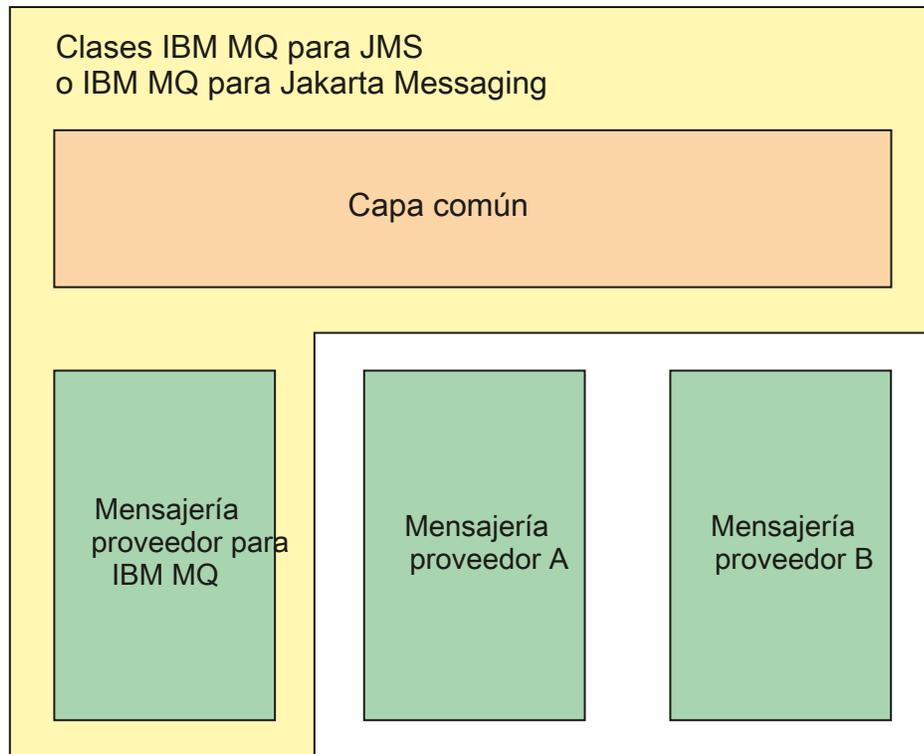


Figura 54. La arquitectura en capas para los proveedores de IBM JMS y Jakarta Messaging

Una arquitectura en capas cumple los siguientes objetivos:

- Para mejorar la coherencia del comportamiento de los diversos proveedores de IBM JMS y Jakarta Messaging
- Facilitar la escritura de una aplicación puente entre dos sistemas de mensajería de IBM
- Para facilitar el puerto de una aplicación de un proveedor de IBM JMS o Jakarta Messaging a otro

#### Tareas relacionadas

[Utilización de clases IBM MQ para JMS/Jakarta Messaging](#)

### Soporte para objetos administrados

IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging dan soporte al uso de objetos administrados.

**V 9.3.0** **JM 3.0** **V 9.3.0** Desde IBM MQ 9.3.0, Jakarta Messaging 3.0 es compatible con el desarrollo de nuevas aplicaciones. IBM MQ 9.3.0 sigue soportando JMS 2.0 para las aplicaciones existentes. No es posible utilizar tanto la API Jakarta Messaging 3.0 como la API JMS 2.0 en la misma aplicación. Para obtener más información, consulte [Uso de las clases IBM MQ para JMS/Jakarta Messaging](#).

El flujo de lógica dentro de una aplicación JMS o IBM MQ classes for Jakarta Messaging se inicia con ConnectionFactory y objetos Destination. La aplicación utiliza un objeto ConnectionFactory para crear un objeto Connection, que representa la conexión activa desde la aplicación hasta un servidor de mensajería. La aplicación utiliza el objeto Connection para crear un objeto Session, que es un contexto de hebra única para producir y consumir mensajes. A continuación, la aplicación puede utilizar el objeto Session y un objeto Destination para crear un objeto MessageProducer, que la aplicación utiliza para enviar mensajes al destino especificado. El destino es una cola o un tema del sistema de mensajería y está encapsulado por el objeto Destination. La aplicación también puede utilizar el objeto Session y un objeto

Destination para crear un objeto MessageConsumer, que la aplicación utiliza para recibir mensajes que se han enviado al destino especificado.

Las especificaciones JMS y Jakarta Messaging esperan que los objetos ConnectionFactory y Destination sean objetos administrados. Un administrador crea y mantiene objetos administrados en un repositorio central, y una aplicación JMS o Jakarta Messaging recupera estos objetos utilizando Java Naming Directory Interface (JNDI). El repositorio de objetos administrados puede ir desde un archivo simple a un directorio LDAP (Lightweight Directory Access Protocol).

IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging dan soporte al uso de objetos administrados. Una aplicación puede utilizar todas las características de IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging que se exponen a través de IBM MQ sin tener ninguna información específica de IBM MQ codificada en la propia aplicación. Este acuerdo proporciona a la aplicación un grado de independencia de la configuración de IBM MQ subyacente.

Para lograr esta independencia, la aplicación puede utilizar JNDI para recuperar fábricas de conexiones y destinos que se almacenan como objetos administrados, y utilizar sólo las interfaces definidas en el paquete `javax.jms` (JMS 2.0) o `jakarta.jms` (Jakarta Messaging 3.0) para realizar operaciones de mensajería.

**JMS 2.0** Para ' JMS 2.0, un administrador puede utilizar la herramienta de administración ' IBM MQ ' JMS ' **JMSAdmin**, o ' IBM MQ Explorer, para crear y mantener objetos administrados en un repositorio central.

**JM 3.0** Para Jakarta Messaging 3.0, no se puede administrar JNDI utilizando IBM MQ Explorer. La administración JNDI está soportada por la variante Jakarta Messaging 3.0 de **JMSAdmin**, que es **JMS30Admin**.

Normalmente, un servidor de aplicaciones proporciona su propio repositorio para objetos administrados y sus propias herramientas para crear y mantener los objetos. Por lo tanto, una aplicación Java

EE **JM 3.0** o Jakarta EE puede utilizar JNDI para recuperar objetos administrados del repositorio del servidor de aplicaciones o de un repositorio central.

### Tareas relacionadas

[Configuración de recursos JMS y Jakarta Messaging](#)

## Tipos de comunicación soportados en plataformas Java EE y Jakarta EE

En las plataformas Java EE y Jakarta EE , IBM MQ classes for JMS y IBM MQ classes for Jakarta Messaging dan soporte a dos tipos de comunicación entre un componente de una aplicación y un gestor de colas IBM MQ .

**V 9.3.0** **JM 3.0** **V 9.3.0** IBM MQ 9.3.0 introduce soporte para [Jakarta Messaging 3.0](#). JMS 2.0 sigue siendo totalmente compatible. Dado que JMS y Jakarta Messaging tienen mucho en común, las referencias posteriores a JMS en este tema pueden entenderse como referencias a ambos. En caso necesario, se señalan las diferencias.

Se da soporte a los dos tipos de comunicación siguientes entre un componente de una aplicación y un gestor de colas de IBM MQ:

- Comunicación de salida
- Comunicación de entrada

### Comunicación de salida

Utilizando la API JMS o Jakarta Messaging directamente, un componente de aplicación crea una conexión con un gestor de colas y, a continuación, envía y recibe mensajes.

Por ejemplo, el componente de la aplicación puede ser un cliente de aplicación, un servlet, una ASP (página de servidor activo) Java, un Enterprise Java Bean (EJB) o un bean controlado por mensaje (MDB). En este tipo de comunicación, el contenedor del servidor de aplicaciones solo proporciona funciones

de bajo nivel como soporte de las operaciones de mensajería como, por ejemplo, agrupaciones de conexiones y gestión de hilos.

## Comunicación de entrada

En el caso de la comunicación de entrada, un mensaje que llega a un destino se entrega a un MDB, que a su vez procesa el mensaje.

Las aplicaciones Java EE **JM 3.0** y Jakarta EE utilizan MDB para procesar mensajes de forma asíncrona. Un MDB actúa como escucha de mensajes JMS y se implementa con un método `onMessage()`, que define cómo se procesa un mensaje. Un MDB se despliega en el contenedor EJB de un servidor de aplicaciones. La forma precisa en que se configura un MDB depende del servidor de aplicaciones que se utilice, pero la información de configuración debe especificar a qué gestor de colas debe conectarse, cómo conectarse al mismo, qué destino debe supervisar mensajes y el comportamiento transaccional del MDB. El contenedor EJB utilizará esta información. Cuando un mensaje que cumple los criterios de selección del MDB llega al destino especificado, el contenedor EJB utiliza IBM MQ classes for JMS o IBM MQ classes for Jakarta Messaging para recuperar el mensaje del gestor de colas y, a continuación, entrega el mensaje al MDB llamando a su método `onMessage()`.

## Relación con las IBM MQ classes for Java

IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging y IBM MQ classes for JMS son iguales que utilizan una interfaz Java común a la MQI.

La Figura 55 en la página 163 muestra la relación entre IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging y IBM MQ classes for Java.

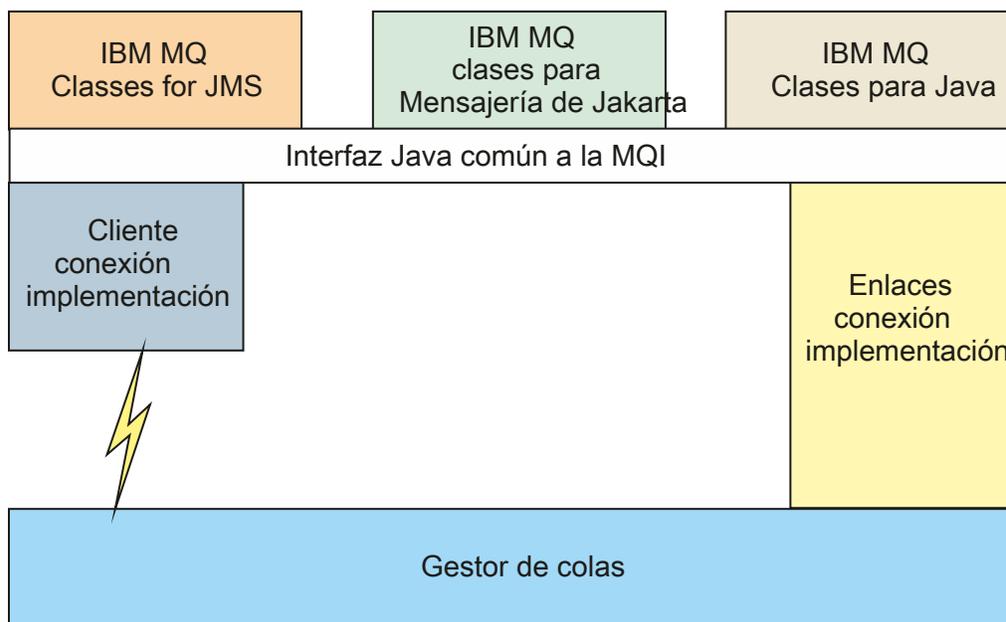


Figura 55. La relación entre IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging y IBM MQ classes for Java

En general, los programas Java sólo deben utilizar una interfaz para interactuar con IBM MQ - IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging o IBM MQ classes for JMS. La combinación de interfaces no está soportada, con una excepción. Para mantener la compatibilidad con los releases anteriores a IBM WebSphere MQ 7.0, las clases de salida de canal escritas en Java todavía pueden utilizar las interfaces IBM MQ classes for Java, incluso si las clases de salida de canal se llaman desde las IBM MQ classes for JMS. Sin embargo, el uso de las interfaces IBM MQ classes for Java significa que las aplicaciones siguen dependiendo de:

- **JMS 2.0** El archivo JAR de IBM MQ classes for Java , `com.ibm.mq.jar`. Si no desea `com.ibm.mq.jar` en la vía de acceso de clases, puede utilizar el conjunto de interfaces del paquete `com.ibm.mq.exits`.
- **V 9.3.0** **JM 3.0** **V 9.3.0** Uso de `com.ibm.mq.jakarta.client.jar`, al interoperar con IBM MQ classes for Jakarta Messaging.

### Conceptos relacionados

**V 9.3.0** **V 9.3.0** [¿Por qué debo utilizar clases de IBM MQ para Jakarta Messaging?](#)

[¿Por qué debo utilizar las clases IBM MQ para JMS?](#)

[¿Por qué debo utilizar las clases de IBM MQ para Java?](#)

## Proveedor de mensajería de IBM MQ

El proveedor de mensajería de IBM MQ tiene tres modalidades de operación: modalidad normal, modalidad normal con restricciones y modalidad de migración.

El proveedor de mensajería de IBM MQ tiene tres modalidades de funcionamiento:

- Modalidad normal del proveedor de mensajería de IBM MQ
- Modalidad normal con restricciones del proveedor de mensajería de IBM MQ
- Modalidad de migración del proveedor de mensajería de IBM MQ

La modalidad normal del proveedor de mensajería de IBM MQ utiliza todas las características de un gestor de colas de IBM MQ para implementar JMS. Esta modalidad está optimizada para utilizar la API y la funcionalidad de JMS 2.0 **V 9.3.0** **JM 3.0** **V 9.3.0** o [Jakarta Messaging 3.0](#) .

If:

- El cliente especifica una versión de proveedor de 6 en un **ConnectionFactory**, el cliente se comporta de una forma compatible con el cliente proporcionado con IBM WebSphere MQ 6.0. Solo se da soporte a las interfaces JMS 1.1 y JMS 2, pero algunas funciones de JMS 2, como suscripciones compartidas, retardo de entrega y envío asíncrono, están inhabilitadas. No hay compartimiento de conexión.
- El cliente especifica una versión de proveedor de 7 en un **ConnectionFactory**, tanto las interfaces JMS 1.1 como JMS 2 están totalmente soportadas.
- No se ha especificado ninguna versión de proveedor, se ha intentado conectar con el proveedor versión 7. Si esto falla, se realiza un intento adicional con la versión 6 del proveedor.

Si desea conectarse a IBM Integration Bus utilizando IBM MQ Enterprise Transport, utilice la modalidad de migración. Si utiliza IBM MQ Real-Time Transport, la modalidad de migración se selecciona automáticamente porque se han seleccionado de manera explícita en el objeto de fábrica de conexiones. La conexión con IBM Integration Bus utilizando IBM MQ Enterprise Transport sigue las reglas generales para la selección de modalidad que se describen en [Configuración de la propiedad JMS PROVIDERVERSION](#).

### Tareas relacionadas

[Configurar recursos de JMS](#)

## **z/OS** Conceptos de IBM MQ for z/OS

Algunos de los conceptos utilizados por IBM MQ for z/OS son exclusivos de la plataforma z/OS. Por ejemplo, el mecanismo de registro, las técnicas de gestión de almacenamiento, la unidad de eliminación de recuperación y los grupos de compartición de colas solo se suministran con IBM MQ for z/OS. Utilice este tema como introducción para obtener más información sobre estos conceptos.

Los conceptos incluyen una visión general de los objetos que utiliza IBM MQ for z/OS, incluyendo:

- El gestor de colas
- El iniciador de canal

- Colas compartidas y grupos de compartición de colas
- Transferencia a colas entre grupos

Los temas siguientes también cubren varios procedimientos que necesita, incluyendo:

- Definiciones de sistema en z/OS
- Gestión del almacenamiento
- Recuperación y reinicio
- Conceptos de seguridad en IBM MQ for z/OS

### **Conceptos relacionados**

[“El gestor de colas en z/OS” en la página 166](#)

Antes de permitir que los programas de aplicación utilicen IBM MQ en el sistema z/OS, debe instalar el producto IBM MQ for z/OS e iniciar un gestor de colas. El gestor de colas es propietario y gestiona el conjunto de recursos que IBM MQ utiliza..

[“El iniciador de canal en z/OS” en la página 167](#)

El iniciador de canal proporciona y gestiona recursos que habilitan la gestión de colas distribuidas de IBM MQ. IBM MQ utiliza *Agentes de canal de mensajes* (MCA) para enviar mensajes de un gestor de colas a otro.

[“Términos y tareas para gestionar IBM MQ for z/OS” en la página 168](#)

Utilice este tema como introducción a la terminología y tareas que son específicas de IBM MQ for z/OS.

[“Colas compartidas y grupos de compartición de colas” en la página 171](#)

Puede utilizar colas compartidas y grupos de compartición de colas para implementar la alta disponibilidad de recursos de IBM MQ. Las colas compartidas y los grupos de compartición de colas son funciones exclusivas de IBM MQ for z/OS en la plataforma z/OS.

[“Transferencia a colas entre grupos” en la página 219](#)

En esta sección se describe la gestión de colas dentro del grupo, una función de IBM MQ for z/OS exclusiva de la plataforma z/OS. Esta función sólo está disponible para los gestores de colas definidos en un grupo de compartición de colas.

[“Gestión de almacenamiento en z/OS” en la página 233](#)

IBM MQ for z/OS requiere estructuras de datos permanentes y temporales y utiliza conjuntos de páginas y almacenamientos intermedios de memoria para almacenar estos datos. Estos temas proporcionan más detalles sobre cómo IBM MQ utiliza estos conjuntos de páginas y almacenamientos intermedios.

[“Registro en IBM MQ for z/OS” en la página 238](#)

IBM MQ mantiene *registros* de cambios de datos y sucesos significativos a medida que se producen. Estos registros se pueden utilizar para recuperar los datos a un estado anterior si es necesario.

[“Recuperación y reinicio en z/OS” en la página 261](#)

Utilice los enlaces de este tema para obtener información sobre las características de IBM MQ for z/OS para el reinicio y la recuperación.

[“Conceptos de seguridad en IBM MQ for z/OS” en la página 279](#)

Utilice este tema para comprender la importancia de la seguridad para IBM MQ y las implicaciones de no tener los valores de seguridad adecuados en el sistema.

[“Disponibilidad en z/OS” en la página 286](#)

IBM MQ for z/OS tiene muchas características de alta disponibilidad. En este tema se describen algunas de las consideraciones de disponibilidad.

[“Disposición de unidad de recuperación en z/OS” en la página 291](#)

Determinadas aplicaciones transaccionales pueden utilizar una disposición de unidad de recuperación GROUP, en lugar de QMGR, al conectarse con un gestor de colas en un grupo de compartición de colas (QSG), especificando el nombre de QSG al conectarse en lugar del nombre del gestor de colas. Ello permite que la recuperación de transacciones sea más flexible y robusta eliminando el requisito de reconexión con el mismo gestor de colas en QSG.

### **Referencia relacionada**

[“Definición de sistema en z/OS” en la página 250](#)

IBM MQ for z/OS utiliza muchas definiciones de objeto predeterminadas y proporciona JCL de ejemplo para crear esos objetos predeterminados. Utilice este tema para comprender estos objetos predeterminados y el JCL ejemplo.

“Supervisión y estadísticas en IBM MQ for z/OS” en la página 289

IBM MQ for z/OS tiene un conjunto de recursos para supervisar el gestor de colas y recopilar estadísticas.

## z/OS El gestor de colas en z/OS

Antes de permitir que los programas de aplicación utilicen IBM MQ en el sistema z/OS, debe instalar el producto IBM MQ for z/OS e iniciar un gestor de colas. El gestor de colas es propietario y gestiona el conjunto de recursos que IBM MQ utiliza..

### El gestor de colas

Un *gestor de colas* es un programa que proporciona servicios de mensajería a las aplicaciones. Las aplicaciones que utilizan la Interfaz de Colas de Mensajes (MQI) pueden transferir mensajes a colas y obtener mensajes de colas. El gestor de colas garantiza que los mensajes se envían a la cola correcta o se dirigen a otro gestor de colas. El gestor de colas procesa tanto las llamadas MQI como los mandatos emitidos (independientemente de la procedencia). El gestor de colas genera los códigos de terminación adecuados para cada llamada o mandato.

Los recursos gestionados por el gestor de colas incluyen:

- Conjuntos de páginas que contienen las definiciones de objeto y los datos de mensaje de IBM MQ
- Registros que se utilizan para recuperar los mensajes y los objetos en caso de una anomalía del gestor de colas
- Almacenamiento del procesador
- Conexiones a través de las cuales diferentes entornos de aplicaciones (CICS, IMS y procesos por lotes) pueden acceder a la API de IBM MQ
- El iniciador de canal de IBM MQ, que permite la comunicación entre IBM MQ en el sistema z/OS y otros sistemas

El gestor de colas tiene un nombre, y las aplicaciones se conectan a él utilizando este nombre.

La Figura 56 en la página 166 ilustra un gestor de colas que muestra conexiones con distintos entornos de aplicación y el iniciador de canal.

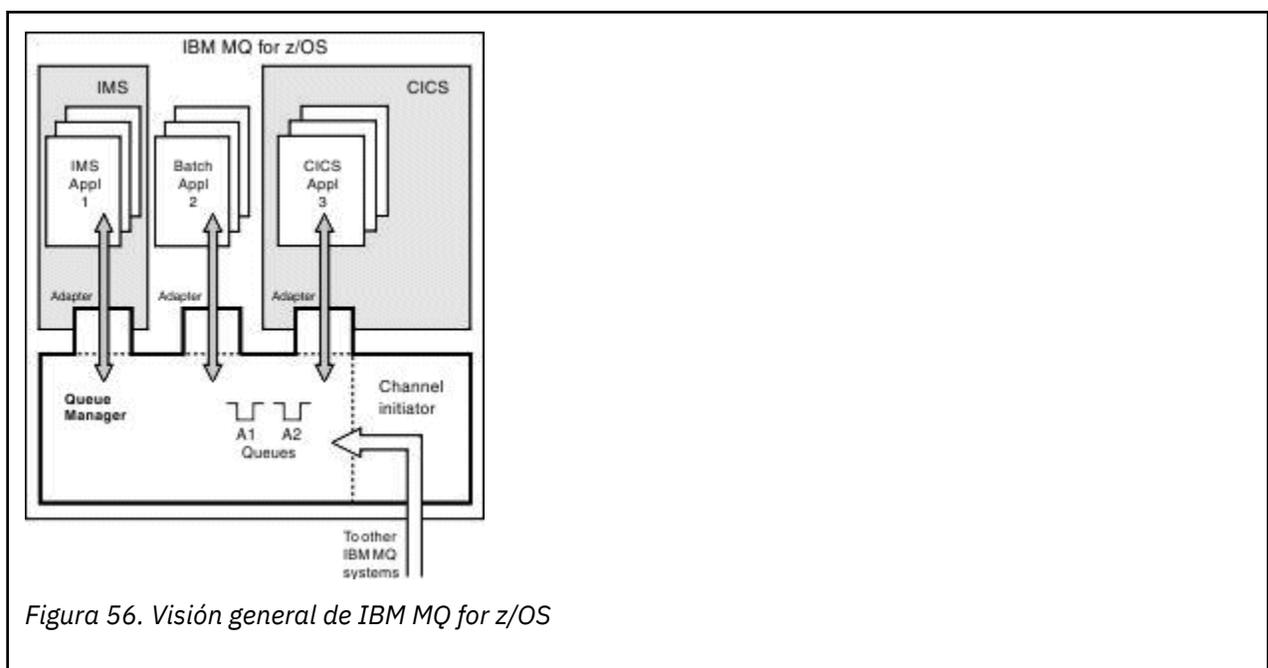


Figura 56. Visión general de IBM MQ for z/OS

## El subsistema del gestor de colas en z/OS

En z/OS, IBM MQ se ejecuta como un subsistema de z/OS que se inicia en el momento de la IPL. En el subsistema, el gestor de colas se crea ejecutando un procedimiento JCL que especifica los conjuntos de datos de z/OS que contienen información sobre los registros y que contienen definiciones de objeto (los conjuntos de páginas). El subsistema y el gestor de colas tienen el mismo nombre, de hasta cuatro caracteres. Todos los gestores de colas de la red deben tener nombres exclusivos, aunque estén en sistemas diferentes, sysplex o plataformas.

### z/OS El iniciador de canal en z/OS

El iniciador de canal proporciona y gestiona recursos que habilitan la gestión de colas distribuidas de IBM MQ. IBM MQ utiliza *Agentes de canal de mensajes* (MCA) para enviar mensajes de un gestor de colas a otro.

Para enviar mensajes del gestor de colas A al gestor de colas B, un MCA de *envío* en el gestor de colas A debe configurar un enlace de comunicaciones con el gestor de colas B. Se debe iniciar un MCA de *recepción* en el gestor de colas B para recibir mensajes del enlace de comunicaciones. Esta vía unidireccional formada por el MCA emisor, el enlace de comunicaciones y el MCA receptor se conoce como *canal*. El MCA emisor toma los mensajes de una cola de transmisión y los envía por un canal al MCA receptor. El MCA receptor recibe los mensajes y los coloca en las colas de destino.

En IBM MQ for z/OS, los MCM de envío y recepción se ejecutan dentro del iniciador de canal (el iniciador de canal también recibe el nombre de *desplazador*). El iniciador de canal se ejecuta como un espacio de direcciones de z/OS bajo el control del gestor de colas. Sólo puede haber un único iniciador de canal conectado a un gestor de colas y se ejecuta bajo la misma imagen de z/OS que el gestor de colas. Pueden haber miles de procesos MCA ejecutándose dentro del iniciador de canal al mismo tiempo.

Figura 57 en la página 167 muestra dos gestores de cola en un sysplex. Cada gestor de colas tiene un iniciador de canal y una cola local. Los mensajes enviados por gestores de colas en AIX y Windows se colocan en la cola local desde donde una aplicación los recupera. Los mensajes de respuesta se devuelven a través de una ruta similar.

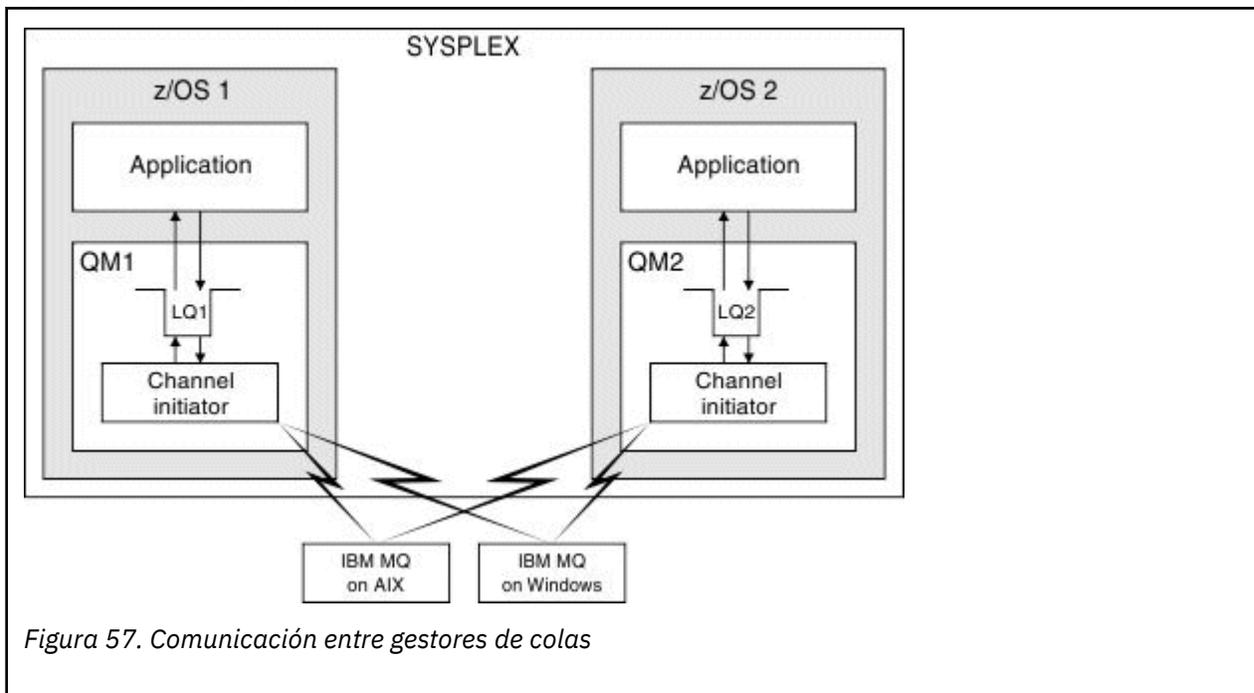


Figura 57. Comunicación entre gestores de colas

El iniciador de canal también contiene otros procesos relacionados con la gestión de los canales. Estos procesos incluyen:

#### Escuchas

Estos procesos escuchan las solicitudes de canal de entrada en un subsistema de comunicaciones como TCP, e inician un MCA con nombre cuando se recibe una solicitud de entrada.

## Supervisor

Esto gestiona el espacio de direcciones del iniciador de canal, por ejemplo, es el responsable de reiniciar los canales después de una anomalía.

## Nombre servidor

Se utiliza para resolver nombres TCP en direcciones.

## Tareas de TLS

Estas tareas se utilizan para realizar el cifrado y descifrado y comprobar las listas de revocación de certificados.

## Registros de SMF para el iniciador de canal

El iniciador de canal (CHINIT) puede generar registros de estadísticas SMF y registros de contabilidad con información sobre tareas y canales.

CHINIT puede generar producir registros de estadísticas SMF y registros de contabilidad con los siguientes tipos de información:

- Tareas: asignador, adaptador, servidor de nombres de dominio (DNS) y SSL. Estas tareas forman lo que se denominan estadísticas CHINIT.
- Canales: proporciona información de contabilidad similar a la que está disponible con el mandato DIS CHSTATUS. Esto se denomina contabilidad de canales.

IBM MQ for Multiplatforms proporciona información similar escribiendo mensajes PCF en SYSTEM.ADMIN.STATISTICS.QUEUE. Consulte [Datos de mensajes estadísticos de canal](#) para obtener más información sobre cómo se registra la información estadística en IBM MQ for Multiplatforms.

## Datos estadísticos

Puede utilizar esta información para averiguar la siguiente información:

- Si necesita más de las tareas CHINIT, como por ejemplo el número de TCB SSL y la cantidad de CPU utilizada por estas tareas.
- El tiempo promedio para las solicitudes en estas tareas.
- La solicitud con la duración más larga del intervalo y la hora del día en que ha sucedido, para tareas DNS y SSL. Puede correlacionar esta hora del día con los problemas que puede experimentar con el canal.

## Datos de contabilidad

Puede utilizar esta información para supervisar el uso del canal y averiguar la siguiente información:

- Los canales con el rendimiento más alto.
- La velocidad a la que se han enviado mensajes y la tasa de envío de datos en MB/segundo.
- El tamaño de lote conseguido. Si el tamaño de lote conseguido es cercano al tamaño de lote especificado para el canal, el canal puede estar cerca de su límite para enviar mensajes.

Los mandatos [START TRACE](#) y [STOP TRACE](#) se utilizan para controlar la recopilación del rastreo de contabilidad y el rastreo de estadísticas. Puede utilizar las opciones [STATCHL](#) y [STATACLS](#) en el canal y el gestor de colas para controlar si los canales generan datos SMF.

## Términos y tareas para gestionar IBM MQ for z/OS

Utilice este tema como introducción a la terminología y tareas que son específicas de IBM MQ for z/OS.

Algunos de los términos y tareas necesarios para gestionar IBM MQ for z/OS son específicos de la plataforma. z/OS La lista siguiente contiene algunos de estos términos y estas tareas.

- [Colas compartidas](#)
- [Conjuntos de páginas y agrupaciones de almacenamiento intermedio](#)
- [Registro](#)

- [Adaptación del entono del gestor de colas](#)
- [Reinicio y recuperación](#)
- [Seguridad](#)
- [Disponibilidad](#)
- [Manipulación de objetos](#)
- [Supervisión y estadísticas](#)
- [Entornos de aplicación](#)

## Colas compartidas

Las colas pueden ser *no compartidas*, propiedad de un único gestor de colas que también es el único que puede acceder a las mismas o pueden ser *compartidas*, propiedad de un *grupo de compartición de colas*. Un grupo de compartición de colas consta de un número de gestores de colas, que se ejecutan en un único sysplex de z/OS, que pueden acceder a las mismas definiciones de objeto de IBM MQ y datos de mensajes simultáneamente. Dentro de un grupo de compartición de colas, las definiciones de objetos compartibles se almacenan dentro de una base de datos de Db2. Los mensajes de colas compartidas se almacenan dentro de una o varias estructuras de recurso de acoplamiento (estructuras CF). Si los datos del mensaje son demasiado grandes como para almacenarse directamente en la estructura (un tamaño de más de 63 KB) o bien o si el mensaje es suficientemente grande como para que las reglas definidas por la instalación lo seleccionen para la descarga, la información de control del mensaje sigue almacenándose en la entrada del recurso de acoplamiento, pero los datos del mensaje se descargan en un conjunto de datos de mensaje compartido (SMDS) o en una base de datos de Db2 compartida. Los conjuntos de datos de mensajes compartidos, la base de datos de Db2 compartida y las estructuras de recurso de acoplamiento son recursos que todos los gestores de colas del grupo gestionan conjuntamente.

## Conjuntos de páginas y agrupaciones de almacenamiento intermedio

Cuando se coloca un mensaje a una cola no compartida, el gestor de colas almacena los datos en un conjunto de páginas de tal forma que pueda recuperarse cuando una operación posterior obtenga un mensaje de la misma cola. Si el mensaje se elimina de la cola, el espacio en el conjunto de páginas que contiene los datos se libera posteriormente para poder volver a ser utilizado. A medida que aumenta el número de mensajes retenidos en una cola, de modo que la cantidad de espacio utilizada en el conjunto de páginas aumenta y se reduce el número de mensajes en una cola, se reduce el espacio utilizado en el conjunto de páginas.

Para reducir el costo de rendimiento de la escritura de datos y la lectura de datos de los conjuntos de la página, el gestor de colas coloca las actualizaciones, como si se tratara de un almacenamiento intermedio, en el almacenamiento del procesador. La cantidad de almacenamiento que se utiliza para el almacenamiento intermedio del acceso del conjunto de páginas se controla a través de objetos IBM MQ denominados *agrupaciones de almacenamiento intermedio*.

Para obtener más información sobre los conjuntos de páginas y las agrupaciones de almacenamientos intermedios, consulte [de Gestión de almacenamiento](#).

## Registro

Los cambios en los objetos conservados en conjuntos de páginas y en operaciones en mensajes permanentes se registran como registros de anotaciones. Estos registros de anotaciones se escriben en un conjunto de datos del registro denominado *registro activo*. El nombre y el tamaño del conjunto de datos de registro activo están incluidos en un conjunto de datos denominado *conjunto de datos de arranque* (BSDS).

Cuando se llena el conjunto de datos de registro activo, el gestor de colas cambia a otro conjunto de datos de registro para que el registro cronológico pueda continuar y copia el contenido del conjunto de datos de registro activo lleno en un conjunto de datos de *registro de archivado*. Encontrará más información sobre estas acciones, incluyendo el nombre del conjunto de datos de registro de archivado, en el conjunto de datos de arranque. Conceptualmente, hay un anillo de conjuntos de datos de registro activo por el que circula el gestor de colas; cuando se llena un registro activo, los datos de registro se descargan en un registro de archivado y el conjunto de datos de registro activo vuelve a estar disponible para su reutilización.

Para obtener más información sobre los conjuntos de datos del registro y de arranque, consulte [“Registro en IBM MQ for z/OS”](#) en la página 238.

## **Adaptación del entorno del gestor de colas**

Cuando se inicia el gestor de colas, se lee un conjunto de parámetros de inicialización que controlan cómo debe funcionar el gestor de colas. Además, se leen los conjuntos de datos que contienen mandatos de IBM MQ y se ejecutan los mandatos que contienen. Normalmente, estos conjuntos de datos contienen definiciones de los objetos del sistema necesarios para que IBM MQ se ejecute, y puede personalizarlos para definir o inicializar los objetos de IBM MQ necesarios para su entorno operativo. Cuando estos conjuntos de datos se han leído, se almacenan los objetos definidos por ellos, en un conjunto de páginas o en un sistema Db2.

Para obtener más información sobre los parámetros de inicialización y objetos del sistema, consulte [“Definición de sistema en z/OS”](#) en la página 250.

## **Recuperación y reinicio**

En cualquier momento durante la operación de IBM MQ, es posible que se realicen cambios en el almacenamiento del procesador que todavía no se han grabado en el conjunto de páginas. Estos cambios se escriben en el conjunto de páginas que es el que una tarea en segundo plano ha utilizado menos recientemente en el gestor de colas.

Si el gestor de colas termina de forma anómala, la fase de recuperación del reinicio del gestor de colas puede recuperar los cambios del conjunto de páginas perdido pues los datos del mensaje permanente se conservan en registros de anotaciones. Esto significa que IBM MQ puede recuperar los datos de mensajes persistentes y los cambios de objeto hasta el punto de error.

Si un gestor de colas que es miembro de un grupo de compartición de colas encuentra una anomalía en el recurso de acoplamiento, los mensajes permanentes en dicha cola sólo se pueden recuperar si ha realizado una copia de seguridad de la estructura de recurso de acoplamiento.

Para obtener más información sobre la recuperación y el reinicio, consulte [“Recuperación y reinicio en z/OS”](#) en la página 261.

## **Seguridad**

Puede utilizar un gestor de seguridad externa como, por ejemplo, Security Server (anteriormente conocido como RACF) para proteger los recursos que IBM MQ posee y gestiona del acceso de usuarios no autorizados. También puede utilizar TLS (Transport Layer Security) para la seguridad del canal. TLS se incluye como parte del producto IBM MQ.

Para obtener más información sobre la seguridad de IBM MQ, consulte [“Conceptos de seguridad en IBM MQ for z/OS”](#) en la página 279.

## **Disponibilidad**

Hay varias características de IBM MQ que están diseñadas para aumentar la disponibilidad del sistema en caso de que se produzca un error en el gestor de colas o en el subsistema de comunicaciones. Para obtener más información sobre estas características, consulte [“Disponibilidad en z/OS”](#) en la página 286.

## Manipulación de objetos

Cuando el gestor de colas se está ejecutando, puede manipular objetos IBM MQ a través de una interfaz de consola de z/OS o mediante un programa de utilidad de administración que utiliza servicios ISPF bajo TSO. Ambos mecanismos permiten definir, alterar o suprimir objetos de IBM MQ. También puede controlar y visualizar el estado de varias funciones del gestor de colas y de IBM MQ.

Para obtener más información sobre estos recursos, consulte [Orígenes desde los que puede emitir mandatos MQSC y PCF en IBM MQ for z/OS](#).

También puede manipular objetos de IBM MQ utilizando IBM MQ Explorer, una interfaz gráfica de usuario que proporciona una forma visual de trabajar con colas, gestores de colas y otros objetos.

## Supervisión y estadísticas

Hay varios recursos disponibles para supervisar los gestores de colas y los iniciadores de canal. También puede recopilar estadísticas para la evaluación de rendimiento y con objetivos de contabilidad.

Para obtener más información sobre estos recursos, consulte [“Supervisión y estadísticas en IBM MQ for z/OS”](#) en la página 289.

## Entornos de aplicación

Cuando el gestor de colas se ha iniciado, las aplicaciones pueden conectarse a él y empezar a utilizar la API de IBM MQ. Pueden ser aplicaciones CICS, IMS, por lotes o bien WebSphere Application Server. Las aplicaciones de IBM MQ también pueden acceder a las aplicaciones en sistemas CICS e IMS que no sepan nada acerca de IBM MQ, utilizando los puentes de CICS e IMS.

Para obtener más información sobre estos recursos, consulte [“IBM MQ y otros productos de z/OS”](#) en la página 293.

Para obtener información sobre cómo escribir aplicaciones IBM MQ, consulte la documentación siguiente:

- [Desarrollo de aplicaciones](#)
- [Utilización de C++](#)
- [Utilización de IBM MQ classes for Java](#)

z/OS

## Colas compartidas y grupos de compartición de colas

Puede utilizar colas compartidas y grupos de compartición de colas para implementar la alta disponibilidad de recursos de IBM MQ. Las colas compartidas y los grupos de compartición de colas son funciones exclusivas de IBM MQ for z/OS en la plataforma z/OS.

Esta sección describe los atributos y ventajas y ofrece información acerca de cómo varios gestores de colas pueden compartir las mismas colas y los mensajes en esas colas.

### ¿Qué es una cola compartida?

Una cola compartida es un tipo de cola local. Uno o varios gestores de colas que estén en un sysplex pueden acceder a los mensajes de esa cola.

### Un grupo de compartición de colas

Los gestores de colas que pueden acceder al mismo conjunto de colas compartidas forman un grupo denominado *grupo de compartición de colas*.

### Cualquier gestor de colas puede acceder a los mensajes

Cualquier gestor de colas del grupo de compartición de colas puede acceder a una cola compartida. Esto significa que puede transferir un mensaje a una cola compartida en un gestor de colas y obtener el

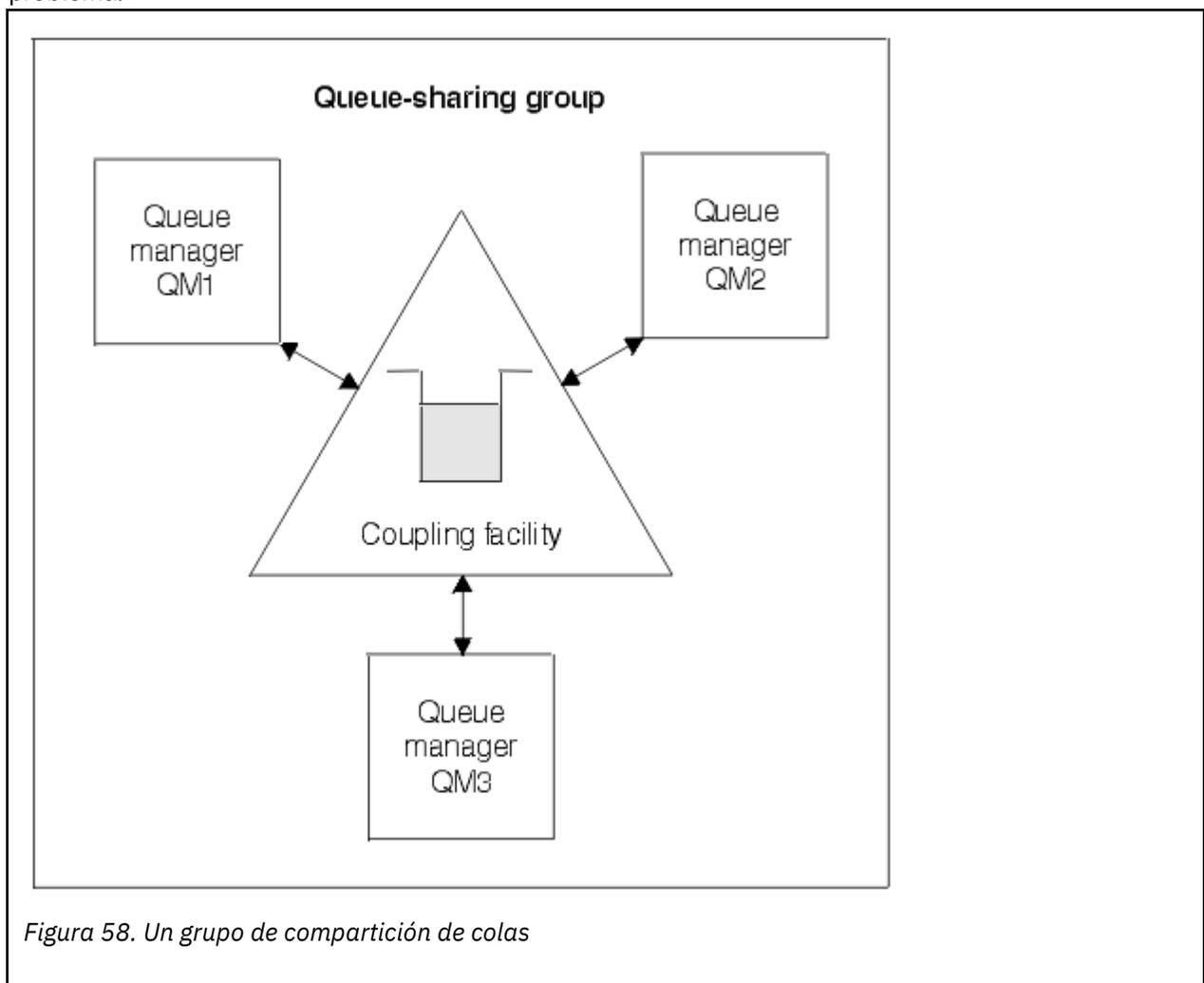
mismo mensaje de la cola de un gestor de colas diferente. Esto proporciona un mecanismo rápido para la comunicación dentro de un grupo de compartición de colas que no requiere que haya canales activos entre los gestores de colas.

IBM MQ da soporte a la descarga de mensajes en Db2 o en un conjunto de datos de mensajes compartidos (SMDS). La descarga de mensajes de cualquier tamaño es configurable.

En la [Figura 58 en la página 172](#) se muestran tres gestores de colas y un recurso de acoplamiento, que forman un grupo de compartición de colas. Los tres gestores de colas pueden acceder a la cola compartida en el recurso de acoplamiento.

Una aplicación se puede conectar a cualquiera de los gestores de colas dentro del grupo de compartición de colas. Debido a que todos los gestores de colas del grupo de compartición de colas pueden acceder a todas las colas compartidas, la aplicación no depende de la disponibilidad de un gestor de colas específico; cualquier gestor de colas del grupo de compartición de colas puede prestar servicio en la cola.

Esto ofrece una mayor disponibilidad, ya que todos los demás gestores de colas del grupo de compartición de colas pueden seguir procesando la cola si uno de los gestores de colas tiene un problema.



*Figura 58. Un grupo de compartición de colas*

### **La definición de cola es compartida por todos los gestores de colas**

Las definiciones de colas compartidas se almacenan en la tabla de base de datos OBJ\_B\_QUEUE de Db2. Por ello, debe definir la cola una sola vez y luego pueden acceder a ella todos los gestores de colas del grupo de compartición de colas. Esto significa que hay menos definiciones que hacer.

Por el contrario, la definición de una cola no compartida se almacena en el conjunto de páginas cero del gestor de colas que posee la cola (como se describe en [Conjuntos de páginas](#)).

No puede definir una cola compartida si ya se ha definido una cola con ese nombre en los conjuntos de páginas del gestor de colas de definición. Del mismo modo, no puede definir una versión local de una cola en los conjuntos de páginas del gestor de colas si ya existe una cola compartida con el mismo nombre.

## ¿Qué es un grupo de compartición de colas?

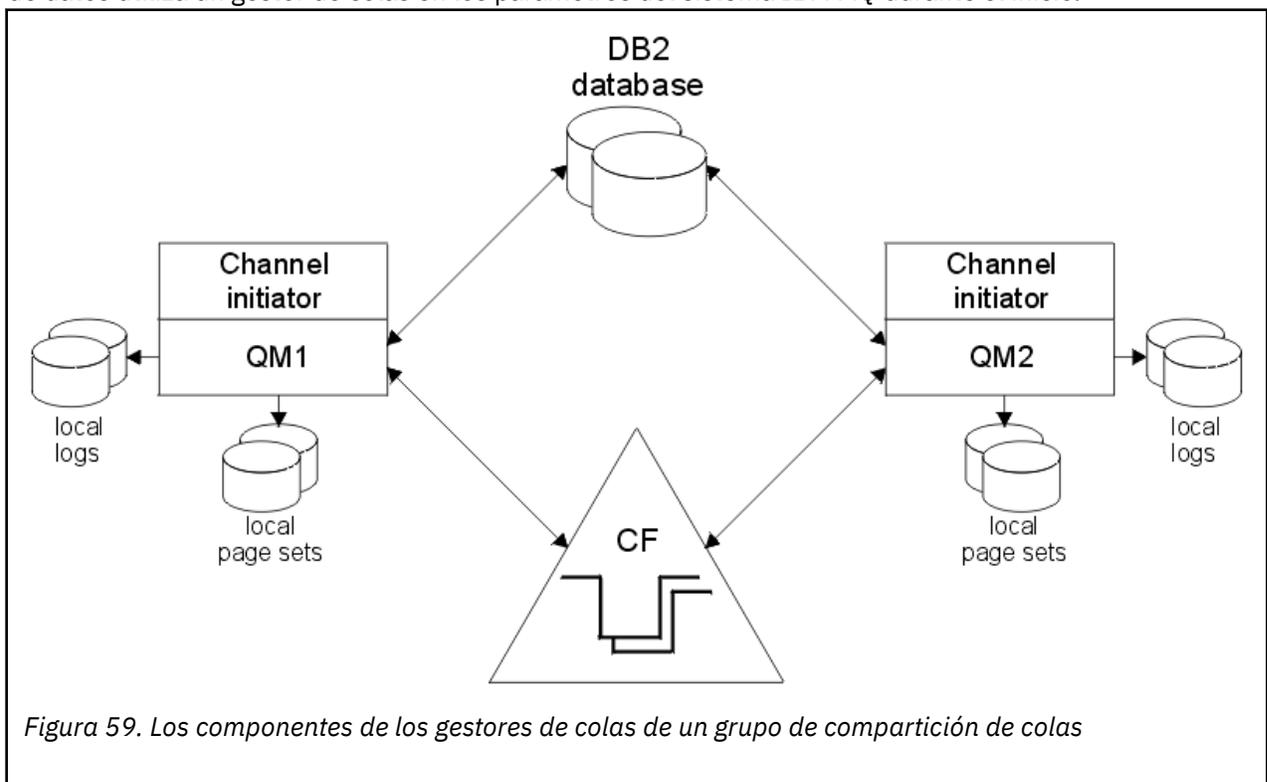
Un grupo de gestores de colas que pueden acceder a las mismas colas compartidas se denomina grupo de compartición de colas. Todos los miembros del grupo de compartición de colas tienen acceso al mismo conjunto de colas compartidas.

Los grupos de compartición de colas tienen un nombre de hasta cuatro caracteres. El nombre debe ser exclusivo en la red y debe ser diferente de los nombres de gestor de colas.

En la [Figura 59](#) en la [página 173](#) se ilustra un grupo de compartición de colas que contiene dos gestores de colas. Cada gestor de colas tiene un iniciador de canal y sus propios conjuntos de páginas locales y conjuntos de datos de registro.

Cada miembro del grupo de compartición de colas también debe conectarse a un sistema Db2. Los sistemas Db2 deben estar todos en el mismo grupo de compartición de datos de Db2 de modo que los gestores de colas puedan acceder al repositorio compartido de Db2 para guardar las definiciones de objetos compartidos. Son definiciones de cualquier tipo de objeto de IBM MQ (por ejemplo, colas y canales) que se definen una sola vez y, a continuación, cualquier gestor de colas del grupo puede utilizarlas. Estas se conocen como definiciones *globales* y se describen en [Definiciones globales y privadas](#).

Más de un grupo de compartición de colas puede hacer referencia a un determinado grupo de compartición de datos. Debe especificar el nombre del subsistema Db2 y qué grupo de compartición de datos utiliza un gestor de colas en los parámetros del sistema IBM MQ durante el inicio.



*Figura 59. Los componentes de los gestores de colas de un grupo de compartición de colas*

Cuando un gestor de colas se ha unido a un grupo de compartición de colas, tiene acceso a los objetos compartidos definidos para dicho grupo, y puede utilizar ese gestor de colas para definir nuevos objetos compartidos dentro del grupo. Si se definen colas compartidas dentro del grupo, puede utilizar este

gestor de colas para transferir mensajes a esas colas compartidas y obtener mensajes de dichas colas. Cualquier gestor de colas del grupo puede recuperar los mensajes contenidos en una cola compartida.

Puede especificar un mandato MQSC una vez y hacer que se ejecute en todos los gestores de colas dentro del grupo de compartición de colas como si se hubiera especificado en cada gestor de colas individualmente. Para ello se utiliza el atributo *ámbito de mandato*. Este atributo se describe en [Dirección de mandatos a distintos gestores de colas](#).

Cuando un gestor de colas se ejecuta como miembro de un grupo de compartición de colas, debe ser posible distinguir entre los objetos de IBM MQ definidos de forma privada en ese gestor de colas y los objetos de IBM MQ definidos globalmente que están disponibles para todos los gestores de colas del grupo de compartición de colas. Para ello se utiliza el atributo *disposición del grupo de compartición de colas*. Este atributo se describe en [Definiciones globales y privadas](#).

Puede definir un único conjunto de perfiles de seguridad que controlen el acceso a objetos de IBM MQ en cualquier lugar del grupo. Esto significa que el número de perfiles que tiene que definir se reduce considerablemente.

Un gestor de colas puede pertenecer a un solo grupo de compartición de colas, y todos los gestores de colas del grupo deben estar en el mismo sysplex. Debe especificar a qué grupo de compartición de colas pertenece el gestor de colas en los parámetros del sistema durante el inicio.

### Conceptos relacionados

[“¿Dónde se almacenan los mensajes de colas compartidas?” en la página 174](#)

Cada mensaje de una cola compartida se representa mediante una entrada en una estructura de lista del recurso de acoplamiento de z/OS. Si los datos del mensaje son demasiados y no caben en la misma entrada, se descargan en un conjunto de datos de mensaje compartido (SMDS) o bien en Db2.

[“Ventajas de la utilización de colas compartidas” en la página 192](#)

La cola compartida permite que las aplicaciones IBM MQ sean escalables, altamente disponibles y permite implementar el equilibrio de carga de trabajo.

[“Gestión de colas distribuidas y grupos de compartición de colas” en la página 213](#)

La gestión de colas distribuidas y los grupos de compartición de colas son dos técnicas que puede utilizar para aumentar la disponibilidad de los sistemas de aplicación. Utilice este tema para encontrar más información sobre estas técnicas.

[“Cómo influir en la distribución de la carga de trabajo con colas compartidas” en la página 217](#)

Utilice este tema si desea conocer los factores que afectan la distribución de la carga de trabajo con colas compartidas en un grupo de compartición de colas.

### Referencia relacionada

[“Dónde encontrar más información sobre colas compartidas y grupos de compartición de colas” en la página 218](#)

Utilice la tabla de este tema para obtener más información sobre cómo IBM MQ for z/OS utiliza colas compartidas y grupos de compartición de colas.

## ¿Dónde se almacenan los mensajes de colas compartidas?

Cada mensaje de una cola compartida se representa mediante una entrada en una estructura de lista del recurso de acoplamiento de z/OS. Si los datos del mensaje son demasiados y no caben en la misma entrada, se descargan en un conjunto de datos de mensaje compartido (SMDS) o bien en Db2.

Si la estructura CF se ha configurado para utilizar la memoria de clase del sistema (SCM), IBM MQ puede utilizarla sin ninguna configuración adicional.

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z<sup>®</sup> para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

## Almacenamiento de mensajes de colas compartidas

Los mensajes que se colocan en colas compartidas no se almacenan en conjuntos de páginas y no utilizan agrupaciones de almacenamiento intermedio.

Los mensajes de las colas compartidas tienen entradas en las estructuras de lista del recurso de acoplamiento (CF) de z/OS. Muchos gestores de colas en el mismo sysplex pueden acceder a esos mensajes utilizando la estructura de lista de CF.

Los datos de mensajes pequeños de colas compartidas normalmente se incluyen en la entrada del recurso de acoplamiento. Para mensajes de mayor volumen, los datos de mensajes se almacenan en un conjunto de datos de mensajes compartidos (SMDS), o como uno o varios objetos grandes binarios (BLOB) en una tabla de Db2 que está compartida por un grupo de compartición de datos de Db2. Los datos de mensajes que exceden de 63 KB siempre se descargan en SMDS o en Db2. Los mensajes más pequeños también pueden descargarse, de manera opcional, de la misma manera para ahorrar espacio en el estructura de recurso de acoplamiento. Consulte [“Especificación de opciones de descarga para mensajes compartidos”](#) en la página 177 para obtener más detalles.

Los mensajes colocados en una cola compartida son referenciados en una estructura de recurso de acoplamiento hasta que se recuperan mediante una llamada MQGET. Las operaciones del recurso de acoplamiento se utilizan para:

- Buscar el siguiente mensaje recuperable
- Bloquear mensajes sin confirmar en colas compartidas
- Notificar a los gestores de colas interesados de la llegada de mensajes confirmados

Las operaciones MQPUT y MQGET en mensajes permanentes se graban en el registro del gestor de colas que realiza dicha operación. Esto minimiza el riesgo de pérdida de datos en el caso de una anomalía del recurso de acoplamiento.

## El recurso de acoplamiento

Los mensajes almacenados en colas compartidas son referenciados dentro de un recurso de acoplamiento. El recurso de acoplamiento reside fuera de cualquiera de las imágenes de z/OS en el sysplex y suele configurarse para ejecutarse en una fuente de alimentación diferente. El recurso de acoplamiento es, por consiguiente, resistente a anomalías de software y puede configurarlo para que sea resistente a anomalías de hardware o cortes de suministro. Esto significa que los mensajes almacenados en el recurso de acoplamiento son de alta disponibilidad.

Cada estructura de lista de recurso de acoplamiento utilizada por IBM MQ está dedicada a un grupo de compartición de colas específico, pero un recurso de acoplamiento puede contener estructuras para más de un grupo de compartición de colas. Los gestores de colas de grupos de compartición de colas diferentes no pueden compartir datos. Se pueden conectar hasta 32 gestores de colas de un grupo de compartición de colas a una estructura de lista de recurso de acoplamiento al mismo tiempo.

Una sola estructura de lista de recurso de acoplamiento puede contener hasta 512 colas compartidas. La cantidad total de datos de mensaje almacenados en la estructura está limitada por la capacidad de la estructura. Sin embargo, con **CFLEVEL (5)** puede utilizar los parámetros de descarga para descargar datos para mensajes de menos de 63 KB, aumentando así el número de mensajes que se pueden almacenar en la estructura, aunque cada mensaje todavía requiere al menos una entrada de recurso de acoplamiento más al menos 768 bytes de datos, que se componen de 256 bytes para la entrada y 512 bytes para los dos elementos de cabecera y descriptor.

El tamaño de la estructura de lista está limitado por los siguientes factores:

- Debe estar dentro de un solo recurso de acoplamiento.
- Puede compartir el almacenamiento de recurso de acoplamiento disponible con otras estructuras para IBM MQ y otros productos.

Las estructuras de lista del recurso de acoplamiento pueden tener asociada una memoria de clase de almacenamiento. En determinados casos, esta memoria de clase de almacenamiento puede ser útil cuando se utiliza con colas compartidas. Consulte [“Uso de la memoria de clase de almacenamiento \(SCM\) con colas compartidas”](#) en la página 194 para obtener más información.

## Planificar el tamaño de la estructura de CF

Si necesita información sobre el tamaño de las estructuras de CF, puede utilizar [MP16: IBM MQ for z/OS Planificación y ajuste de la capacidad supportpac](#). También puede utilizar la herramienta [CFSizer](#), que IBM proporciona para ayudar con los tamaños de CF.

## El objeto de estructura de CF

El uso del gestor de colas de una estructura de recurso de acoplamiento se especifica en un objeto de estructura de CF (CFSTRUCT) IBM MQ.

Estos objetos de estructura se almacenan en Db2.

Cuando se utilizan mandatos o definiciones de z/OS relativos a una estructura del recurso de acoplamiento, los cuatro primeros caracteres del nombre del grupo de compartición de colas son obligatorios. Sin embargo, siempre existe un objeto CFSTRUCT de IBM MQ dentro de un único grupo de compartición de colas, por lo que su nombre no incluye los cuatro primeros caracteres del nombre del grupo de compartición de colas. Por ejemplo, CFSTRUCT(MYDATA) definido en el grupo de compartición de colas que empieza por SQ03 utilizaría la estructura de lista de recurso de acoplamiento SQ03MYDATA.

Las estructuras de CF tienen un atributo CFLEVEL que determina su capacidad funcional:

- 1, 2 - se puede utilizar para mensajes no persistentes de menos de 63 KB
- 3 - se puede utilizar para mensajes persistentes y no persistentes de menos de 63 KB
- 4 - se puede utilizar para mensajes persistentes y no persistentes de hasta 100 MB
- 5 - se puede utilizar para los mensajes permanentes y no permanentes de hasta 100 MB y se puede descargar de forma selectiva en los conjuntos de datos de mensajes compartidos (SMDS) o en Db2.

**Nota:** Al utilizar IBM MQ, puede cifrar una estructura de recurso de acoplamiento. Consulte [Cifrado de datos de estructura de recurso de asociación](#) para obtener más información.

## Copia de seguridad y recuperación del recurso de acoplamiento

Puede realizar una copia de seguridad de las estructuras de lista de recurso de acoplamiento utilizando el mandato de IBM MQ [BACKUP CFSTRUCT](#). Se coloca una copia de los mensajes permanentes que actualmente están en la estructura de CF en el conjunto de datos de registro activo que forma la copia de seguridad y se graba un registro de la copia de seguridad en Db2.

Si el recurso de acoplamiento falla, puede utilizar el mandato de IBM MQ [RECOVER CFSTRUCT](#). Este mandato utiliza el registro de copia de seguridad de Db2 para localizar y restaurar los mensajes permanentes de la copia de seguridad de la estructura CF. Cualquier actividad realizada desde la última copia de seguridad se reproduce utilizando los registros de todos los gestores de colas del grupo de compartición de colas, y la estructura de CF se restaura entonces hasta el momento antes de la anomalía.

Consulte los mandatos [BACKUP CFSTRUCT](#) y [RECOVER CFSTRUCT](#) para obtener más información.

### Conceptos relacionados

[“Especificación de opciones de descarga para mensajes compartidos”](#) en la página 177

Puede elegir dónde almacenar los datos de mensajes para un mensaje de colas compartidas, en una tabla de Db2 o bien un conjunto de datos de mensaje compartido (SMDS). También puede seleccionar qué mensajes deben descargarse basándose en el tamaño del mensaje y el uso actual de la estructura del recurso de acoplamiento (CF).

[“Gestión del entorno de conjunto de datos de mensaje compartido \(SMDS\)”](#) en la página 179

Si selecciona conjuntos de datos de mensajes compartidos para descargar mensajes grandes, también debe tener en cuenta la información que IBM MQ utiliza para gestionar estos conjuntos de datos y los mandatos que se utilizan para trabajar con esta información. Utilice este tema para aprender a gestionar los conjuntos de datos de mensaje compartidos.

## **z/OS** **Especificación de opciones de descarga para mensajes compartidos**

Puede elegir dónde almacenar los datos de mensajes para un mensaje de colas compartidas, en una tabla de Db2 o bien un conjunto de datos de mensaje compartido (SMDS). También puede seleccionar qué mensajes deben descargarse basándose en el tamaño del mensaje y el uso actual de la estructura del recurso de acoplamiento (CF).

Los datos de mensaje para colas compartidas se pueden descargar desde el recurso de acoplamiento y almacenarse en una tabla de Db2 o en un conjunto de datos gestionado por IBM MQ denominado *conjunto de datos de mensaje compartido* (SMDS).

En el caso de los mensajes que superan el tamaño de entrada del recurso de acoplamiento de 63 KB, la descarga de datos de mensaje en un SMDS puede suponer una mejora de rendimiento significativa en comparación con la descarga en Db2.

Cada mensaje de cola compartida todavía se gestiona utilizando una entrada de lista en una estructura de recurso de acoplamiento, pero cuando los datos de mensaje se descargan en el conjunto de datos de mensaje compartido, la entrada del recurso de acoplamiento sólo contiene parte de la información de control y una lista de referencias a los bloques de disco relevantes en los que se almacena el mensaje. La utilización de este mecanismo significa que la cantidad de almacenamiento del elemento del recurso de acoplamiento de cada mensaje es sólo una fracción del tamaño real del mensaje.

### **Selección del lugar en el que se almacenan los mensajes de cola compartida**

La selección del almacenamiento de mensaje compartido Db2 o SMDS se controla con el parámetro **OFFLOAD(SMDS|DB2)** en la definición **CFSTRUCT**. **OFFLOAD(SMDS)** es el valor predeterminado.

Este parámetro también requiere que **CFSTRUCT** utilice **CFLEVEL(5)** o un valor mayor.

El parámetro **OFFLOAD** solamente es válido desde **CFLEVEL(5)**. Consulte [DEFINE CFSTRUCT](#) para obtener más información.

**OFFLOAD(DB2)** está soportado principalmente para la migración.

### **Selección de los mensajes de cola compartida que se descargan**

Los datos del mensaje se descargan en SMDS o en Db2 basándose en el tamaño de los datos de mensaje y el uso actual de la estructura del recurso de acoplamiento. Existen tres reglas y cada una de ellas especifica un par de parámetros coincidente. Estos parámetros consisten en un porcentaje del umbral de uso de estructura del recurso de acoplamiento correspondiente (**OFFLDnTH**) y un límite de tamaño de mensaje (**OFFLDnSZ**).

La implementación actual de las tres reglas se especifica utilizando los siguientes pares de palabras clave:

- OFFLD1TH y OFFLD1SZ
- OFFLD2TH y OFFLD2SZ
- OFFLD3TH y OFFLD3SZ

<b>Par de reglas</b>	<b>Valor predeterminado</b>	<b>Descripción</b>
Par de reglas 1	OFFLD1TH(70) y OFFLD1SZ(32K)	Si la estructura del recurso de acoplamiento está llena en más de un 70%, descargue los datos de los mensajes que superen los 32KB

Par de reglas	Valor predeterminado	Descripción
Par de reglas 2	OFFLD2TH(80) y OFFLD2SZ(4K)	Si la estructura del recurso de acoplamiento está llena en más de un 80%, descargue los datos de los mensajes que superen los 4 KB
Par de reglas 3	OFFLD3TH(90) y OFFLD3SZ(0K)	Si la estructura del recurso de acoplamiento está lleno en más de un 90%, descargue los datos de los mensajes que superen los 0 KB (todos los mensajes)

Si una regla de descarga tiene el valor OFFLD x SZ de 64K, esto indica que la regla no está en vigor. En este caso, los mensajes sólo se descargarán si otra regla de descarga está en vigor o si el mensaje tiene más de 63,75 KB y por lo tanto, es demasiado grande para almacenarse en la estructura.

Cada mensaje que se descarga sigue requiriendo 0,75 KB de almacenamiento en el recurso de acoplamiento.

Las tres reglas de descarga que se pueden especificar para cada estructura están pensadas para ser utilizadas como se indica a continuación.

- Rendimiento

- Si hay mucho espacio en la estructura de la aplicación, los datos de mensajes sólo deberían descargarse si son demasiado grandes para almacenarse en la estructura o si superan el umbral inferior del tamaño de un mensaje de tal manera que el valor de rendimiento para dicho almacenamiento en la estructura no compensa en relación con el tamaño del espacio de la estructura que necesitaría.
- Si es necesario un umbral de tamaño de mensaje específico, se especifica normalmente mediante la primera regla de descarga.

- Capacidad

- Cuando hay muy poco espacio en la estructura de la aplicación, debe descargarse la máxima cantidad de datos de mensaje debe ser de máximo de manera que se haga el mejor uso del espacio restante.
- La tercera regla de descarga se suele utilizar normalmente para indicar que la estructura está casi llena; la mayoría de los mensajes deben descargarse, de modo que las entradas en la estructura de la aplicación deberían tener un tamaño mínimo (sobre 0,75K bytes).
- El parámetro de umbral de uso debe elegirse en función del tamaño de la estructura de la aplicación y la reserva máxima anticipada. Por ejemplo, si la reserva máxima anticipada es de 1M mensajes, el volumen de almacenamiento de la estructura para este número de mensajes es aproximadamente 0,75G. Esto significa, por ejemplo, que si la estructura es de aproximadamente 10G bytes, el umbral de uso para la descarga de todos los mensajes debe establecerse en 92% o inferior.
- El espacio de la estructura se divide en elementos y entradas, y aunque en conjunto haya espacio suficiente, uno de estos espacios podría agotar el espacio antes que el otro. El sistema proporciona funciones AUTOALTER para ajustar la proporción cuando sea necesario, pero esto no es muy preciso, ya que la cantidad de espacio realmente disponible puede ser algo menor. Se recomienda no utilizar más del 90% del espacio máximo de estructura. Así, en el ejemplo anterior, lo ideal sería establecer el umbral de uso de descarga de todos los mensajes aproximadamente en el 80%.

- Transición protegida:

- A medida que desciende la cantidad de espacio que queda en la estructura del recurso de acoplamiento, no sería deseable que las características de rendimiento cambiaran de repente. Tampoco sería conveniente para la gestión del recurso de acoplamiento que hubiera un cambio

de umbral repentino en cuanto a la proporción normal de entradas de los elementos que se utilizan.

- La segunda regla de descarga se suele utilizar para proporcionar una amortiguación intermedia entre las reglas de rendimiento y las reglas de descarga que favorecen a la capacidad. Puede definirse para aumentar significativamente la actividad de descarga cuando el espacio utilizado en la estructura del recurso de acoplamiento excede un umbral intermedio. Esto significa que el espacio restante se utiliza más lentamente, y proporciona al recurso de acoplamiento un proceso automático de modificación con más tiempo para adaptarse a los niveles de uso superiores.

Si la estructura de recurso de acoplamiento no puede expandirse y existe la necesidad de almacenar al menos un determinado número de mensajes, la tercera regla puede modificarse según sea necesario para asegurarse de que la descarga de datos para todos los mensajes empiece en un umbral adecuado para garantizar que se reserve el espacio para dicho número predeterminado de mensajes.

Por ejemplo, si el tamaño de la estructura del recurso de acoplamiento es de 4 GB y el número predeterminado de mensajes es de 1 millón, entonces se necesitarán  $1.000.000 * 0,75$  KB, es decir 768 MB, 18,75% de 4 GB. En este caso, el umbral para descargar todos los mensajes necesita estar establecido en 80% en lugar de en 90%. Ello es lo que proporcionan los parámetros OFFLD3TH(80) y OFFLD3SZ(0K). Los otros parámetros de descarga también tendrán que ajustarse.

Si se observa que la descarga de mensajes muy pequeños tiene un impacto significativo en el rendimiento, pero el impacto relativo es menor para los mensajes más grandes, los umbrales de uso de las demás reglas puede reducirse para descargar más mensajes anteriores, dejando más espacio en la estructura para mensajes pequeños antes de que necesiten ser descargados.

Por ejemplo, si los mensajes que exceden los 32KB se producen con frecuencia, pero el rendimiento del tiempo transcurrido para descargarlos (determinado a partir de estadísticas RMF o del rendimiento de la aplicación) es muy similar al de mantenerlos en el recurso de acoplamiento, entonces el umbral para la primera regla podría establecerse en 0% para descargar todos estos mensajes. Esto es lo que proporcionan los parámetros OFFLD1TH(0) y OFFLD1SZ(32K). De nuevo, tendrán que ajustarse los otros parámetros de descarga.

Si hay muchos mensajes en torno a tamaños intermedios específicos, tales como 16 KB y 6 KB, puede resultar útil cambiar la opción de tamaño de mensaje para la segunda regla de modo que los más grandes se descarguen en un umbral de utilización bastante bajo (ahorrando una cantidad significativa de espacio) pero que los pequeños sigan almacenándose únicamente en el recurso de acoplamiento.

## **Gestión del entorno de conjunto de datos de mensaje compartido (SMDS)**

Si selecciona conjuntos de datos de mensajes compartidos para descargar mensajes grandes, también debe tener en cuenta la información que IBM MQ utiliza para gestionar estos conjuntos de datos y los mandatos que se utilizan para trabajar con esta información. Utilice este tema para aprender a gestionar los conjuntos de datos de mensaje compartidos.

### **Objetos SMDS**

Se realiza un seguimiento de las propiedades y el estado de cada conjunto de datos de mensaje compartido en un objeto de conjunto de datos de mensaje compartido (SMDS) que se puede actualizar a través de cualquier gestor de colas en el grupo de compartición de colas.

Hay un conjunto de datos de mensaje compartido para cada gestor de colas que puede acceder a cada estructura de aplicación de recurso de acoplamiento. El conjunto de datos de mensaje compartido se identifica mediante el nombre del gestor de colas propietario (especificado utilizando la palabra clave SMDS) y por el nombre de estructura de aplicación (especificado utilizando la palabra clave CFSTRUCT).

**Nota:** Al definir conjuntos de datos SMDS para una estructura, debe tener uno para cada gestor de colas.

El objeto SMDS se almacena en una matriz (con una entrada por cada gestor de colas del grupo) que forma una extensión del correspondiente objeto CFSTRUCT almacenado en Db2.

No hay ningún mandato para DEFINE (definir) ni DELETE (suprimir) el objeto SMDS porque se crea o se suprime como parte del objeto CFSTRUCT, pero hay un mandato para ALTER (modificar) dicho objeto para cambiar valores de una persona que posee el gestor de colas.

Para obtener más información sobre los mandatos SMDS, consulte [“Mandatos relacionados con SMDS”](#) en la [página 192](#)

## Información SMDSCONN

Es posible que un conjunto de datos de mensaje compartido esté en estado normal, pero uno o varios gestores de colas pueden no poder conectarse al mismo, por ejemplo, debido a un problema con una definición de seguridad o con la conectividad del dispositivo de acceso directo. Por lo tanto, es necesario que cada gestor de colas realice un seguimiento del estado de conexión y de la información de disponibilidad de cada conjunto de datos de mensaje compartido, mientras indica, por ejemplo, si actualmente se puede conectar y, si no es posible, por qué no.

La información SMDSCONN representa una conexión de un gestor de colas con un conjunto de datos de mensaje compartido. En cuanto al propio conjunto de datos de mensaje compartido, se identifica mediante el gestor de colas que posee el conjunto de datos de mensaje compartido (tal como se especifica en la palabra clave SMDS para el objeto compartido propiamente dicho) combinado con el nombre CFSTRUCT.

No existe ningún parámetro para identificar al gestor de colas que se conecta porque los mandatos dirigidos a un gestor de colas específico, sólo pueden hacer referencia a información SMDSCONN del mismo gestor de colas.

Las entradas de información SMDSCONN se actualizan en un almacenamiento principal del gestor de colas propietario y se vuelven a crear cuando se reinicia el gestor de colas. Sin embargo, si una conexión desde un gestor de colas individual se ha detenido de forma explícita, esta información también se almacena como un distintivo en una matriz de conexión en el correspondiente objeto CFSTRUCT o SMDS, por lo que se conserva al reiniciar el gestor de colas.

## Información de estado y disponibilidad

La información de estado indica el estado de un recurso o conexión (por ejemplo, si todavía no se utiliza, si se utiliza de forma habitual o si necesita recuperarse). Normalmente se describe utilizando la palabra clave STATUS. Los posibles valores dependen del tipo de objeto.

La información de estado habitualmente se actualiza de forma automática, por ejemplo, cuando se detecta un error al utilizar el recurso o conexión. No obstante, en algunos casos, también se puede utilizar un mandato para actualizar el estado, para permitir aquellos casos en los que un gestor de colas no puede determinar el estado correcto automáticamente.

La información de disponibilidad indica si se puede utilizar el recurso o la conexión y si normalmente viene determinado sobre todo por la información de estado. Se implementan tres niveles de disponibilidad para los tipos de recurso o conexión utilizados en el soporte del conjunto de datos de mensaje compartido:

### Disponible

Significa que el recurso está disponible para ser utilizado con normalidad. Ello no significa necesariamente que en este momento esté siendo utilizado (que se puede determinar en lugar del valor STATUS). Para un conjunto de datos, si requiere el proceso de reinicio, permite que el gestor de colas propietario pueda abrirlo, pero otros gestores de colas deberán esperar hasta que el conjunto de datos vuelva al estado ACTIVE.

### No disponible debido a error

Esto significa ha pasado a estar no disponible automáticamente debido a un error y no se espera que vuelva a estar disponible hasta que se realice algún tipo de reparación o proceso de recuperación. Sin embargo, los intentos para que vuelva a estar disponible otra vez están permitidos sin intervención del operador. Tal intento también se puede desencadenar mediante un mandato para marcar el recurso como habilitado, o un mandato que cambia el estado de tal modo que indica que el proceso de recuperación ha finalizado.

La razón por la que el recurso ha pasado a estar no disponible normalmente queda clara por el valor STATUS relacionado, pero en algunos casos pueden haber otras razones para hacer que el recurso esté no disponible, en cuyo caso se proporciona un valor REASON separado para indicar el motivo.

### **No disponible debido a un mandato de operador**

Esto significa que el acceso al recurso se ha inhabilitado explícitamente mediante un mandato. Sólo puede volver a estar disponible utilizando un mandato para habilitarlo de nuevo.

### **Disponibilidad de SMDS**

Para el objeto SMDS compartido, la disponibilidad viene descrita por la palabra clave ACCESS, con los valores posibles ENABLED, SUSPENDED y DISABLED.

La disponibilidad se pueden actualizar utilizando un mandato **RESET SMDS** para el objeto compartido relevante desde cualquier gestor de colas del grupo para establecer ACCESS(ENABLED) o ACCESS(DISABLED).

Si la disponibilidad previamente se ha suspendido ACCESS(SUSPENDED), el hecho de cambiarla por ACCESS(ENABLED) desencadenará un nuevo intento para utilizar el conjunto de datos de mensajes compartido, pero si el error anterior aún está presente, la disponibilidad volverá a ser ACCESS(SUSPENDED).

### **Disponibilidad SMDSCONN**

Para una entrada de información SMDSCONN local, la disponibilidad se describe mediante la palabra clave AVAIL, con los valores posibles NORMAL, ERROR o STOPPED. La disponibilidad se puede actualizar utilizando un mandato **START SMDSCONN** o **STOP SMDSCONN** dirigido a un gestor de colas específico para habilitar o inhabilitar su conexión.

Si la disponibilidad anteriormente era AVAIL(ERROR), el hecho de cambiarla por AVAIL(NORMAL) desencadenará un nuevo intento para utilizar el conjunto de datos de mensajes compartido, pero si el error anterior aún está presente, la disponibilidad volverá a ser de AVAIL(ERROR).

## **Disponibilidad y estado compartido del conjunto de datos de mensaje compartido**

La disponibilidad de cada conjunto de datos de mensaje compartido se gestiona dentro del grupo utilizando información de estado compartido, que se puede visualizar utilizando el mandato **DISPLAY CFSTATUS** con TYPE (SMDS). Se visualiza información de estado de cada gestor de colas que ha activado un conjunto de datos para cada estructura. Cada conjunto de datos puede estar en uno de los estados siguientes:

### **NOTFOUND**

Significa que el correspondiente conjunto de datos aún no se ha activado. Este estado sólo aparece cuando se ha especificado un gestor de colas específico, ya que los conjuntos de datos que no se han activado se omiten cuando se seleccionan todos los gestores de colas.

### **NEW**

Se está abriendo e inicializando el conjunto de datos por primera vez; está listo para su activación.

### **ACTIVE**

Significa que el conjunto de datos está completamente disponible y debería ser asignado y abierto por todos los gestores de colas activos para la estructura.

### **FAILED**

Significa que el conjunto de datos no está disponible para todos (excepto para el proceso de recuperación) y debe ser cerrado y desasignado por todo los gestores de colas.

### **INRECOVER**

Significa que la recuperación de soportes (utilizando RECOVER CFSTRUCT) está en proceso para este conjunto de datos.

### **RECOVERED**

Indica que se ha emitido un mandato para conmutar un conjunto de datos fallido de nuevo al estado activo, pero que primero necesita realizar un proceso de reinicio que aún no se ha completado, por

lo que el conjunto de datos sólo puede ser abierto por el gestor de colas propietario del proceso de reinicio.

### **EMPTY**

El conjunto de datos no contiene mensajes. El conjunto de datos se coloca en este estado si el gestor de colas lo cierra con normalidad y en un momento en el que no contiene mensajes. También puede establecerse en estado EMPTY cuando el contenido del conjunto de datos anteriores debe descartarse porque se ha vaciado la estructura de aplicación (utilizando **RECOVER CFSTRUCT** con **TYPE PURGE** o sólo para una estructura no recuperable, suprimiendo la instancia anterior de la estructura). La próxima vez que el gestor de colas propietario abra el conjunto de datos, la correlación de espacio se restablecerá en vacía y el estado pasará a ser ACTIVE. Como el contenido del conjunto de datos anterior ya no es necesario, un conjunto de datos en este estado se puede sustituir por un conjunto de datos recién asignado, por ejemplo para cambiar la asignación de espacio o trasladarlo a otro volumen.

La salida del mandato incluye la fecha y la hora de habilitación del registro de recuperación, si es que la hay, así como la fecha y la hora en la que el conjunto de datos ha fallado, si actualmente no está activa.

Un conjunto de datos de mensaje compartido se puede poner en un estado FAILED mediante un mandato **RESET SMDS** o automáticamente cuando se detecta cualquiera de los siguientes tipos de error:

- El conjunto de datos no puede ser asignado ni abierto por el gestor de colas propietario.
- La validación de la cabecera de conjunto de datos falla después de que algún gestor de colas lo haya abierto satisfactoriamente.
- Se produce un error de E/S permanente cuando el gestor de colas propietario está leyendo o escribiendo datos.
- Se produce un error de E/S permanente cuando otro gestor de colas está leyendo datos de un conjunto de datos que ha completado satisfactoriamente el proceso de apertura y validación.

Cuando un conjunto de datos tiene el FAILED o INRECOVER, no está disponible para su utilización normal, por lo que si el estado de disponibilidad es ACCESS(ENABLED) se cambia por ACCESS(SUSPENDED).

Se ha colocado un conjunto de datos en estado FAILED pero no se requiere la recuperación de soportes, por ejemplo, debido a que los datos aún eran válidos pero el dispositivo de almacenamiento estaba temporalmente fuera de línea, entonces se puede utilizar el mandato **RESET SMDS** para solicitar un cambio directo al estado RECOVERED.

Cuando el conjunto de datos pasa a tener el estado RECOVERED, ya sea en la terminación del proceso de recuperación o como resultado del mandato **RESET SMDS**, entonces está listo para ser utilizado de nuevo una vez se haya completado el proceso de reinicio. Si tenía el estado ACCESS(SUSPENDED), se conmuta automáticamente de nuevo al estado ACCESS(ENABLED), que permite que el gestor de colas propietario realice el proceso de reinicio. Cuando finaliza el proceso de reinicio, el estado pasa a ACTIVE y el resto de gestores de colas se puede conectar de nuevo al conjunto de datos.

## **Disponibilidad y estado de conexión del conjunto de datos de mensaje compartido**

Cada gestor de colas mantiene el estado local y la información de disponibilidad para su conexión con cada conjunto de datos de mensaje compartido, ya sea de su propiedad o de otros gestores de colas del grupo. Esta información se puede visualizar utilizando el mandato **DISPLAY SMDSCONN**.

Si no puede acceder a un conjunto de datos de mensaje compartido en estado ACTIVE que pertenezca a otro gestor de colas, marca la conexión como no disponible desde su propio punto de vista.

Si el error indica que definitivamente se ha producido un problema con el propio conjunto de datos, el gestor de colas también cambia automáticamente el estado compartido para indicar que el conjunto de datos ahora tiene el estado FAILED. Sin embargo, si el error puede deberse a un problema del entorno, como no estar autorizado a abrir el conjunto de datos, el gestor de colas emite mensajes de error y trata el conjunto de datos como "no disponible", pero no modifica el estado del conjunto de datos compartido. Si el error de entorno resulta ser un problema con el conjunto de datos (por ejemplo, se ha asignado a un dispositivo al que muchos de los gestores de colas no pueden acceder), un operador puede utilizar el

mandato **RESET SMDS** especificando **STATUS(FAILED)** para permitir el conjunto de datos sea recuperado o reparado, según sea necesario.

Si no se ha podido establecer una conexión con un conjunto de datos de mensaje compartido pero el conjunto de datos parece ser válido, se puede desencadenar un nuevo intento de utilizarlo emitiendo un mandato **START SMDSCONN** para el gestor de colas propietario.

Si hay una necesidad operativa de terminar temporalmente la conexión entre un gestor de colas específico y un conjunto de datos, pero el conjunto de datos propiamente dicho no está dañado, el conjunto de datos se puede cerrar y desasignar utilizando el mandato **STOP SMDSCONN**. Si se está utilizando el conjunto de datos, el gestor de colas lo cerrará con normalidad (aunque cualquier solicitud de datos en dicho conjunto de datos se rechazará con un código de retorno). Si se trata del conjunto de datos de propiedad, el gestor de colas guardará la correlación de espacio durante el proceso **CLOSE**, evitando la necesidad de reiniciar el proceso.

Si un conjunto de datos debe estar fuera de servicio temporalmente de todos los gestores de colas (por ejemplo, para moverlo) pero no está dañado, es mejor utilizar **STOP SMDSCONN** para el conjunto de datos relevantes con la opción **CMDScope(\*)** para detener los gestores de colas que lo utilizan en primer lugar, pues así evitará utilizar el proceso de reinicio cuando el conjunto de datos vuelva a estar en servicio. Por el contrario, si el conjunto de datos se marca como **FAILED**, indica a los gestores de colas que deben dejar de utilizarlo inmediatamente, lo que significa que la correlación de espacio no se guardará y tendrá que ser reconstruida por el proceso de reinicio.

Se reintentará acceder a los conjuntos de datos de mensajes compartidos anteriormente en el estado **ACCESS (SUSPENDED)** si se reinicia el gestor de colas.

## Registro de recuperación del conjunto de datos de mensaje compartido

Los mensajes compartidos persistentes se registran para la recuperación de soportes. Esto significa que los mensajes se pueden recuperar después de cualquier error de las estructuras de recurso de acoplamiento o conjuntos de datos de mensajes compartidos, siempre que los registros de recuperación sigan intactos. Los mensajes persistentes también se pueden volver a crear a partir de los registros de recuperación en otro sitio para la recuperación tras desastre.

Cuando los datos de mensaje se graban en un conjunto de datos de mensaje compartido, cada bloque escrito en el conjunto de datos se registra por separado seguido de la entrada de mensaje (incluida la correlación de datos) tal como se ha escrito en el recurso de acoplamiento. El proceso de recuperación siempre recupera la estructura de recurso de acoplamiento, pero no necesita recuperar conjuntos de datos de mensaje compartido individuales excepto cuando el estado del conjunto de datos es **FAILED**, o cuando el estado es **ACTIVE**, pero el registro de cabecera del conjunto de datos ya no es válido, lo que indica que el conjunto de datos se ha vuelto a crear. Un conjunto de datos no se selecciona para la recuperación si su estado es **ACTIVO** y la cabecera del conjunto de datos sigue siendo válida, ni tampoco si su estado es **EMPTY**, lo que indica que no se ha almacenado ningún mensaje en el momento de producirse el error.

## Copias de seguridad del conjunto de datos de mensaje compartido

Cuando se utiliza **BACKUP CFSTRUCT** para efectuar una copia de seguridad de los mensajes compartidos en una estructura de aplicación, se hace una copia de seguridad de todos los datos para mensajes permanentes almacenados en conjuntos de datos de mensaje compartidos al mismo tiempo, tal como se hace para mensajes compartidos permanentes previamente almacenados en BD.

## Recuperación del conjunto de datos de mensaje compartido

Si un conjunto de datos de mensaje compartido se ha dañado o se ha perdido, debe colocarse en el estado **FAILED** para impedir que los gestores de colas lo utilicen hasta que se haya reparado. Normalmente esto sucede automáticamente, pero también puede hacerse utilizando el mandato **RESET SMDS** especificando **STATUS(FAILED)**.

Si el conjunto de datos de mensaje compartido contenía mensajes persistentes, estos se pueden recuperar utilizando el mandato **RECOVER CFSTRUCT**. Este mandato restaura primero los datos de

mensaje persistentes para ese conjunto de datos de mensaje compartido del mandato **BACKUP CFSTRUCT** más reciente y, a continuación, aplica todos los cambios registrados desde ese momento. Si no se ha realizado ningún mandato **BACKUP CFSTRUCT** desde el momento en que se activó por primera vez el conjunto de datos, se restablece en vacío y, a continuación, se aplican todos los cambios desde la activación.

Si el contenido de **CFSTRUCT** y todos los conjuntos de datos de mensajes compartidos no están disponibles, por ejemplo, en una situación de recuperación tras desastre, todos se pueden recuperar en un único mandato **RECOVER CFSTRUCT**.

Si un conjunto de datos de mensaje compartido está dañado, pero la recuperación no estaba activa para la **CFSTRUCT**, el registro que contiene la última copia de seguridad **CFSTRUCT** no está disponible o no se puede utilizar, los mensajes descargados en ese conjunto de datos no se pueden recuperar. En este caso, el mandato **RECOVER CFSTRUCT** con el parámetro **TYPE(PURGE)** se puede utilizar para marcar el conjunto de datos de mensaje compartido como vacío y suprimir todos los mensajes de la estructura que tenía datos almacenados en ese conjunto de datos.

Cuando se emite el mandato **RECOVER CFSTRUCT**, el estado del conjunto de datos de mensaje compartido cambia de **FAILED** a **INRECOVER**. Si la recuperación se completa satisfactoriamente, el estado se cambia automáticamente a **RECOVERED** y, en caso contrario, pasa de nuevo a **FAILED**.

Cuando el conjunto de datos se cambia al estado **RECOVERED**, esto indica al gestor de colas propietario que ahora puede intentar abrir el conjunto de datos y ejecutar el proceso de reinicio.

## Recuperación del conjunto de datos de mensaje compartido y puntos de sincronismo

El proceso de recuperación del conjunto de datos de recuperación de mensaje compartido vuelve a aplicar los cambios para todos los registros de anotaciones completos hasta el final del registro, independientemente de puntos de sincronismo.

Si se han realizado cambios en el punto de sincronismo, el proceso de reinicio o recuperación de **CFSTRUCT** puede dar como resultado una copia de seguridad de solicitudes no confirmadas, por lo que es posible que algunos de los cambios recuperados no se utilicen realmente, pero de todos modos no hay ningún problema en recuperarlos.

También es posible que un mensaje **MQPUT** no confirmado pueda haberse grabado en la estructura, pero los datos correspondientes no se hayan grabado en el conjunto de datos o en el registro (puesto que la terminación de E/S sólo se fuerza al principio del proceso del punto de sincronismo). Es inofensivo porque el proceso de reinicio restituirá la entrada de mensaje en la estructura, por lo que no importa el hecho de que haga referencia a datos no recuperados.

## Proceso de reinicio del conjunto de datos de mensaje compartido

Si la conexión del gestor de colas con **CFSTRUCT** termina con normalidad, el gestor de colas escribe la correlación de espacio de bloque libre para cada conjunto de datos de mensaje compartido en un área de punto de comprobación dentro del conjunto de datos, justo antes de que se cierre el conjunto de datos de mensaje. A continuación, la correlación de espacio se puede volver a leer durante el reinicio de la conexión, siempre y cuando ni **CFSTRUCT** ni el conjunto de datos de mensaje compartido requieran ningún proceso de recuperación antes del siguiente reinicio.

Sin embargo, si un gestor de colas termina de forma anómala, o la estructura o el conjunto de datos requieren algún proceso de recuperación, es necesario efectuar un proceso adicional para reconstruir la correlación de espacio dinámicamente cuando se reinicie la conexión del gestor de colas con la estructura.

Siempre que no sea necesario recuperar el propio conjunto de datos, el reinicio del gestor de colas simplemente explora el contenido actual de la estructura para localizar referencias a los datos de mensaje propiedad del gestor de colas actual y marca los bloques de datos relevantes como ya "poseídos" en la correlación de espacio. Otros gestores de colas pueden continuar utilizando la estructura y leyendo los datos que posee reiniciando el gestor de colas mientras se reconstruye el espacio.

## Reinicio del conjunto de datos de mensaje compartido tras la recuperación

Si se ha tenido que recuperar una copia de seguridad de un conjunto de datos de mensaje compartido, todos los mensajes no permanentes almacenados en el conjunto de datos se perderán y, si se ha recuperado el conjunto de datos utilizando TYPE(PURGE), entonces se perderán todos los mensajes almacenados en el conjunto de datos. Hasta que se complete la recuperación, el conjunto de datos estará marcado como FAILED o INRECOVER, por lo que cualquier intento de leer uno de los mensajes afectados desde otro gestor de colas generará un código de error que indica que el conjunto de datos está temporalmente no disponible.

Cuando el conjunto de datos se ha recuperado, el estado cambia a RECOVERED, lo que permite que el gestor de colas propietario pueda abrirlo para el proceso de reinicio, pero el conjunto de datos sigue estando no disponible para otros gestores de colas. El reinicio del gestor de colas explora la estructura para reconstruir la correlación de espacio de todos los mensajes restantes. La exploración también comprueba los mensajes cuyos datos se han perdido y los suprime de la estructura (o si es necesario los marca como perdidos para suprimirlos posteriormente).

El estado del conjunto de datos se cambia automáticamente de RECOVERED (recuperado) a ACTIVE (activo) cuando finaliza la exploración de reinicio, momento en el que otros gestores de colas pueden empezar a utilizarlo de nuevo.

## Información de uso del conjunto de datos de mensaje compartido

El mandato DISPLAY USAGE ahora muestra información sobre el espacio del conjunto de datos de mensaje compartido y sobre el uso de la agrupación de almacenamiento intermedio de cualquiera de los conjuntos de datos de mensaje compartidos actualmente abierto. Esta información se visualiza tanto si se especifica la opción nueva TYPE(SMDS) o una opción existente TYPE(ALL).

## Consideraciones sobre el rendimiento y la capacidad de los datos de mensaje

### Supervisión del uso del conjunto de datos

El mandato **DISPLAY USAGE** puede visualizar el porcentaje actual lleno de cada conjunto de datos de mensaje compartido en propiedad con la opción **TYPE (SMDS)**.

El gestor de colas normalmente expandirá automáticamente un conjunto de datos de mensaje compartido cuando alcance el 90% completo, siempre que la opción **DSEXPAND (YES)** esté en vigor para la definición SMDS. Esto se aplica cuando la opción SMDS se establece en **DSEXPAND (YES)** o la opción SMDS se establece en **DSEXPAND (DEFAULT)** y la opción predeterminada CFSTRUCT se establece en **DSEXPAND (YES)**.

Si el intento de expansión falla porque no se ha especificado un tamaño de asignación secundaria al crear el conjunto de datos (proporcionando el mensaje IEC070I con el código de razón 203), el gestor de colas repite la solicitud de expansión utilizando una asignación secundaria de alteración temporal de aproximadamente el 20% del tamaño actual.

Cuando un conjunto de datos se expande, las extensiones del nuevo conjunto de datos se formatean como parte del proceso de expansión, que puede tardar décimas de segundo, o minutos si las extensiones son de gran tamaño. El espacio nuevo podrá utilizarse una vez que ha finalizado el formateo y que el catálogo se ha actualizado y muestra el nuevo intervalo superior de control de uso.

Si se crean mensajes nuevos rápidamente, es posible que el conjunto de datos existente se llene antes de que finalice el proceso de expansión. En este caso, se suspende temporalmente toda solicitud que no haya podido asignar espacio hasta que el intento de expansión se complete y el nuevo espacio quede disponible para su uso. Si la expansión ha sido satisfactoria, la solicitud se volverá a repetir automáticamente.

Si un intento de expansión falla, debido a la falta de espacio disponible o debido a que ya se ha alcanzado el número máximo de extensiones, se emite un mensaje que da la razón de la anomalía, la opción de alteración temporal para el SMDS afectado se modifica automáticamente a **DSEXPAND (NO)** para evitar más intentos de expansión. En este caso, existe el riesgo de que el conjunto de datos

pueda llenarse, en cuyo caso deberá tomar medidas, tal como se describe en [El conjunto de datos se llena](#).

### Supervisión del uso de la estructura de la aplicación

El nivel de uso de una estructura de aplicación se puede visualizar utilizando el mandato MVS **DISPLAY XCF, STRUCTURE** especificando el nombre completo de la estructura de aplicación (incluido el prefijo de grupo de compartición de colas). El mensaje de respuesta IXC360I muestra el uso actual de elementos y entradas.

Cuando el uso de estructura supera el valor **FULLTHRESHOLD** especificado en la política CFRM, el sistema emite el mensaje IXC585E y puede que lleve a cabo acciones **ALTER** automáticas si se especifica, que podrían alterar la entrada a la proporción de elementos o aumentar el tamaño de la estructura.

### Optimización del tamaño de la agrupación de almacenamiento intermedio

Cada almacenamiento intermedio de una agrupación de almacenamiento intermedio compartido se utiliza para leer o escribir un rango contiguo de páginas de un mensaje hasta el tamaño de bloque lógico. Si el mensaje se desborda en más bloques, cada rango de páginas de un bloque aparte requiere un almacenamiento intermedio separado.

Los almacenamientos intermedios que contienen los datos de mensaje después de una operación de lectura o escritura se conservan en el almacenamiento y se reutilizan utilizando un esquema de memoria caché LRU (el menos utilizado recientemente) de modo que una solicitud para leer los mismos datos de nuevo poco después no tendrá que ir al disco. Esto provoca una optimización significativa cuando las aplicaciones que se ejecutan en el mismo sistema escriben y vuelven a leer los mensajes compartidos poco después. Si los mensajes que pertenecen a otro gestor de colas se examinan para fines de selección y se recuperan, también se evita la necesidad de volver a leer el mensaje desde el disco.

Esto significa que el número de almacenamientos intermedios necesarios para cada estructura de aplicación es uno para cada solicitud de API simultánea que lee o escribe mensajes grandes para dicha estructura de aplicación más el número de almacenamientos intermedios adicionales que se utilizará para los datos de acceso reciente a fin de optimizar los accesos de lectura subsiguientes.

Para agrupaciones de almacenamiento intermedio compartidas, si no hay suficientes almacenamientos intermedios, las solicitudes de la API deberán esperar si un almacenamiento intermedio no está disponible de inmediato. Sin embargo, esta situación debe evitarse, ya que puede provocar la degradación del rendimiento de forma significativa.

Las estadísticas del mandato **DISPLAY USAGE** para agrupaciones de almacenamientos intermedios compartidos muestran si han habido esperas de almacenamiento intermedio en el intervalo de estadísticas actual y también muestra el número más bajo de almacenamientos intermedios libres (o un valor negativo que indica el número máximo de hebras que han esperado un almacenamiento intermedio en cualquier momento), el número de almacenamientos intermedios que han guardado datos y el porcentaje de veces que una solicitud de almacenamiento intermedio ha encontrado satisfactoriamente datos guardados en la cadena LRU ("Aciertos de LRU") en lugar de tener que leerlos ("Errores de LRU")<sup>1</sup>.

- Si ha habido esperas, el número de almacenamientos intermedios debe aumentarse.
- Si hay muchos almacenamientos intermedios que no se utilizan, el número de almacenamientos intermedios puede reducirse para tener más espacio disponible en la región para otros fines.
- Si hay muchos almacenamientos intermedios que contienen datos guardados, pero la proporción de lecturas que fueron aciertos contra dichos datos guardados es muy pequeña, el número de almacenamientos intermedios se puede reducir si el almacenamiento se pudiera aprovechar para otros fines. El número de almacenamientos intermedios no debe reducirse más que el número más bajo de almacenamientos intermedios libres, ya que podría desencadenar esperas, y preferiblemente debe ser más alto que el número más bajo de almacenamientos intermedios libres, que generalmente está por encima de cero.

---

<sup>1</sup>  $(\text{Hits} / (\text{Hits} + \text{Misses})) * 100$

## Supresión de conjuntos de datos de mensaje compartidos

El mandato `DELETE CFSTRUCT` (que sólo se permite cuando todas las colas compartidas de la estructura están vacías y cerradas) no suprime los conjuntos de datos de mensajes compartidos por sí mismos, pero se pueden suprimir de la forma habitual después de que se haya completado este mandato. Si el mismo conjunto de datos se va a reutilizar como un conjunto de datos de mensajes compartido, primero deberá ser reformateado para restablecerlo al estado vacío.

## Situaciones de excepción para conjuntos de datos de mensajes compartidos

Hay una serie de situaciones de excepción que pueden producirse durante el uso normal, aunque no haya errores de software o hardware.

### El conjunto de datos se llena

Si un conjunto de datos se llena, pero no se puede expandir, o el intento de expansión falla, las aplicaciones que utilizan el gestor de colas correspondiente para escribir mensajes grandes en la estructura de aplicación correspondiente recibirán el error 2192, `MQRC_STORAGE_MEDIUM_FULL` (también conocido como `MQRC_PAGESET_FULL`).

Un conjunto de datos puede llenarse debido a una anomalía en la aplicación que debería procesar los datos, lo que provoca que se acumule una gran reserva de mensajes. De ser así, expandir el conjunto de datos sólo será una solución temporal; es importante que vuelva a funcionar la aplicación cuando antes.

Si hay más espacio que pudiera estar disponible, el mandato **ALTER SMDS** podría utilizarse para establecer **DSEXPAND(YES)** o **DSEXPAND(DEFAULT)** (se presupone que se ha establecido YES o el valor predeterminado **DSEXPAND** para la definición `CFSTRUCT`) para activar un reintento. Sin embargo, si el motivo del error era que se habían alcanzado las extensiones máximas, se rechazará el nuevo intento de expansión con un mensaje y se volverá a establecer **DSEXPAND(NO)**. En este caso, el único modo de expandirlo más es reasignándolo, que implica que no esté disponible temporalmente, tal como se describe a continuación.

### El conjunto de datos debe moverse o reasignarse

Si un conjunto de datos debe moverse o expandirse, pero su uso es normal, puede dejarse fuera de uso temporalmente para permitir que se mueva o se reasigne. Cualquier solicitud de la API que intente utilizar el conjunto de datos mientras éste no está disponible recibirá el código de razón `MQRC_DATA_SET_NOT_AVAILABLE`.

1. Utilice el mandato **RESET SMDS** para marcar el conjunto de datos como **ACCESS(DISABLED)**. Esto hará que todos los gestores de colas conectados actualmente cierren con normalidad el conjunto de datos y lo desasignen.
2. Mueva o reasigne el conjunto de datos como precise, copiando el contenido antiguo en el conjunto de datos de reciente asignación, por ejemplo utilizando el mandato **REPRO** de servicios de método de acceso (AMS).

No intente preformatear el nuevo conjunto de datos antes de copiar los datos antiguos en éste, ya que los datos copiados se añadirían al final del conjunto de datos formateados.

3. Utilice el mandato **RESET SMDS** para marcar de nuevo el conjunto de datos como **ACCESS(ENABLED)** y activar su uso.

Si el contenido antiguo es más pequeño que el tamaño del nuevo conjunto de datos, el resto del espacio se preformateará automáticamente cuando el nuevo conjunto de datos se abra.

Si el tamaño del contenido antiguo es mayor que el del nuevo conjunto de datos, el gestor de colas debe explorar los mensajes de la estructura del recurso de acoplamiento y reconstruir la correlación de espacio para garantizar que no se haya perdido ningún dato activo. Si se encuentra alguna referencia a un bloque de datos que está fuera de las nuevas extensiones, el conjunto de datos se marca como **STATUS(FAILED)** y se debe reparar sustituyendo el conjunto de datos

por uno del tamaño correcto y copiando de nuevo el conjunto de datos antiguo en él o utilizando **RECOVER CFSTRUCT** para recuperar los mensajes persistentes.

### **La estructura de recurso de acoplamiento tiene poco espacio**

Si la estructura de recurso de acoplamiento se está quedando sin espacio, y aparece el mensaje IXC585E, compruebe si las reglas de descarga se han establecido para garantizar que se descargue la máxima cantidad de datos en este caso. Si no es así, las reglas de descarga se pueden modificar utilizando el mandato **ALTER CFSTRUCT**.

### **Situaciones de error para conjuntos de datos de mensajes compartidos**

Hay una serie de problemas a tener en cuenta, que sólo suceden a causa de un error y no en situaciones de funcionamiento normal.

#### **Los conjuntos de datos de propiedad no pueden abrirse**

Si el gestor de colas que posee un conjunto de datos de mensaje compartido no puede asignarlo ni abrirlo, o los atributos de conjunto de datos no están soportados, el gestor de colas establece un valor de estado **SMDSCONN** adecuado de **ALLOCFAIL** o **OPENFAIL** y establece la disponibilidad de **SMDSCONN** en **AVAIL (ERROR)**. También establece la disponibilidad de SMDS en **ACCESS (SUSPENDED)**. Cuando se haya corregido el error, utilice el mandato **RESET SMDS** para establecer **ACCESS (ENABLED)** para desencadenar un reintento, o emita el mandato **START SMDSCONN** para el gestor de colas propietario.

#### **No se puede abrir el conjunto de datos de sólo lectura**

Si un gestor de colas no puede asignar o abrir un conjunto de datos de mensaje compartido que es propiedad de otro gestor de colas y está marcado como **STATUS (ACTIVE)**, supone que probablemente se deba a un problema específico con su conexión al conjunto de datos (representado por el objeto **SMDSCONN**) en lugar de un problema con el propio conjunto de datos.

Marca el **SMDSCONN** como **STATUS (ALLOCFAIL)** o **STATUS (OPENFAIL)** según corresponda y marca la disponibilidad de **SMDSCONN** como **AVAIL (ERROR)** para evitar más intentos de utilizarlo.

Si el problema se puede corregir sin que ello afecte al estado del conjunto de datos, utilice el mandato **START SMDSCONN** para desencadenar un reintento.

Si el problema resulta ser un problema con el propio conjunto de datos, se puede utilizar el mandato **RESET SMDS** para marcar el conjunto de datos como **STATUS (FAILED)** hasta que se haya recuperado. Cuando se haya recuperado el conjunto de datos, la acción de volver a cambiar el estado a **STATUS (ACTIVE)** hará que se notifiquen otros gestores de colas. Si el **SMDSCONN** se marca como **AVAIL (ERROR)**, se volverá a cambiar automáticamente a **AVAIL (NORMAL)** para desencadenar un nuevo intento de abrir el conjunto de datos.

#### **La cabecera de conjunto de datos está dañada**

Si el conjunto de datos se ha abierto correctamente pero el formato de la información de cabecera es incorrecto, el gestor de colas cierra y desasigna el conjunto de datos y establece el estado establecido en **STATUS (FAILED)** y la disponibilidad en **ACCESS (SUSPENDED)**. Esto permite utilizar **RECOVER CFSTRUCT** para recuperar el contenido.

Si el error se ha producido porque el conjunto de datos contenía datos residuales de otro uso y no se había vuelto a formatear posteriormente, formatee el conjunto de datos y utilice el mandato **RESET SMDS** para cambiar el estado a **STATUS (RECOVERED)**.

De lo contrario, el conjunto de datos se recuperará.

#### **El conjunto de datos se ha vaciado de forma inesperada**

Si el gestor de colas abre un conjunto de datos que está marcado como **STATUS (ACTIVE)**, pero encuentra que no se ha inicializado o se ha preformateado recientemente, pero de lo contrario es válido, el gestor de colas cierra y desasigna el conjunto de datos de mensajes compartidos y, a continuación, establece el estado en **STATUS (FAILED)** y la disponibilidad en **ACCESS (SUSPENDED)**.

## El conjunto de datos tiene errores de E/S permanentes

Si un conjunto de datos tiene errores de E/S permanentes después de un proceso **OPEN** satisfactorio, probablemente deba recuperarse. El gestor de colas marcará el conjunto de datos como **STATUS (FAILED)** para que todos los gestores de colas conectados actualmente se cierren y desasignen.

## El conjunto de datos tiene errores de E/S recuperables

Si hay problemas con el hardware del conjunto de datos, es posible que dé como resultado errores recuperables de E/S que no se reflejan en el gestor de colas, pero que causa una degradación significativa del rendimiento, y también indica un riesgo de errores de E/S permanente en el futuro cercano.

En este caso, el conjunto de datos puede desactivarse para la recuperación utilizando el mandato **RESET SMDS** para marcarlo como **STATUS (FAILED)**. Esto hará que todos los gestores de colas cierren y desasignen el conjunto de datos. Así, por ejemplo, podría trasladarse a un nuevo volumen antes de que volver a establecerlo como disponible.

Cuando un conjunto de datos se establece como disponible de este modo, la correlación de espacio no se guarda de modo que el proceso de reinicio de la conexión del gestor de colas deberá explorar la estructura de recurso de acoplamiento para localizar los mensajes en el conjunto de datos y reconstruir la correlación de espacio antes de que el conjunto de datos pueda estar disponible de nuevo. Como alternativa, si el conjunto de datos de mensaje compartido todavía se puede utilizar, puede establecerse como no disponible más suavemente mediante el mandato **RESET SMDS** para marcar el conjunto de datos como **ACCESS (DISABLED)** hasta que esté preparado para volver a establecerse como disponible.

## El contenido del conjunto de datos es incorrecto

El gestor de colas no puede detectar directamente que un conjunto de datos contiene datos incorrectos o que no están actualizados, por ejemplo debido a que un volumen que incluye dicho conjunto de datos se haya tenido que restaurar a partir de copias de seguridad. Sin embargo, realiza comprobaciones de integridad que hacen que sea muy improbable que dichos errores den lugar a que programas de aplicación pudieran ver datos de mensaje incorrectos.

Por razones de comprobación de integridad, a cada bloque de mensajes del conjunto de datos se añade un prefijo con una copia del ID de entrada del recurso de acoplamiento correspondiente, que incluye una indicación de la hora exclusiva, que se comprueba siempre que se lee el bloque de mensajes, antes de pasar los datos de mensaje al programa de usuario. Si el prefijo del bloque de mensaje no coincide con el ID de entrada (y la entrada del recurso de acoplamiento no se ha suprimido mientras tanto), se presupone que el bloque de mensaje está dañado y no puede utilizarse.

Si mensaje dañado es persistente, el conjunto de datos se marca como **STATUS (FAILED)** y el contenido de la estructura debe recuperarse utilizando el mandato **RECOVER CFSTRUCT**. Si el mensaje dañado es no persistente, no hay modo de recuperarlo, por lo que se emite un mensaje de diagnóstico y se elimina la entrada del mensaje del recurso de acoplamiento correspondiente.

Si no está disponible ninguna correlación de espacio guardada cuando se abre el conjunto de datos, se reconstruye mediante la exploración de la estructura de recurso de acoplamiento en busca de referencias a los datos del conjunto de datos. Durante esta exploración, el gestor de colas lleva a cabo varias acciones:

1. El gestor de colas determina la ubicación del mensaje más reciente (si lo hubiera) que queda actualmente en el conjunto de datos.
2. A continuación, el gestor de colas lee dicho mensaje del conjunto de datos para asegurarse de que el prefijo del bloque coincida con el ID de entrada.

Estas acciones garantizan que el gestor de colas detecte cualquier caso en el que el conjunto de datos es de nivel inferior, y marca el conjunto de datos como **FAILED**. Esta comprobación, no obstante, tolera el caso en el que el conjunto de datos se ha restaurado desde una copia anterior y no se han añadido mensajes desde entonces o todos los nuevos mensajes añadidos desde la copia se han leído y eliminado.

Para proteger contra datos que sean de un nivel inferior, en el caso en que el conjunto de datos se haya cerrado normalmente, el gestor de colas lleva a cabo varias acciones:

1. El gestor de colas guarda una copia de la indicación de hora de la correlación de espacio en el objeto SMDS dentro de Db2 cuando el conjunto de datos de cierra normalmente.
2. Después, el gestor de colas comprobará que la indicación de la hora de la correlación de espacio sea la misma cuando se vuelvan a abrir los datos.

Si la indicación de hora no coincide, esto podría indicar que se ha utilizado una copia del conjunto de datos de nivel inferior, por lo que el gestor de colas ignora la correlación de espacio existente y la vuelve a recrear, que sólo será correcta si no se ha perdido ningún dato de mensajes.

**Nota:** Estas comprobaciones de integridad no garantizan la detección de un conjunto de datos dañado o de nivel inferior en todos los casos teóricamente posibles. Por ejemplo, no detectarán un caso en el que el inicio de un bloque de mensaje es válido, pero el resto de los datos se ha sobrescrito parcialmente.

## Escenarios de recuperación para conjuntos de datos de mensaje compartido

En esta sección se describen los escenarios de recuperación del conjunto de datos de mensaje compartido.

### Recuperación de conjuntos de datos en los que no se ha perdido ningún dato

En algunos casos, el contenido correcto de un conjunto de datos con errores puede restaurarse sin necesidad real de recuperación. Un ejemplo es cuando un conjunto de datos contiene datos residuales de un uso anterior y no se ha preformateado de nuevo, lo cual se puede solucionar preformateándolo. Otro caso es cuando se ha movido un conjunto de datos y se ha producido un error en el proceso de copiar los datos. Esto se puede solucionar copiando los datos de nuevo correctamente.

En estos casos, el conjunto de datos corregido puede volver a estar disponible utilizando el mandato **RESET SMDS** para establecer **STATUS (RECOVERED)**. Si la disponibilidad es actualmente **ACCESS (SUSPENDED)**, se volverá a establecer automáticamente en **ACCESS (ENABLED)**.

Cuando se notifica al gestor de colas propietario que el conjunto de datos se ha recuperado, explora el contenido de la estructura para reconstruir el mapa de espacios y, a continuación, cambia el estado a **STATUS (ACTIVE)**. Los otros gestores de colas pueden empezar a leer de nuevo el conjunto de datos.

### Recuperación del conjunto de datos con **TYPE(NORMAL)**

Si el contenido de un conjunto de datos se ha perdido, pero la estructura de la aplicación se ha definido con **RECOVER (YES)** y los registros de recuperación están disponibles, el mandato **RECOVER CFSTRUCT** se puede utilizar para recuperar los mensajes permanentes almacenados en la estructura, incluidos los datos de mensaje descargados en conjuntos de datos de mensaje compartido. Este mandato restaura el estado actual utilizando la información registrada por el mandato **BACKUP CFSTRUCT** junto con todos los cambios registrados en los mensajes permanentes desde el momento de la copia de seguridad.

El mandato **RECOVER CFSTRUCT** siempre recupera todos los mensajes permanentes en la estructura de recurso de acoplamiento junto con los datos de mensajes descargados almacenados en Db2. Para los datos descargados almacenados en conjuntos de datos de mensajes compartidos, cada conjunto de datos solo se selecciona para el proceso de recuperación si ya está marcado como **STATUS (FAILED)** o si se encuentra que está inesperadamente vacío o no es válido cuando se abre mediante el proceso de recuperación. No es necesario recuperar los conjuntos de datos de mensaje compartido marcados como activos y que pasan las comprobaciones de validación, ya que los datos de mensaje existentes ya son correctos, pero la cabecera se actualiza para indicar que cualquier correlación de espacio guardada deberá reconstruirse después de la recuperación.

El proceso de recuperación sólo es posible cuando la estructura se ha marcado como anómala (Failed), ya que el contenido completo de la estructura deberá reconstruirse durante el proceso de recuperación. Sin embargo, si al menos un conjunto de datos de mensaje compartido se ha

marcado como anómalo, el mandato **RECOVER CFSTRUCT** marcará automáticamente la estructura como anómala si es necesario para permitir que continúe el proceso de recuperación.

La recuperación se puede realizar desde cualquier gestor de colas del grupo de compartición de colas, siempre que se le haya otorgado acceso de escritura a los conjuntos de datos relevantes.

Sólo se hace una copia de seguridad y se registran los mensajes permanentes, de modo que el proceso de recuperación normal restaura todos los mensajes permanentes, pero se perderán los mensajes no permanentes de la estructura.

Cuando se ha completado la recuperación, cualquier conjunto de datos seleccionado para la recuperación se cambia automáticamente a **STATUS (RECOVERED)** y si la disponibilidad es **ACCESS (SUSPENDED)** se cambia a **ACCESS (ENABLED)**. El gestor de colas vuelve a crear el mapa de espacios para cada conjunto de datos explorando los mensajes en el recurso de acoplamiento y, a continuación, marca el conjunto de datos como **STATUS (ACTIVE)** para que se pueda volver a utilizar.

### Recuperación del conjunto de datos con **TYPE(PURGE)**

Para una estructura recuperable, si el contenido del conjunto de datos se ha perdido, pero no se puede llevar a cabo la recuperación por alguna razón, por ejemplo porque los registros de recuperación no están disponibles o la recuperación tardaría demasiado, puede utilizarse el mandato **RECOVER CFSTRUCT** con **TYPE (PURGE)** para que la estructura vuelva a recuperar un estado que pueda utilizarse. La estructura se restablece al estado vacío y marca todos los conjuntos de datos asociados como **STATUS (EMPTY)**.

### Supresión de la estructura de la aplicación

Si se suprime una estructura de aplicación no recuperable utilizando el mandato MVS **SETXCF FORCE**, o como resultado de una anomalía de estructura, la próxima vez que se conecte la estructura, se emitirá el mensaje CSQE028I para indicar que la estructura se ha restablecido y que se han descartado todos los mensajes existentes, y que los conjuntos de datos existentes se restablecerán automáticamente también en **STATUS (EMPTY)**. Esta acción hace que una estructura no recuperable se pueda volver a utilizar después de la pérdida de datos en la estructura o en cualquiera de los conjuntos de datos asociados.

Si se suprime una estructura de la aplicación recuperable, se tratará de la misma forma que si la estructura hubiera fallado.

### Error en la recuperación del conjunto de datos

Si **RECOVER CFSTRUCT** no se puede completar por algún motivo, por ejemplo debido a que un conjunto de datos de registro ya no está disponible, o porque el gestor de colas terminó mientras la recuperación estaba en proceso, todo conjunto de datos para el que se ha iniciado como mínimo la recuperación se marcará en la cabecera para indicar que se ha intentado la recuperación parcial y el conjunto de datos se dejará en el estado **STATUS (FAILED)**.

En este caso, las opciones son repetir la solicitud de recuperación original o recuperarse con **TYPE (PURGE)** en su lugar, descartando los datos existentes.

Si se intenta marcar el conjunto de datos como **STATUS (RECOVERED)** sin recuperarlo realmente, la próxima vez que se abra el gestor de colas, detectará que la cabecera indica una recuperación incompleta y la marcará como **STATUS (FAILED)** de nuevo.

### Recuperación tras desastre fuera de sitio

Para la recuperación tras desastre fuera de sitio, se pueden volver a crear mensajes compartidos permanentes utilizando únicamente los objetos compartidos de Db2 que contienen las definiciones CFSTRUCT y la información de estado asociada de SMDS.

Después de configurar las tablas de Db2 que contienen las definiciones, la estructura de la aplicación y los conjuntos de datos de mensajes compartidos se pueden configurar como vacíos. Cuando un gestor de colas se conecta a ellos y descubre que están inesperadamente vacíos, los marca como anómalos, después de lo cual se puede utilizar un único mandato **RECOVER CFSTRUCT** para recuperar todos los mensajes permanentes de todas las estructuras afectadas.

## Mandatos relacionados con SMDS

Este tema describe y proporciona acceso a los mandatos relacionados con los conjuntos de datos de mensaje compartidos.

Visualizar y modificar las opciones de **CFSTRUCT** relacionadas con la descarga de mensajes grandes (**OFFLOAD** y reglas de descarga) y los conjuntos de datos de mensajes compartidos (**DSGROUP**, **DSBLOCK**, **DSBUFS**, **DSEXPAND**):

- [DISPLAY CFSTRUCT](#)
- [DEFINE CFSTRUCT](#)
- [ALTER CFSTRUCT](#)
- [DELETE CFSTRUCT](#)

Visualizar el estado de **CFSTRUCT** relacionado con la descarga de mensajes grandes (**OFFLDUSE**):

- [DISPLAY CFSTATUS](#)

Visualizar y alterar temporalmente las opciones de conjuntos de datos (**DSEXPAND** y **DSBUFS**) para gestores de colas individuales:

- [DISPLAY SMDS](#)
- [ALTER SMDS](#)

Visualizar o modificar el estado y la disponibilidad de los conjuntos de datos dentro del grupo de compartición de colas:

- [DISPLAY CFSTATUS TYPE\(SMDS\)](#)
- [RESET SMDS](#)

Visualizar el uso del espacio de conjunto de datos de SMDS y la información del uso del almacenamiento para un gestor de colas:

- [DISPLAY USAGE TYPE\(SMDS\)](#)

Visualizar o modificar el estado y la disponibilidad de las conexiones (**SMDSCONN**) a los conjuntos de datos desde un gestor de colas individual:

- [DISPLAY SMDSCONN](#)
- [START SMDSCONN](#)
- [STOP SMDSCONN](#)

Copia de seguridad y recuperación de mensajes compartidos, incluidos los datos de mensajes grandes en SMDS cuando sea necesario:

- [BACKUP CFSTRUCT](#)
- [RECOVER CFSTRUCT](#)

## Ventajas de la utilización de colas compartidas

La cola compartida permite que las aplicaciones IBM MQ sean escalables, altamente disponibles y permite implementar el equilibrio de carga de trabajo.

### Las ventajas de las colas compartidas

La arquitectura de cola compartida, donde los servidores clonados extraen el trabajo de una sola cola compartida, tiene algunas propiedades muy útiles:

- Es escalable, ya que añade nuevas instancias de la aplicación de servidor o incluso añade una nueva imagen de z/OS a un gestor de colas (en el grupo de compartición de colas) y una copia de la aplicación.
- Es de alta disponibilidad.

- Realiza de forma natural el equilibrio de carga de trabajo de *extracción*, basándose en la capacidad de proceso disponible de cada gestor de colas del grupo de compartición de colas.

## Utilización de colas compartidas para alta disponibilidad

Los ejemplos siguientes ilustran cómo puede utilizar una cola compartida para aumentar la disponibilidad de las aplicaciones.

Considere un caso de ejemplo de IBM MQ en el que las aplicaciones cliente que se ejecutan en la red desean realizar solicitudes de aplicaciones de servidor que se ejecutan en z/OS. La aplicación de cliente construye un mensaje de solicitud y lo coloca en una cola de solicitudes. El cliente espera entonces una respuesta del servidor, que se envía a la cola de respuestas especificada en el descriptor de mensaje del mensaje de solicitud.

IBM MQ gestiona el transporte del mensaje de solicitud desde la máquina cliente hasta la cola de entrada del servidor en z/OS y de la respuesta del servidor de nuevo al cliente. Mediante la definición de la cola de entrada del servidor como una cola compartida, todos los mensajes transferidos a la cola se pueden recuperar en cualquier gestor de colas del grupo de compartición de colas. Esto significa que puede configurar un gestor de colas en cada imagen de z/OS en el sysplex y, conectándolos todos al mismo grupo de compartición de colas, cualquiera de ellos puede acceder a los mensajes en la cola de entrada del servidor.

Los mensajes en la cola de entrada del servidor aún están disponibles, incluso si uno de los gestores de colas termina anormalmente o tiene que detenerlo por razones administrativas. Puede colocar una imagen completa de z/OS fuera de línea y los mensajes seguirán estando disponibles.

Para aprovechar esta disponibilidad de los mensajes en una cola compartida, ejecute una instancia de la aplicación de servidor en cada imagen de z/OS en el sysplex para ofrecer una mayor capacidad y disponibilidad de la aplicación de servidor, tal como se muestra en [Figura 60 en la página 194](#).

Una instancia de la aplicación de servidor recupera un mensaje de solicitud de la cola compartida y, basándose en el contenido, realiza su proceso, produciendo un resultado que se envía de nuevo al cliente como un mensaje de IBM MQ. El mensaje de respuesta se destina a la cola de respuestas y el gestor de colas de respuesta especificados en el descriptor de mensaje del mensaje de solicitud.

Hay una serie de opciones que puede utilizar para configurar la vía de retorno. Para obtener más información sobre estas opciones, consulte [“Gestión de colas distribuidas y grupos de compartición de colas” en la página 213](#).

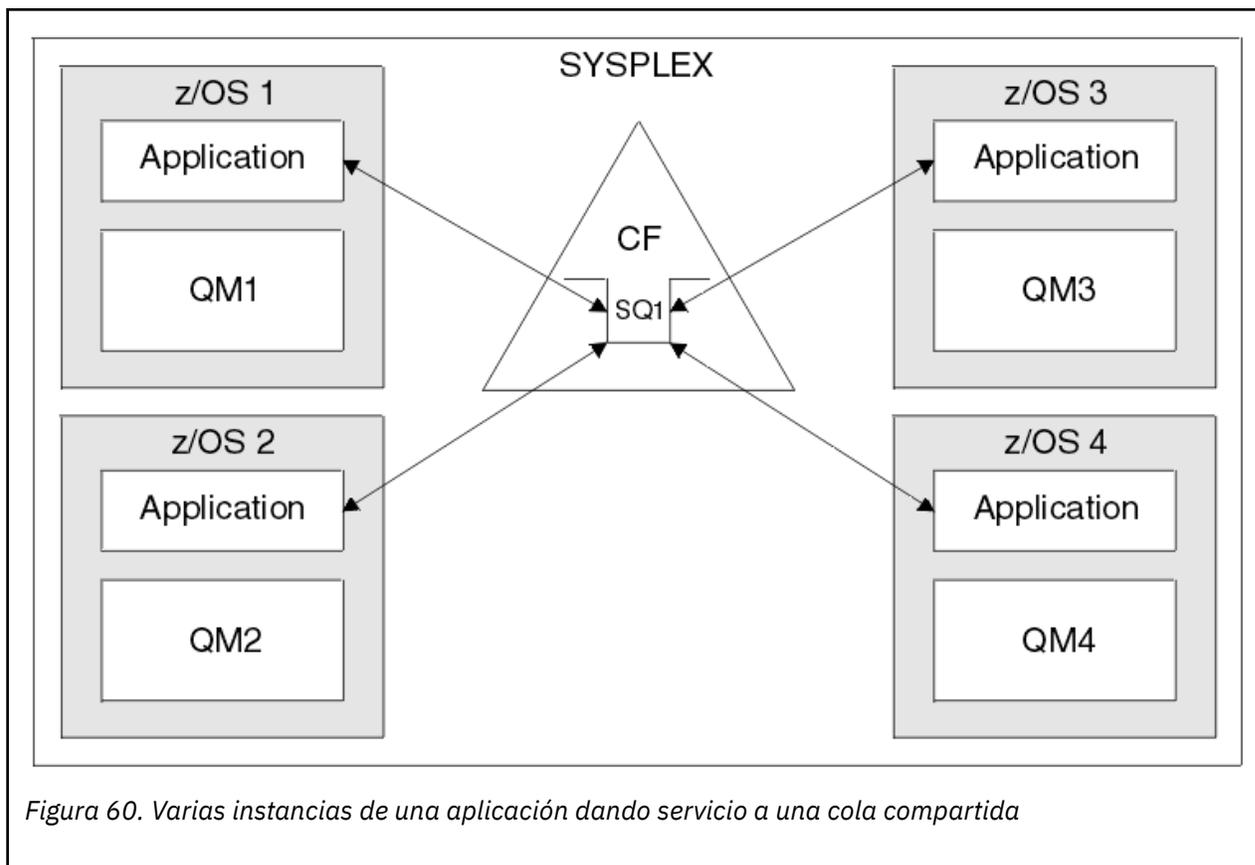


Figura 60. Varias instancias de una aplicación dando servicio a una cola compartida

## Recuperación de iguales

Para mejorar aún más la disponibilidad de mensajes en un grupo de compartición de colas, IBM MQ detecta si otro gestor de colas del grupo se desconecta del recurso de acoplamiento de forma anómala y completa las unidades de trabajo para ese gestor de colas que todavía están pendientes, cuando ello sea posible. Esta característica se conoce como *recuperación de iguales*.

Supongamos que un gestor de colas termina de forma anómala en un punto donde una aplicación ha recuperado un mensaje de solicitud de una cola en el punto de sincronización, pero todavía no ha colocado el mensaje de respuesta o ha comprometido la unidad de trabajo. Otro gestor de colas del grupo de compartición de colas detecta la anomalía y restituye las unidades de trabajo en curso que se ejecutan en el gestor de colas que ha fallado. Esto significa que el mensaje de solicitud se coloca de nuevo en la cola de solicitudes y está disponible para que lo procese una de las otras instancias de servidor, sin esperar a que el gestor de colas anómalo se reinicie.

Si IBM MQ no puede resolver automáticamente una unidad de trabajo, puede resolver la parte compartida manualmente para habilitar otro gestor de colas en el grupo de compartición de colas para continuar procesando ese trabajo.

## ► z/OS **Uso de la memoria de clase de almacenamiento (SCM) con colas compartidas**

El uso de la memoria de clase de almacenamiento (SCM) puede ser ventajoso cuando se utiliza con colas compartidas de IBM MQ for z/OS.

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z® para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

Las máquinas z13, zEC12 y zBC12 permiten la instalación de tarjetas Flash Express. Estas tarjetas contienen unidades SSD (unidades de estado sólido) flash. Después de la instalación, el almacenamiento flash de las tarjetas se puede asignar a una o varias LPAR, donde se conocen normalmente como SCM.

SCM se sitúa entre el almacenamiento real y el dispositivo de almacenamiento de acceso directo (DASD) en términos tanto de latencia de E/S como de coste. Puesto que SCM no tiene ninguna parte móvil, muestra latencias de E/S mucho menores que DASD.

SCM también es mucho más económico que el almacenamiento real. En consecuencia, se puede instalar una gran cantidad de almacenamiento a un coste relativamente bajo; por ejemplo, un par de tarjetas Flash de Express contienen 1424 GB de almacenamiento utilizable.

Estas características significan que SCM es útil cuando hay que extraer una gran cantidad de datos del almacenamiento real en un breve periodo de tiempo, ya que los datos se pueden grabar en SCM de forma mucho más rápida que lo que se pueden grabar en DASD. Este punto específico puede ser muy útil cuando se utilizan estructuras de lista de recurso de acoplamiento (CF) que contienen colas compartidas de IBM MQ.

## Por qué las estructuras de lista se llenan

Cuando se define una estructura de CF, se configura con un atributo SIZE que describe el tamaño máximo de la estructura. Dado que las estructuras CF siempre residen permanentemente en el almacenamiento real, la suma de los atributos SIZE de las estructuras que están definidas en un CF debe ser menor que la cantidad de almacenamiento real asignado al CF.

En consecuencia, hay una presión constante para mantener el valor de SIZE de cualquier estructura determinada en el mínimo valor posible, de modo que más estructuras puedan caber en el CF. Sin embargo, garantizar que las estructuras sean suficientemente grandes para lograr su objetivo puede implicar una presión constante, porque si se crea una estructura demasiado pequeña significa que es posible que se pueda llenar, interrumpiendo las aplicaciones o los subsistemas que hacen uso de ella.

Existe una gran necesidad de dimensionar de forma precisa una estructura en función del uso previsto. Sin embargo, esta tarea es difícil de llevar a cabo porque las cargas de trabajo pueden cambiar con el tiempo, y tener en cuenta sus fluctuaciones no es tarea fácil.

Las colas compartidas IBM MQ utilizan estructuras de lista CF para almacenar mensajes. IBM MQ llama a estructuras CF que contienen mensajes y estructuras de aplicación.

Las estructuras de aplicación se consultan utilizando la información almacenada en objetos CFSTRUCT de IBM MQ. Cuando un mensaje menor de 63 KB se coloca en una cola compartida, el mensaje se almacena por completo en una estructura de aplicación como una entrada de lista única y cero o más elementos de la lista.

Puesto que las colas compartidas de IBM MQ utilizan estructuras de lista, las presiones que se describen también afectan a las colas compartidas. En este caso, el número máximo de mensajes que se puede almacenar en una cola compartida está en función de:

- El tamaño de los mensajes en la cola
- El tamaño máximo de la estructura
- El número de entradas y elementos disponibles en la estructura

Dado que hasta 512 colas compartidas pueden utilizar la misma estructura y compiten eficazmente por las entradas y los elementos, esto complica mucho más las cosas.

Las colas de IBM MQ se utilizan para la transferencia de datos entre aplicaciones, por lo que una situación común es que una aplicación transfiere mensajes a una cola, cuando la aplicación de socio, que debería recibir esos mensajes, no está en ejecución.

Cuando esto sucede, el número de mensajes de la cola de aumenta lo largo del tiempo hasta que se producen una o más de las situaciones siguientes:

- La aplicación de transferencia deja de transferir mensajes.
- La aplicación de obtención empieza a obtener mensajes.

- Los mensajes existentes en la cola empiezan a caducar y se eliminan de la cola.
- La cola alcanza la profundidad máxima en cuyo caso se devuelve un código de razón MQRC\_Q\_FULL a la aplicación de transferencia.
- La estructura que contiene la cola compartida alcanza su tamaño máximo o el CF que contiene la estructura se ejecuta fuera del almacenamiento disponible. En cualquier caso, se devuelve un código de razón MQRC\_STORAGE\_MEDIUM\_FULL a la aplicación que realiza la transferencia.

En los tres últimos casos, la cola está llena. En este punto, la aplicación de transferencia tiene un problema porque no tiene ningún lugar en el que colocar los mensajes. Normalmente, la aplicación de transferencia resuelve este problema recurriendo a una o más de las soluciones siguientes:

- Transfiere reiteradamente el mensaje, opcionalmente con un retraso entre reintentos.
- Transfiere los mensajes a otro lugar, como por ejemplo, una base de datos o un archivo. Se puede acceder a los mensajes más tarde y colocarse en la cola de forma normal.
- Descarte el mensaje si es no persistente.

Sin embargo, para algunas clases de aplicaciones, por ejemplo aquellas con un gran volumen de mensajes entrantes o sin acceso a un sistema de archivos, estas soluciones no son prácticas. Existe una necesidad real de asegurarse de que las colas nunca o casi nunca se llenen en primer lugar y esto es especialmente pertinente en el caso de las colas compartidas.

## SMDS y reglas de descarga

Las reglas de descarga presentadas en IBM WebSphere MQ 7.1 proporcionan una forma de reducir la probabilidad de que una estructura de aplicación se llene.

Cada estructura de aplicación tiene tres reglas asociadas, especificadas mediante tres pares de palabras clave:

- OFFLD1SZ y OFFLD1TH
- OFFLD2SZ y OFFLD2TH
- OFFLD3SZ y OFFLD3TH

Cada regla especifica las condiciones que deben cumplirse para que los datos de mensajes se descarguen en el mecanismo de almacenamiento que está asociado a la estructura de aplicación. Hay dos tipos de mecanismos de almacenamiento disponibles actualmente:

- Db2
- El grupo de conjuntos de datos lineales del método de acceso de almacenamiento virtual (VSAM), que IBM MQ llama conjunto de datos de mensaje compartido (SMDS).

El ejemplo siguiente muestra el mandato MQSC para crear una estructura de aplicación llamada LIST1, utilizando el mandato [DEFINE CFSTRUCT](#).

Esta estructura tiene en vigor las reglas de descarga predeterminadas y utiliza SMDS como mecanismo de descarga. Esto significa que cuando la estructura está llena al 70% (OFFLD1TH), todos los mensajes que son de 32 KB o más (OFFLD1SZ) se descargan en SMDS.

De forma similar, cuando la estructura está llena al 80% (OFFLD2TH), todos los mensajes que tienen 4 KB o más (OFFLD2SZ) se descargan. Cuando la estructura está completa al 90% (OFFLD3TH), todos los mensajes (OFFLD3SZ) se descargan.

```
DEFINE CFSTRUCT(LIST1)
CFLEVEL(5)
OFFLOAD(SMDS)
OFFLD1SZ(32K) OFFLD1TH(70)
OFFLD2SZ(4K) OFFLD2TH(80)
OFFLD3SZ(0K) OFFLD3TH(90)
```

Un mensaje descargado se almacenan en el soporte de descarga y un puntero al mensaje se almacena en la estructura. Mientras que las reglas de descarga reducen la posibilidad de que la estructura se llene, si

se transfieren datos de mensajes en la estructura a medida que se queda sin almacenamiento, algunos datos siguen grabándose en la estructura para cada mensaje. Es decir, el puntero al mensaje descargado.

Además, las reglas de descarga se suministran con un coste de rendimiento. Grabar un mensaje en una estructura es una tarea relativamente rápida que está condicionada por el tiempo que se tarda en enviar la solicitud para la grabación en el CF. La grabación real en la estructura es rápida, sucede a velocidades de almacenamiento real.

Grabar un mensaje en SMDS es una tarea mucho más lenta porque implica grabar en la estructura para el puntero del mensaje y grabar los datos del mensaje en SMDS. Esta segunda operación de grabación se realiza a velocidad DASD y tiene el potencial de añadir latencia. Si Db2 se utiliza como mecanismo de descarga, el coste de rendimiento es mucho mayor.

### *Cómo funciona la memoria de clase de almacenamiento con IBM MQ for z/OS*

Una visión general del uso de la memoria de clase de almacenamiento (SCM) con las colas compartidas de IBM MQ for z/OS.

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z<sup>®</sup> para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

Un recurso de acoplamiento (CF) que está en CFLEVEL 19 o superior, puede tener asignado SCM. A continuación, se pueden configurar las estructuras definidas en este CF, para hacer uso de la SCM con el fin de reducir las posibilidades de que las estructuras se llenen (esto se conoce como condición de estructura completa). Cuando una estructura configurada para hacer uso de la SCM se llena más allá de un punto determinado del sistema, el CF empieza a mover datos desde la estructura a la SCM, lo cual libera espacio en la estructura para datos nuevos.

**Nota:** Dado que la propia SCM se puede llenar, asignar SCM a una estructura sólo reduce la probabilidad de que se dé una condición de estructura completa, pero no elimina completamente la posibilidad de que haya una.

Una estructura se configura para utilizar SCM especificando tanto las palabras clave **SCMALGORITHM** como **SCMMAXSIZE** en la política del gestor de recursos del recurso de acoplamiento (CFRM) que contiene la definición de esa estructura.

Tenga en cuenta que después de que se especifiquen estas palabras clave y de que se aplique la política CFRM, la estructura debe volverse a compilar o desasignar para que las palabras clave entren en vigor.

## **Palabra clave SCMALGORITHM**

Debido a que la velocidad de entrada/salida de SCM es más lenta que la del almacenamiento real, CF utiliza un algoritmo que se adapta al uso previsto de la estructura con el fin de reducir el impacto de grabación en la SCM o de lectura de la SCM.

El algoritmo se configura con la palabra clave **SCMALGORITHM** en la política CFRM de la estructura, utilizando el valor *KEYPRIORITY1*. Tenga en cuenta que debe utilizar el valor *KEYPRIORIDAD1* solamente con las estructuras de lista utilizadas por las colas compartidas de IBM MQ.

El funcionamiento del algoritmo *KEYPRIORIDAD1* presupone que la mayoría de las aplicaciones obtendrán mensajes de una cola compartida por orden de prioridad; es decir, cuando una aplicación obtiene un mensaje, obtiene el mensaje más antiguo con la prioridad más alta.

Cuando una estructura empieza a sobrepasar el umbral del 90% definido por el sistema, el CF empieza a migrar asincrónicamente los mensajes que es menos probable que se obtengan a continuación. Son mensajes con prioridades más bajas que se han transferido recientemente a la cola.

Esta migración asíncrona de mensajes a SCM se conoce como "transferencia previa".

La transferencia previa reduce el coste de rendimiento que supone utilizar la SCM porque reduce la posibilidad de que una aplicación se bloquee durante la aparición de la entrada/salida síncrona en la SCM.

Además de la transferencia previa, el algoritmo *KEYPRIORITY1* también mueve los mensajes de forma asíncrona desde la SCM y a la estructura cuando hay suficiente espacio libre disponible. Para el algoritmo *KEYPRIORITY1*, esto quiere decir, cuando la estructura es menor o igual al 70% de su capacidad.

El acto de mover los mensajes de SCM a la estructura se conoce como "captación previa".

La captación previa reduce la probabilidad de que una aplicación intente obtener un mensaje del que previamente se ha realizado una transferencia a SCM y tenga que esperar mientras el CF traslada de nuevo de forma síncrona el mensaje a la estructura.

## Palabra clave **SCMMAXSIZE**

La palabra clave **SCMMAXSIZE** define la cantidad máxima de SCM que una estructura puede utilizar. Dado que CF asigna la SCM a la estructura cuando es necesario, es posible especificar un valor **SCMMAXSIZE** que sea mayor que la cantidad total de SCM libre disponible. Esto se conoce como "confirmar en exceso".

**Importante:** No confirme nunca en exceso la SCM. Si lo hace, las aplicaciones que dependen de ella no obtendrán el comportamiento previsto. Por ejemplo, las aplicaciones de IBM MQ que utilizan colas compartidas pueden obtener códigos de razón `MQRC_STORAGE_MEDIUM_FULL` inesperados.

El CF utiliza diversas estructuras de datos para realizar un seguimiento del uso de SCM. Estas estructuras de datos residen en el almacenamiento real que se asigna al CF y en consecuencia reducen la cantidad de almacenamiento real que las estructuras pueden utilizar. El almacenamiento utilizado por estas estructuras de datos se conoce como "espacio aumentado".

Cuando una estructura se configura con SCM, se asigna una pequeña cantidad de almacenamiento real del CF a la estructura que se conoce como espacio aumentado fijo. Se asigna incluso si la estructura nunca utiliza realmente ninguna memoria SCM. A medida que los datos de la estructura se almacenan en SCM, se va asignando espacio aumentado dinámico adicional desde el almacenamiento real de repuesto en el CF.

Cuando se eliminan los datos de SCM, el espacio aumentado dinámico se devuelve al CF. El espacio aumentado, ya sea fijo o dinámico nunca se extrae del almacenamiento real que se asigna a una estructura.

Además del almacenamiento aumentado, cuando se configura una estructura para que utilice SCM, la cantidad de almacenamiento de control que esa estructura utiliza aumenta. Esto significa que una estructura de lista configurada con SCM puede contener menos entradas y elementos que una estructura del mismo tamaño sin que se configure SCM.

Para comprender el impacto de SCM en las estructuras nuevas o existentes, utilice la herramienta [CFSizer](#).

Un punto final importante a tener en cuenta es que después de los datos se muevan de la estructura a SCM, y de que se haya utilizado el espacio aumentado dinámico se ha utilizado, la estructura no se puede modificar ni manual ni automáticamente.

Es decir, la cantidad de almacenamiento asignado a la estructura no se puede aumentar o disminuir, la proporción de entradas y elementos que se utiliza en la estructura no se puede cambiar, y así sucesivamente. Para que la estructura se vuelva a alterar, la estructura no debe tener ningún dato almacenado en SCM y no debe hacer uso del almacenamiento aumentado dinámico.

### ¿Por qué utilizar SCM?

El almacenamiento de emergencia y el rendimiento mejorado son dos casos de uso para utilizar SCM con IBM MQ for z/OS.

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z® para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

En esta sección se presenta la teoría detrás de los dos escenarios posibles. Para obtener más detalles sobre cómo configurar los escenarios, consulte:

- [“Almacenamiento de emergencia - configuración básica” en la página 202](#)
- [“Mejor rendimiento – configuración básica” en la página 208](#)

**Importante:** El uso de SCM con estructuras CF no depende de ninguna versión específica de IBM MQ. Sin embargo, el escenario de almacenamiento de emergencia sólo funciona con IBM WebSphere MQ 7.1 y versiones posteriores, porque requiere SMDS y las reglas de descarga.

## Almacenamiento de emergencia

SMDS y la descarga de mensajes se pueden utilizar junto con SCM para reducir la probabilidad de que se devuelva un código de razón MQRC\_STORAGE\_MEDIUM\_FULL a una aplicación de IBM MQ durante una parada prolongada.

### Visión general

Hay una única cola compartida configurada en una estructura de aplicación. La aplicación de transferencia transfiere mensajes a la cola compartida; la aplicación de obtención obtiene mensajes de la cola compartida.

Durante la ejecución normal, está previsto que la profundidad de cola sea próxima a cero, pero un requisito de la empresa indica que el sistema debe ser capaz de tolerar una parada de dos horas de la aplicación de obtención. Esto significa que la cola compartida debe ser capaz de contener dos horas de mensajes de la aplicación de transferencia.

Actualmente, este proceso se consigue mediante las reglas de descarga predeterminadas y SMDS, de modo que el tamaño de la estructura se minimiza al tiempo que se reduce el coste de rendimiento que está asociado con la descarga.

El índice de mensajes que se envía a la cola compartida se espera que sea el doble de corto a medio plazo. Aunque el requisito de que el sistema pueda tolerar una parada de dos horas aún existe, no hay suficiente almacenamiento real disponible en el CF para duplicar el tamaño de la estructura.

Debido a que el CF, que contiene la estructura de aplicación, reside en una máquina zEC12, existe la posibilidad de asociar suficiente SCM con la estructura para almacenar suficientes mensajes de modo que se pueda tolerar una interrupción de dos horas.

Tenga en cuenta lo que ocurre durante un período de tiempo:

1. Inicialmente, el sistema está en estado estable. Tanto la aplicación de transferencia como la de obtención se ejecutan con normalidad y la profundidad de cola está en cero o cerca de cero. El resultado es que la estructura de la aplicación está bastante vacía.
2. En un momento determinado, la aplicación de obtención sufre una anomalía inesperada y se detiene. La aplicación de transferencia continúa transfiriendo mensajes a la cola y la estructura de la aplicación se empieza a llenar.
3. Después de la estructura alcance el 70% de su capacidad, las condiciones de la regla de descarga se cumplen por primera vez y todos los mensajes con un tamaño mayor o igual a 32 KB se descargan en SMDS.

Consulte [“SMDS y reglas de descarga” en la página 196](#) para obtener una visión general de las reglas de descarga.

4. Como los mensajes se siguen transfiriendo a la cola compartida, la estructura se sigue llenando (ya sea debido a que los datos del mensaje se almacenan en la estructura, o como resultado de los punteros a los mensajes descargados que se almacenan en la estructura).

Cuando la estructura alcanza el 80% de su capacidad, la segunda regla de descarga se empieza a aplicar y los mensajes que tienen 4 KB o más se descargan en SMDS.

5. Cuando la estructura está al 90% de su capacidad, todos los mensajes se descargan en SMDS y sólo los punteros de mensaje se colocan en la estructura.

Aproximadamente en este momento, se empieza a ejecutar el algoritmo de transición y empieza a mover los datos de la estructura a SCM. Suponiendo que todos los mensajes en la cola tengan la misma prioridad, se realiza una transferencia previa de los mensajes más recientes.

Dado que ahora todos los mensajes se descargan en SMDS, los datos que se mueven a SCM no son los datos de mensajes reales, sino los punteros a los mensajes en SMDS.

El resultado es que el número de mensajes que se puede almacenar en la combinación de la estructura, y la SCM y SMDS asociados a la estructura, es muy elevado.

**Rendimiento:** Durante esta etapa de la parada, la aplicación de transferencia puede sufrir a una degradación de rendimiento debido a que tiene que grabar en SMDS. En este caso, el uso de SCM no debe ser un factor limitante para la aplicación de transferencia en términos de rendimiento. SCM proporciona espacio adicional para impedir que la estructura se llene.

6. Finalmente, la aplicación de obtención vuelve a estar disponible y la parada termina.

Sin embargo, SCM aún está siendo utilizado por la estructura. La aplicación de obtención empieza a leer mensajes de la cola, obteniendo primero los mensajes más antiguos, de mayor prioridad.

Puesto que estos mensajes se grabaron antes de que la estructura comenzara a llenarse, salen completamente de la parte de almacenamiento real de la estructura.

7. A medida que la estructura se empieza a vaciar, se sitúa por debajo del umbral en el cual la transferencia previa está activa, por consiguiente, la transferencia previa se detiene.
8. El uso de la estructura se reduce por debajo del punto en el cual las reglas de descarga entran en vigor, de modo que los mensajes ya no se descargan en SMDS a menos que tengan más de 63 KB.

En estos momentos aproximadamente, el algoritmo de captación previa empieza a mover datos de SCM a la estructura. Como la aplicación de obtención obtiene mensajes de la cola en el orden previsto por los algoritmos de SCM, los mensajes llegan antes de que la aplicación de obtención los necesite.

El resultado es que la aplicación de obtención nunca necesita esperar a que lleguen los mensajes de forma síncrona desde SCM.

9. A medida que la aplicación de obtención va bajando por la cola, inicia la recuperación de los mensajes que se descargaron en SMDS.
10. Inicialmente, el sistema está de nuevo en un estado estable. No se almacena ningún mensaje en SCM o SMDS, y la profundidad de la cola es casi cero.

## Mejor rendimiento

En este escenario se describe la utilización de SCM para aumentar el número de mensajes que se pueden almacenar en una cola compartida sin que ello afecte al coste de rendimiento de uso de SMDS.

### Descripción

En este escenario, una aplicación de transferencia y de obtención se comunican a través de una cola compartida que se almacena en la estructura de la aplicación.

La aplicación de transferencia tiende a ejecutarse en ráfagas, cuando coloca un gran número de mensajes en un corto periodo de tiempo. A continuación, en un período de tiempo más extendido, no se genera ningún mensaje en absoluto.

La aplicación de obtención procesa de forma secuencial cada mensaje, y realiza un proceso complejo en cada uno. El resultado es que la mayor parte del tiempo la profundidad de cola es cero, excepto cuando la aplicación de transferencia se empieza a ejecutar, entonces la profundidad de cola de mensajes empieza a aumentar a medida que se transfieren mensajes de forma más rápida que lo que se obtienen.

La profundidad de la cola aumenta hasta que la aplicación de transferencia se detiene y la aplicación de obtención dispone de suficiente tiempo para procesar todos los mensajes de la cola.

### Notas:

1. En este escenario, el factor clave es el rendimiento. Los mensajes que se envían a la cola siempre tienen menos de 63 KB y, por lo tanto nunca es necesario que se descarguen en SMDS.
2. La estructura de la aplicación se ha dimensionado para que sea lo suficientemente grande como para contener todos los mensajes que se colocarán en ella al colocar la aplicación en una sola "ráfaga".
3. Las reglas de descarga se deben inhabilitar para que, incluso cuando la estructura empieza a llenarse, los mensajes no se descarguen en SMDS. Esto se debe a que los costes de rendimiento que están asociados a la escritura de mensajes en SMDS y a la lectura de mensajes de SMDS se consideran inaceptables.

Con el tiempo, el número de mensajes que la aplicación de transferencia envía en una ráfaga debe aumentar mediante varios órdenes de magnitud. Como la aplicación de obtención debe procesar cada mensaje de forma secuencial, el número de mensajes en la cola aumenta hasta el punto en que la estructura se llena.

En este punto, la aplicación de transferencia recibe un código de razón (MQRC\_STORAGE\_MEDIUM\_FULL) al transferir un mensaje, y la operación de transferencia no se realiza correctamente. La aplicación de transferencia sólo puede tolerar brevemente períodos cuando no es capaz de transferir mensajes a la cola. Si el periodo de tiempo es demasiado largo, la aplicación finaliza.

Suponiendo que no disponga de tiempo o bien que no disponga de conocimientos, para reescribir la aplicación de transferencia o de obtención, este problema tiene tres soluciones posibles:

1. Aumentar el tamaño de la estructura de aplicación.
2. Añadir reglas de descarga a la estructura de la aplicación para que los mensajes se descarguen en SMDS en cuanto la cola se empieza a llenar.
3. Asociar SCM con la estructura.

La primera solución es rápida de implementar, pero no hay suficiente almacenamiento real disponible en el CF.

La segunda solución también puede ser rápida de implementar, pero el impacto de rendimiento de la descarga en SMDS se considera demasiado importante para utilizar esta opción.

La tercera solución, asociar SCM con la estructura, proporciona una relación de equilibrio entre coste y rendimiento aceptable.

Asociar SCM con una estructura da como resultado un mayor uso de almacenamiento real en el CF debido al almacenamiento ampliado que las operaciones de obtención han utilizado. Sin embargo, la cantidad real de almacenamiento real será menor que la cantidad utilizada en la primera opción.

Otra consideración es el coste de SCM. Sin embargo, este coste es mucho más económico que el almacenamiento real. Estos factores se combinan para hacer que la tercera opción sea más económica que la primera.

Aunque la tercera opción, posiblemente, puede que no se ejecute tan bien como la primera, los algoritmos de captación previa y de transferencia previa utilizados por el CF se pueden combinar para que las diferencias en el rendimiento sean aceptables y en algunos casos, insignificantes.

Sin duda el rendimiento puede ser mucho mejor que utilizar SMDS para descargar mensajes.

Tenga en cuenta lo que ocurre durante un período de tiempo:

1. Inicialmente, la aplicación de obtención está activa y en espera de que los mensajes se entreguen a la cola compartida. La aplicación de transferencia no está activa y la cola compartida está vacía.
2. En un momento determinado, la aplicación de transferencia pasa a estar activa y comienza a transferir un gran número de mensajes a la cola compartida. La aplicación de obtención comienza a obtener los mensajes, pero la profundidad de la cola empieza a aumentar rápidamente debido a que la aplicación de obtención es más lenta que la aplicación de transferencia.

El resultado es que la estructura de la aplicación se empieza a llenar.

3. A medida que el tiempo aumenta, la aplicación de transferencia sigue activa. La estructura de aplicación se llena hasta aproximadamente el 90%.

Aquí es cuando el algoritmo de transferencia previa de SCM empieza a mover mensajes de la estructura a SCM, liberando espacio en la estructura.

Como la aplicación de obtención obtiene primero los mensajes más antiguos y de mayor prioridad de la cola, siempre obtiene mensajes de la estructura y no necesita esperar hasta que los mensajes pasen de forma síncrona de SCM a la estructura.

4. La aplicación de transferencia sigue activa y transfiere mensajes a la cola compartida. Sin embargo, la aplicación nunca recibe un código de razón MQRC\_STORAGE\_MEDIUM\_FULL, porque hay suficiente espacio en SCM para almacenar todos los mensajes que no caben en la estructura.

5. Finalmente, la aplicación de transferencia se detiene porque no tiene más mensajes para transferir.

El algoritmo de transferencia previa se detiene porque el uso de la estructura se sitúa por debajo del 90% y la aplicación de obtención continúa procesando los mensajes en la cola.

6. A medida que la aplicación de obtención empieza a liberar espacio en la estructura, el algoritmo de búsqueda empieza a colocar de nuevos mensajes de SCM en la estructura.

Debido a que la aplicación de obtención procesa los mensajes en el orden esperado por el algoritmo de captación previa, la aplicación de obtención nunca se queda bloqueada en espera de que los datos de mensajes pasen de forma síncrona de SCM a la estructura.

7. Finalmente, la aplicación de obtención procesa todos los mensajes de la cola compartida y espera hasta que el siguiente mensaje esté disponible. La estructura y SCM están vacíos de mensajes.

## Almacenamiento de emergencia - configuración básica

Cómo configurar un escenario básico para el almacenamiento de emergencia en IBM MQ.

### Acerca de esta tarea

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z<sup>®</sup> para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

SMDS y la descarga de mensajes se pueden utilizar junto con SCM para reducir la probabilidad de que se devuelva un código de razón MQRC\_STORAGE\_MEDIUM\_FULL a una aplicación de IBM MQ durante una parada prolongada.

Por ejemplo, la empresa tiene una aplicación que transfiere mensajes a la cola y una aplicación que obtiene mensajes de la cola. Durante la ejecución normal, está previsto que la profundidad de cola sea casi cero, pero un requisito de la empresa indica que el sistema debe ser capaz de tolerar una parada de dos horas de la aplicación que obtiene los mensajes.

Esto significa que la cola compartida que se está utilizando debe ser capaz de contener dos horas de mensajes de la aplicación de transferencia. Actualmente, puede lograrlo utilizando las reglas de descarga predeterminadas y SMDS.

Está previsto que el índice de mensajes que se envían a la cola compartida se duplique de corto a medio plazo. Aunque el requisito de que el sistema pueda tolerar una parada de dos horas aún existe, no hay suficiente almacenamiento real disponible en el CF para duplicar el tamaño de la estructura. Debido a que el CF que contiene la estructura de aplicación reside en una máquina zEC12, tiene la posibilidad de asociar suficiente SCM con la estructura para almacenar suficientes mensajes de modo que se pueda tolerar una interrupción de dos horas.

Este escenario inicial utiliza lo siguiente:

- Un grupo de compartición de colas, IBM1 que contiene un único gestor de colas, CSQ3. Además de la estructura de administración, el grupo de compartición de colas definidas en una estructura de aplicación individual, SCEN1.
- Un recurso de acoplamiento (CF) CF01, en el que la estructura de la aplicación SCEN1 se almacena como la estructura IBM1SCEN1. Esta estructura tiene un tamaño máximo de 1 GB.

- Una única cola compartida, SCEN1.Q que la estructura de la aplicación utiliza.

Esta configuración se muestra en la [Figura 61 en la página 203](#).

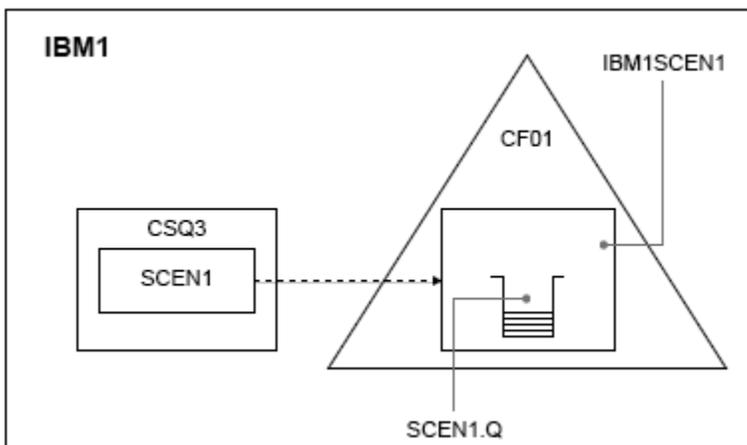


Figura 61. Configuración básica

Además, suponga que el gestor de colas CSQ3 ya es el único miembro del grupo de compartición de colas de IBM1.

Debe añadir la definición de la estructura IBM1SCEN1 a la política del gestor de recursos del recurso de acoplamiento (CFRM). Para mayor simplificación, la estructura está definida para que se pueda crear con tan sólo un recurso de acoplamiento individual, CF01, especificando PREFLIST(CF01).



**Atención:** Para permitir una alta disponibilidad en el sistema de producción, debe incluir al menos dos CF en PREFLIST para cualquier estructura utilizada por IBM MQ.

## Procedimiento

1. Renueve la política CFRM, emitiendo el mandato siguiente:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN1
```

Ejemplo de política CFRM de la estructura IBM1SCEN1:

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
```

2. Verifique si la estructura se ha creado correctamente, utilizando el mandato siguiente:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

En este punto, la estructura no se ha asignado, mostrada por la línea STATUS, al grupo de compartición de colas.

3. Configure IBM MQ para que utilice la estructura definida en la política CFRM.

- a. Utilice el mandato `DEFINE CFSTRUCT`, con el nombre de estructura de SCEN1 para crear un objeto CFSTRUCT de IBM MQ:

```
DEFINE CFSTRUCT(SCEN1)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 1')
RECOVER(NO)
```

```
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

- b. Valide la estructura, utilizando el mandato [DISPLAY CFSTRUCT](#).
- c. Defina la cola compartida SCEN1.Q, para utilizar la estructura SCEN1 utilizando el siguiente mandato MQSC:

```
DEFINE QLOCAL(SCEN1.Q) QSGDISP(SHARED) CFSTRUCT(SCEN1) MAXDEPTH(999999999)
```

4. Utilice IBM MQ Explorer para transferir un único mensaje a la cola SCEN1.Q y volverlo a quitar.
5. Emita el mandato siguiente para comprobar que la estructura está ahora asignada:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Registre la salida del mandato, que la línea STATUS muestra ALLOCATED.

## Resultados

Ha creado la configuración básica. Ahora, puede obtener una idea del rendimiento de la configuración de línea base utilizando cualquier método que seleccione.

## Qué hacer a continuación

Añada SMDS y SCM a la estructura inicial

### Conceptos relacionados

[“Uso de la memoria de clase de almacenamiento \(SCM\) con colas compartidas” en la página 194](#)

El uso de la memoria de clase de almacenamiento (SCM) puede ser ventajoso cuando se utiliza con colas compartidas de IBM MQ for z/OS.

 *Adición de SMDS y SCM a la estructura inicial*

Cómo añadir SMDS y SCM para el almacenamiento de urgencia en IBM MQ.

## Acerca de esta tarea

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z® para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

Esta parte de la tarea utiliza la configuración básica que se describe en [“Almacenamiento de emergencia - configuración básica” en la página 202](#). El escenario describe cómo añadir conjuntos de datos de mensaje compartidos (SMDS) y luego SCM a la estructura inicial.

Esta configuración final se ilustra en la [Figura 62 en la página 205](#).

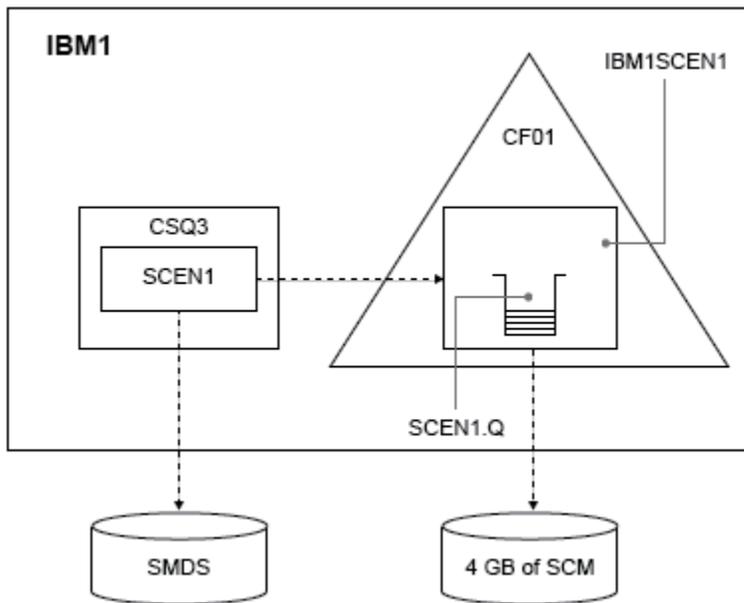


Figura 62. Configuración agregando SMDS y SCM para el almacenamiento de emergencia

## Procedimiento

1. Cree el conjunto de datos SMDS que la estructura de la aplicación SCEN1 utiliza editando el JCL de ejemplo, **CSQ4SMDS** tal como se muestra a continuación:

```
//CSQ4SMDS JOB NOTIFY=&SYSUID
//*
//* Allocate SMDS
//*
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER          -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS) -
MEGABYTES(5000 3000) -
LINEAR                 -
SHAREOPTIONS(2 3) ) -
DATA                   -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS.DATA) )
/*
//*
//* Format the SMDS
//*
//FORM EXEC PGM=CSQJUFMT,COND=(0,NE),REGION=0M
//STEPLIB DD DSN=MQ800.SCSQANLE,DISP=SHR
// DD DSN=MQ800.SCSQAUTH,DISP=SHR
//SYSUT1 DD DISP=OLD,DSN=CSQSMDS.SCEN1.CSQ3.SMDS
//SYSPRINT DD SYSOUT=*
```

2. Emita el mandato **ALTER CFSTRUCT** para cambiar la estructura de aplicación SCEN1, para utilizar SMDS para la descarga, implementando las reglas de descarga predeterminadas:

```
ALTER CFSTRUCT(SCEN1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ(4K) OFFLD3SZ(0K)
DSGROUP('CSQSMDS.SCEN1.*.SMDS') DSBLOCK(1M)
```

Tenga en cuenta lo siguiente:

- Puesto que SCEN1.Q es la única cola compartida en la estructura de la aplicación SCEN1, el valor de **DSBLOCK** se ha establecido en 1M, el mayor valor posible. Este sería el valor más eficaz para nuestro escenario.
- Dado que los mensajes enviados por la aplicación que realiza la transferencia son 30 de KB, la descarga en SMDS no se iniciará hasta que se cumpla la segunda regla de descarga, cuando la estructura esté llena al 80%.

3. Ejecute de nuevo la aplicación de prueba.

Tenga en cuenta que el almacenamiento de los mensajes en la cola ha aumentado.

4. Añada 4 GB de SCM a la estructura de IBM1SCEN1 llevando a cabo el procedimiento siguiente:

a) Compruebe cuánta SCM se ha instalado y asignado a CF01, emitiendo el siguiente mandato:

```
D CF,CFNAME=CF01
```

b) Compruebe las figuras STORAGE-CLASS MEMORY en la sección STORAGE CONFIGURATION de la salida visualizada para ver el almacenamiento disponible.

c) Actualice la política CFRM con las palabras clave SCMMAXSIZE y SCMALGORITHM tal como se muestra a continuación:

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
SCMMAXSIZE(4G)
SCMALGORITHM(KEYPRIORITY1)
```

5. Active la política CFRM, emitiendo el mandato siguiente:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
```

6. Vuelva a crear la estructura IBM1SCEN1.

Debe llevar a cabo este procedimiento porque la estructura se asignó cuando realizó los cambios anteriores.

Emita el mandato siguiente para volver a crear la estructura:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN1
```

## Resultados

Ha añadido satisfactoriamente SCM a la configuración.

## Qué hacer a continuación

Optimice el rendimiento del sistema. Consulte [“Optimización del uso de memoria de la clase de almacenamiento”](#) en la página 206 para obtener más información.

 *Optimización del uso de memoria de la clase de almacenamiento*  
Cómo mejorar el uso de la memoria de clase de almacenamiento (SCM).

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z® para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

Ejecute el siguiente mandato:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

A medida que la estructura ya estaba llena con datos de mensaje, debido a las pruebas anteriores, parte de la recompilación implicaba colocar en etapas previas algunos de los mensajes de la estructura en SCM. Este proceso se inició mediante el mandato anterior.

La salida de este mandato genera, por ejemplo:

```
ACTIVE STRUCTURE
-----
ALLOCATION TIME: 06/17/2014 09:28:50
CFNAME : CF01
COUPLING FACILITY: 002827.IBM.02.00000000B8D7
PARTITION: 3B CPCID: 00
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
ACTUAL SIZE: 1024 M 1024 M 100
AUGMENTED SPACE: 3 M 142 M 2
STORAGE-CLASS MEMORY: 88 M 4096 M 2
ENTRIES: 120120 1089536 11
ELEMENTS: 240240 15664556 1
SPACE USAGE IN-USE TOTAL %
ENTRIES: 84921 219439 38
ELEMENTS: 2707678 3149050 85
EMCS: 2 282044 0
LOCKS: 1024
SCMHIGHTHRESHOLD : 90
SCMLOWTHRESHOLD : 70
ACTUAL SUBNOTIFYDELAY: 5000
PHYSICAL VERSION: CD5186A0 2BD8B85C
LOGICAL VERSION: CD515C50 CE2ED258
SYSTEM-MANAGED PROCESS LEVEL: 9
XCF GRPNAME : IXCLO053
DISPOSITION : KEEP
ACCESS TIME : NOLIMIT
MAX CONNECTIONS: 32
# CONNECTIONS : 1
CONNECTION NAME ID VERSION SYSNAME JOBNAME ASID STATE
-----
CSQEIBM1CSQ301 01 00010059 SC61 CSQ3MSTR 0091 ACTIVE
```

Tenga en cuenta lo siguiente en la salida del mandato:

- Esa STORAGE\_CLASS MEMORY proporciona confirmación de que se ha añadido un **MAXIMUM** de 4096 MB de SCM a la estructura.
- La cifra ALLOCATED para el importe de STORAGE-CLASS MEMORY que se ha utilizado para la transferencia previa. Ahora, hay espacio libre en la estructura donde antes no había nada antes de que se añadiera SCM.
- La cantidad de AUGMENTED SPACE que se utiliza para realizar un seguimiento del uso de SCM.
- El punto en el que el algoritmo de transferencia previa empieza a mover datos de la estructura a SCM es cuando la estructura está llena al 90%. Esto se indica mediante la propiedad **SCMHIGHTHRESHOLD** no configurable.
- El punto por debajo del cual el algoritmo de captación previa empieza a mover datos de SCM a la estructura es cuando la estructura está al 70% de su capacidad. Esto se indica mediante la propiedad **SCMLOWTHRESHOLD** no configurable.

Ahora puede probar formas de optimizar el uso de SCM. Tenga en cuenta lo siguiente:

- Después de se utilice SCM para almacenar mensajes, no se puede modificar la estructura hasta que haya eliminado todos los datos de SCM.

En este caso, esto significa que la proporción entre entradas y elementos se congela en el valor que había cuando se utilizó por primera vez SCM. Debe asegurarse cuidadosamente de que la estructura se encuentra en el estado deseado, antes de que el algoritmo de transición empiece a mover datos a SCM.

- ¿El tamaño de la estructura actual es el correcto antes de utilizar SCM?

Por ejemplo, ¿ha aumentado **INITSIZE** desde 512 MB hasta un SIZE de 1 GB?

Si no lo hace, es posible que aunque haya habilitado la estructura para la modificación automática, el algoritmo de transición vaya a empezar a mover datos a SCM antes de que la modificación tenga la oportunidad de iniciarse. El resultado es que la estructura está congelada utilizando 512 MB de almacenamiento real.

- ¿La proporción de entradas y elementos es correcta antes de utilizar SCM?

El objetivo de este escenario es aumentar el número de punteros de mensajes descargados que se pueden almacenar en la estructura y SCM en su totalidad, así como mantener el número máximo de

mensajes enteros que sea posible en el almacenamiento de la estructura. El acceso a estos mensajes es más rápido que el acceso a los mensajes en SMDS.

Por lo tanto, debe disponer de una estructura que empiece con una proporción de entadas y elementos que sea buena para almacenar mensajes y, a continuación, realice la transición a una proporción que sea buena para almacenar punteros de mensajes antes de que el algoritmo de transferencia previa se inicie por primera vez. Esta transición se puede lograr, en parte, haciendo uso de las reglas de descarga de IBM MQ.

Cambie las reglas de descarga emitiendo el siguiente mandato:

```
ALTER CFSTRUCT(SCEN1) OFFLD1SZ(0K)
```

Puede que tenga que llevar a cabo varias ejecuciones para optimizar las proporciones de entradas y elementos.

La tabla siguiente muestra posibles mejoras en el número de mensajes transferidos a la cola durante las diferentes fases del escenario de almacenamiento de emergencia.

Descripción de la prueba	Número de mensajes	Tiempo para llenar la cola (segundos)
Configuración básica	27.850	3.2
SMDS con reglas de descarga predeterminada	205.000	158
SCM con reglas de descarga predeterminadas	828.610	469
SCM con reglas de descarga ajustadas	1.135.775	679

La última fila de la tabla muestra que ajustar las reglas de descarga ha tenido el efecto necesario.

Es necesario que examine el sistema para ver si puede mejorar de alguna manera estas cifras. Por ejemplo, podría quedarse sin almacenamiento SMDS disponible. Si puede asignar más almacenamiento SMDS, debe ser capaz de aumentar el número de mensajes en la cola de forma bastante significativa.

### Mejor rendimiento – configuración básica

Cómo configurar un escenario básico para mejorar el rendimiento utilizando colas compartidas en IBM MQ.

## Acerca de esta tarea

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z<sup>®</sup> para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

En este escenario se describe la utilización de SCM para aumentar el número de mensajes que se pueden almacenar en una cola compartida sin que ello afecte al coste de rendimiento de uso de SMDS.

Este escenario inicial es muy similar al que se utiliza en el almacenamiento de emergencia y utiliza lo siguiente:

- Un grupo de compartición de colas, IBM1 que contiene un único gestor de colas, CSQ3. Además de la estructura de administración, el grupo de compartición de colas definido en una estructura de aplicación individual, SCEN2.
- Un recurso de acoplamiento (CF) CF01, en el que la estructura de la aplicación SCEN2 se almacena como la estructura IBM1SCEN2. Esta estructura tiene un tamaño máximo de 2 GB.

- La cola compartida única, SCEN2.Q, que está configurada para utilizar la estructura de la aplicación. Esta configuración se muestra en la [Figura 63 en la página 209](#).

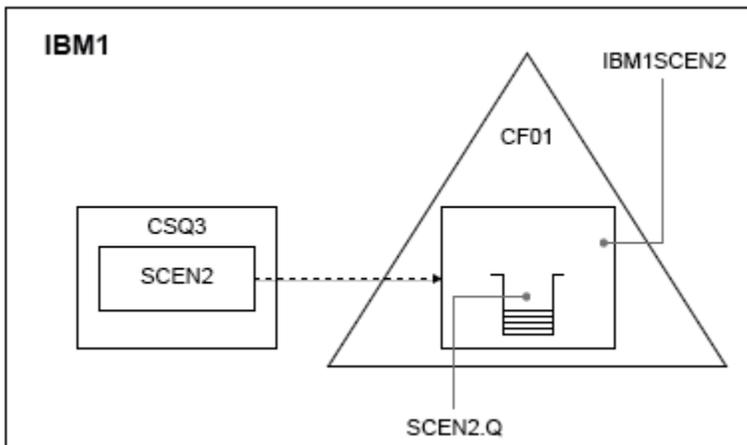


Figura 63. Configuración básica

Además, suponga que el gestor de colas CSQ3 ya es el único miembro del grupo de compartición de colas de IBM1.

Debe añadir la definición de la estructura IBM1SCEN2 a la política del gestor de recursos del recurso de acoplamiento (CFRM). Para mayor simplificación, la estructura está definida para que se pueda crear con tan sólo un recurso de acoplamiento individual, CF01, especificando PREFLIST (CF01).

Política CFRM de ejemplo para la estructura IBM1SCEN2:

```
STRUCTURE
NAME (IBM1SCEN2)
SIZE (2048M)
INITSIZE (2048M)
ALLOWAUTOALT (YES)
FULLTHRESHOLD (85)
PREFLIST (CF01)
ALLOWREALLOCATE (YES)
DUPLEX (DISABLED)
ENFORCEORDER (NO)
```

Las palabras clave **INITSIZE** y **SIZE** tienen el valor 2048M de modo que la estructura no se puede redimensionar.

## Procedimiento

1. Renueve la política CFRM, emitiendo el mandato siguiente:

```
SETXCF START ,POLICY ,TYPE=CFRM ,POLNAME=IBM1SCEN2
```

2. Verifique si la estructura se ha creado correctamente, utilizando el mandato siguiente:

```
D XCF ,STR ,STRNAME=IBM1SCEN2
```

La emisión del mandato anterior proporciona la siguiente salida:

```
RESPONSE=SC61
IXC360I 07.58.51 DISPLAY XCF 581
STRNAME: IBM1SCEN2
STATUS: NOT ALLOCATED
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
```

```
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

```
EVENT MANAGEMENT: MESSAGE-BASED   MANAGER SYSTEM NAME: SC53
MANAGEMENT LEVEL : 01050107
```

En este punto, la estructura no se ha asignado, mostrada por la línea STATUS, al grupo de compartición de colas.

3. Configure IBM MQ para que utilice la estructura definida en la política CFRM.

- a. Utilice el mandato `DEFINE CFSTRUCT`, con el nombre de estructura de SCEN2 para crear un objeto CFSTRUCT de IBM MQ.

```
DEFINE CFSTRUCT(SCEN2)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 2')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

- b. Compruebe la estructura, utilizando el mandato `DISPLAY CFSTRUCT`.

- c. Defina la cola compartida SCEN2 . Q, para utilizar la estructura SCEN2 utilizando el siguiente mandato MQSC:

```
DEFINE QLOCAL(SCEN2.Q) QSGDISP(SHARED) CFSTRUCT(SCEN2) MAXDEPTH(99999999)
```

4. Utilice IBM MQ Explorer para colocar un único mensaje en la cola SCEN2 . Q y volverlo a quitar.

5. Emita el mandato siguiente para comprobar que la estructura está ahora asignada:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Revise la salida del mandato, una parte del cual se muestra y asegúrese de que la línea STATUS muestra ALLOCATED.

```
RESPONSE=SC61
IXC360I 08.31.27 DISPLAY XCF 703
STRNAME: IBM1SCEN2
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

Además, tome nota de los valores de los campos de la sección SPACE USAGE:

- ENTRIES
- ELEMENTS
- EMCS
- LOCKS

A continuación se muestra un ejemplo de los valores:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	344686	345242	99

ELEMENTS:	6548455	6548467	99
EMCS:	2	780318	0
LOCKS:	1024		

## Resultados

Ha creado la configuración básica. Ahora, puede obtener una idea del rendimiento de la configuración de línea base utilizando cualquier método que seleccione.

## Qué hacer a continuación

Debe probar el escenario básico. Como ejemplo, puede utilizar las tres aplicaciones siguientes, iniciando las aplicaciones en el orden mostrado y ejecutándolas simultáneamente.

1. Utilice una aplicación PCF para solicitar el valor de profundidad actual (**CURDEPTH**) para SCEN2 . Q cada cinco segundos. La salida se puede utilizar para trazar la profundidad de la cola a lo largo del tiempo.
2. Una aplicación de obtención de hebra única obtiene repetidamente mensajes de SCEN2 . Q, utilizando un solicitud de obtención (get) con una espera infinita. Para simular el proceso de los mensajes que se han eliminado, la aplicación de obtención se coloca en pausa durante cuatro milisegundos para cada cuatro mensajes de diez aplicaciones que elimina.
3. Una aplicación de transferencia de hebra única transfiere un total de un millón 4 KB mensajes no persistentes a SCEN2 . Q. Esta aplicación no se coloca en pausa entre la transferencia de cada mensaje por lo que los mensajes se transfieren a SCEN2 . Q de modo más rápido que lo que la aplicación de obtención puede obtenerlos.

El resultado es que cuando se ejecuta la aplicación de transferencia, la profundidad de SCEN2 . Q aumenta.

Cuando la estructura IBM1SCEN2 está llena y la aplicación de transferencia recibe un código de razón MQRC\_STORAGE\_MEDIUM\_FULL, la aplicación de transferencia se pone en suspenso durante cinco segundos antes de intentar transferir el siguiente mensaje a la cola.

Puede trazar los resultados de la aplicación CURDEPTH durante un período de tiempo. Se obtiene alguna forma de salida de onda en forma de dientes de sierra cuando la aplicación de transferencia se coloca en pausa para permitir que la cola se vacíe parcialmente.

Vaya a “Adición de SCM a la estructura inicial” en la [página 211](#).

### Conceptos relacionados

“Uso de la memoria de clase de almacenamiento (SCM) con colas compartidas” en la [página 194](#)

El uso de la memoria de clase de almacenamiento (SCM) puede ser ventajoso cuando se utiliza con colas compartidas de IBM MQ for z/OS.

 *Adición de SCM a la estructura inicial*

Cómo añadir SCM para mejorar el rendimiento en IBM MQ.

## Acerca de esta tarea

**Importante:** IBM z16 está previsto que sea la última generación de IBM Z<sup>®</sup> para dar soporte al uso de memoria flash virtual (también conocida como memoria de clase de almacenamiento o SCM) para imágenes del recurso de acoplamiento. Para obtener más información, consulte: [IBM Z e IBM LinuxONE 4Q 2023 Statements of Direction](#).

Como alternativa, debe utilizar estructuras más grandes o descargar mensajes en SMDS.

Esta parte de la tarea utiliza la configuración básica que se describe en “Mejor rendimiento – configuración básica” en la [página 208](#). El escenario describe cómo añadir SCM a la estructura inicial.

Esta configuración final se ilustra en la [Figura 64 en la página 212](#).

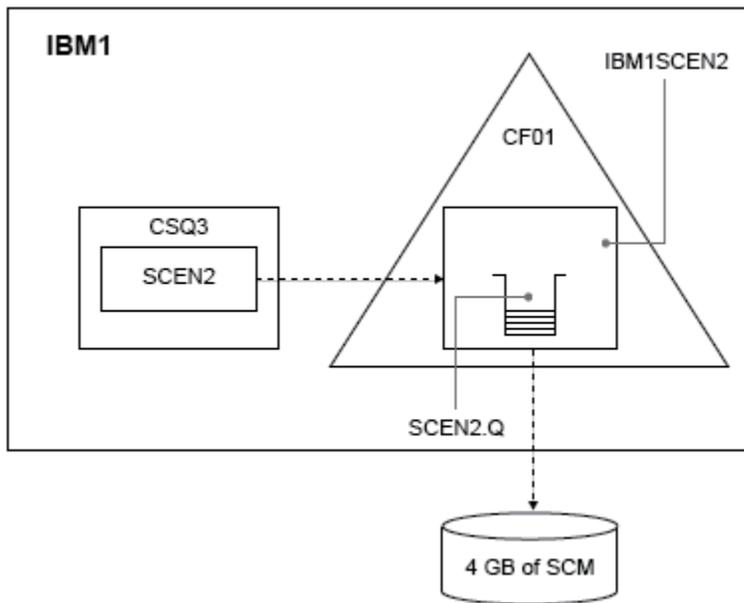


Figura 64. Configuración añadiendo SCM para mejorar el rendimiento

## Procedimiento

1. Añada 4 GB de SCM a la estructura IBM1SCEN2 llevando a cabo el procedimiento siguiente:

- a) Compruebe cuánta SCM se ha instalado y asignado a CF01, emitiendo el siguiente mandato:

```
D CF,CFNAME=CF01
```

- b) Compruebe las figuras STORAGE-CLASS MEMORY en la sección STORAGE CONFIGURATION de la salida visualizada para ver el almacenamiento disponible.
- c) Actualice la política CFRM con las palabras clave SCMMAXSIZE y SCMALGORITHM tal como se muestra a continuación:

```
STRUCTURE
NAME(IBM1SCEN2)
SIZE(2048M)
INITSIZE(2048M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
SCMMAXSIZE(4G)
SCMALGORITHM(KEYPRIORITY1)
```

2. Active la política CFRM, emitiendo el mandato siguiente:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
```

3. Vuelva a crear la estructura IBM1SCEN2.

Debe llevar a cabo este procedimiento porque la estructura se asignó cuando realizó los cambios anteriores.

Emita el mandato siguiente para volver a crear la estructura:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN2
```

4. Emita el mandato siguiente para confirmar la nueva configuración de la estructura:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Revise la salida del mandato, una parte de la cual es la siguiente:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	33	342684	0
ELEMENTS:	48	6503697	0
EMCS:	2	575600	0
LOCKS:		1024	

## Resultados

Calcule el cambio en el uso de almacenamiento real con el aumento en el almacenamiento de control necesario para utilizar SCM.

- Antes de que se añada SCM a la estructura, la estructura contiene estos totales tal y como se muestra en la [“Mejor rendimiento – configuración básica”](#) en la página 208:
  - 345.242 entradas
  - 6.548.467 elementos
  - 780.318 EMCS
- Después de que se añada SCM a la estructura, la estructura contiene estos totales:
  - 342.684 entradas
  - 6.503.697 elementos
  - 575.600 EMCS

Con estas cifras, después de que se añadiera la SCM, la estructura se reduce de tamaño en:

- 2558 entradas
- 44.770 elementos
- 204.718 EMCS

La cantidad de almacenamiento de la estructura que se utiliza para gestionar SCM es la siguiente para una estructura de 2 GB con 4 GB de SCM asignados:

$$(2558 + 44,770 + 204,718) * 256 = 61.5 \text{ MB}$$

Tenga en cuenta que si se añade más SCM probablemente sólo se conseguirá una reducción marginal del tamaño de la estructura, ya que la cantidad de almacenamiento de control que se utiliza para realizar un seguimiento de SCM aumenta, igual que el tamaño de la estructura y la cantidad de SCM asignada aumenta.

## Qué hacer a continuación

Repita las pruebas que se describen en la sección final de [“Mejor rendimiento – configuración básica”](#) en la página 208.

Puede trazar los resultados de la aplicación revisada durante un período de tiempo. Comparando el gráfico con el que se obtuvo anteriormente, ahora obtendrá una salida sin una ola en forma dentada ya que la aplicación de transferencia no tiene que esperar a que la cola se vacíe parcialmente.

Para obtener más información, consulte [MP16: WebSphere MQ for z/OS -Planificación y ajuste de capacidad](#).

## Gestión de colas distribuidas y grupos de compartición de colas

La gestión de colas distribuidas y los grupos de compartición de colas son dos técnicas que puede utilizar para aumentar la disponibilidad de los sistemas de aplicación. Utilice este tema para encontrar más información sobre estas técnicas.

Para complementar la alta disponibilidad de mensajes en colas compartidas, el componente de gestión de colas distribuidas de IBM MQ contiene funciones adicionales que proporcionan:

- Mayor disponibilidad a la red.
- Mayor capacidad para conexiones de red de entrada al grupo de compartición de colas.

La [Figura 65 en la página 214](#) ilustra la gestión de colas distribuidas y los grupos de compartición de colas. Muestra dos gestores de colas dentro de un sysplex, que pertenecen al mismo grupo de compartición de colas. Ambos pueden acceder a la cola compartida SQ1. Los gestores de colas de la red (en AIX y Windows por ejemplo) pueden colocar mensajes a esta cola mediante el iniciador de canal de cada gestor de colas. Las aplicaciones clonadas en ambos gestores de colas dan servicio a la cola.

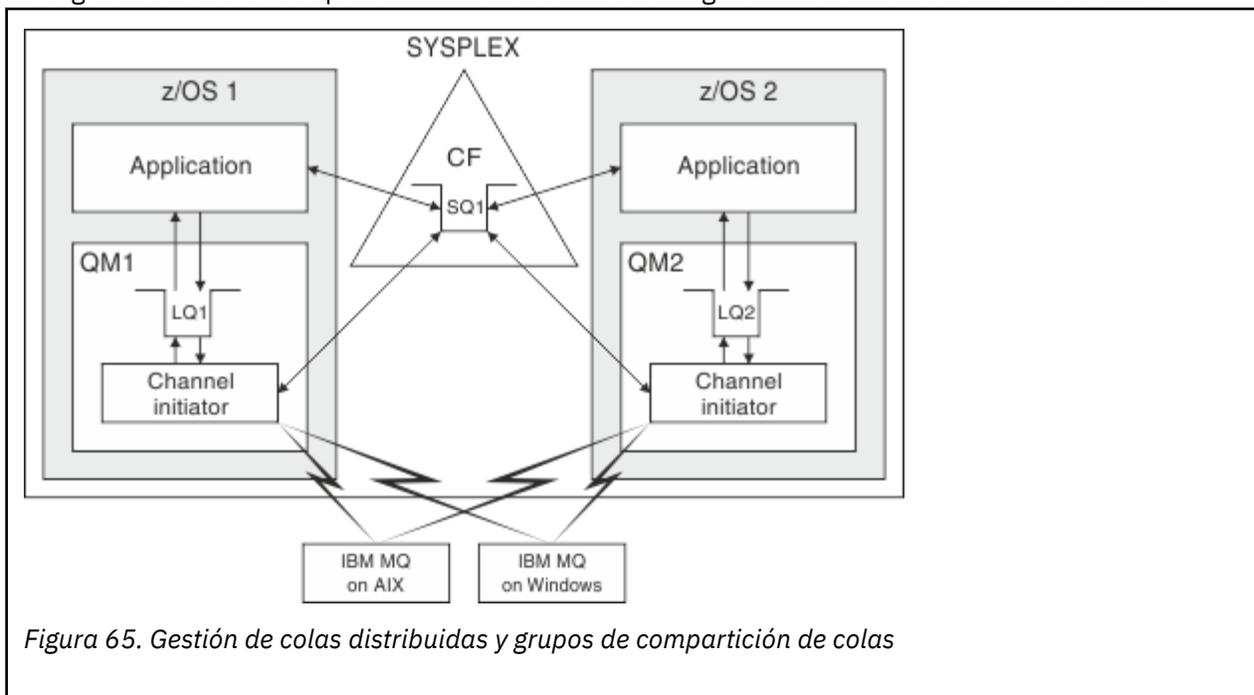


Figura 65. Gestión de colas distribuidas y grupos de compartición de colas

### Conceptos relacionados

“Canales compartidos” en la [página 214](#)

Utilice este tema para comprender los conceptos de canales compartidos y su uso con IBM MQ for z/OS.

“Transferencia a colas entre grupos” en la [página 219](#)

En esta sección se describe la gestión de colas dentro del grupo, una función de IBM MQ for z/OS exclusiva de la plataforma z/OS. Esta función sólo está disponible para los gestores de colas definidos en un grupo de compartición de colas.

“clústeres y grupos de compartición de colas” en la [página 216](#)

Utilice este tema si desea conocer cómo puede utilizar los grupos de compartición de colas con clústeres.

### ► z/OS Canales compartidos

Utilice este tema para comprender los conceptos de canales compartidos y su uso con IBM MQ for z/OS.

Un número de productos de red proporcionan un mecanismo para ocultar anomalías del servidor de la red, o para equilibrar las solicitudes de red de entrada a través de un conjunto de servidores elegibles. Los productos de red realizan un *puerto genérico* disponible para solicitudes de conexión de red de entrada, y la solicitud de entrada puede satisfacerse conectándose a uno de los servidores elegibles.

Estos productos de red incluyen:

- recursos genéricos de VTAM
- distribuidor de sysplex

El iniciador de canal aprovecha estos productos para utilizar las capacidades de las colas compartidas.

Hay dos tipos de canales compartidos, el *canal de entrada compartido* y el *canal de salida compartido*.

- [Canales de entrada compartidos](#)

- [Canales de salida compartidos](#)

Si desea más información sobre canales consulte

- [Resumen de canales compartidos](#)
- [Estado de canal compartido](#)

## **Canales de entrada compartidos**

Cada iniciador de canal del grupo de compartición de colas inicia una tarea de escucha adicional para escuchar en un *puerto genérico*. Una de las tecnologías de soporte (VTAM, TCP/IP) pone a disposición el puerto genérico a la red. Las solicitudes de conexión de red de entrada al puerto genérico son asignadas por la tecnología de red, a cualquiera de los escuchas del grupo de compartición de colas (QSG) que estén escuchando en el puerto genérico.

Puede iniciar un canal en el iniciador de canal al que se direcciona la conexión de entrada si el iniciador de canal tiene acceso a una definición de canal para un canal con ese nombre. Puede definir que una definición de canal sea privada en un gestor de colas o que se almacene en un repositorio compartido y esté disponible en cualquier lugar (una definición global). Esto significa que puede hacer que una definición de canal esté disponible en cualquier iniciador de canal del grupo de compartición de colas definiéndolo como una definición global.

Existe una diferencia adicional cuando se inicia un canal a través del puerto genérico; la sincronización de canal se realiza con el grupo de compartición de colas y no con un gestor de colas individual. Por ejemplo, imagine un gestor de colas remoto que inicia un canal a través de un puerto genérico. Cuando se inicie el canal por primera vez, es posible que se inicie en el gestor de colas QM1 y flujo de mensajes. Si el canal se detiene y se reinicia en el gestor de colas QM2, la información sobre el número de mensajes que han fluido seguirá siendo correcta porque la sincronización se realiza con el grupo de compartición de colas.

Puede utilizar un canal de entrada iniciado a través del puerto genérico para colocar mensajes en cualquier cola. El gestor de colas remoto no sabe si la cola de destino es compartida o no. Si la cola de destino es una cola compartida, el gestor de colas remoto se conecta a través de cualquier iniciador de canal disponible de forma que consiga equilibrio de carga y los mensajes se colocan en la cola compartida.

Si la cola de destino es una cola privada, los mensajes se colocan en la cola privada propiedad del gestor de colas al que esté conectada la instancia actual del canal. En este entorno, conocido como *colas locales duplicadas*, cada gestor de colas debe tener definido el mismo conjunto de colas privadas.

## **Configuración de canales SVRCONN para un grupo de compartición de colas**

La configuración óptima de canales SVRCONN en un grupo de uso compartido sería establecer escuchas privados en cada CHINIT que utilice un número de puerto diferente al de los canales punto a punto. Estos puertos de escucha, a continuación, se utilizan como los recursos "de fondo" para un nuevo mecanismo de distribución de la carga de trabajo como el distribuidor Sysplex utilizando direcciones IP virtuales (VIPA). La dirección VIPA externa se utiliza como la dirección de destino para las definiciones CLNTCONN en la red. El canal SVRCONN se puede definir con QSGDISP(GROUP) de forma que la misma definición esté disponible para todos los gestores de colas en el QSG. Esta configuración evita utilizar un escucha compartido y, por lo tanto, reduce el efecto de rendimiento del grupo de compartición de colas que mantiene el estado de canal compartido, que no es necesario para los canales de cliente/servidor.

## **Canales de salida compartidos**

Se considera que un canal de salida es un canal compartido si adopta mensajes de una cola de transmisión compartida. Si es compartido, conserva información de sincronización a nivel de grupo de compartición de colas. Esto significa que el canal puede reiniciarse en un gestor de colas diferente y una instancia del iniciador de canal dentro del grupo de compartición de colas si falla el subsistema de

comunicaciones, iniciador de canal o el gestor de colas. El hecho de reiniciar los canales fallidos de esta forma constituye una característica de los canales compartidos denominados *recuperación de canal igual*.

### **Equilibrio de carga de trabajo para canales de salida compartidos**

Puede elegir un canal de salida compartido para que se inicie en cualquier iniciador de canal dentro del grupo de compartición de colas, si no ha especificado que desea que se reinicie en un iniciador de canal en concreto. El iniciador de canal seleccionado por IBM MQ se determina utilizando los criterios siguientes:

- ¿El subsistema de comunicaciones necesario actualmente está disponible para el iniciador de canal?
- ¿Es una conexión de Db2 disponible para el iniciador de canal?
- ¿Qué iniciador de canal tiene la carga de trabajo actual más baja? La carga de trabajo incluye canales que están activos y reintentando.

### **Resumen de canal compartido**

Los canales compartidos difieren de los canales privados de la siguiente manera:

#### **Canal privado**

Vinculado a un único iniciador de canal.

- El canal de salida utiliza una cola de transmisión.
- El canal de entrada iniciado a través del puerto local.
- Información de sincronización retenida en la cola SYSTEM.CHANNEL.SYNCQ.

#### **Canal compartido**

Equilibrio de carga de trabajo con alta disponibilidad.

- El canal de salida utiliza una cola de transmisión compartida.
- El canal de entrada se ha iniciado a través de un puerto genérico.
- Información de sincronización retenida en la cola SYSTEM.QSG.CHANNEL.SYNCQ.

Puede especificar si un canal es privado o compartido durante el inicio del canal mediante las opciones CHLDISP con el mandato `START CHANNEL`. Un canal compartido se puede iniciar realizando un lanzamiento igual al de un canal privado. Sin embargo, cuando se inicia un canal compartido, IBM MQ realiza el equilibrado de la carga de trabajo e inicia el canal en el iniciador de canal más adecuado dentro del grupo de compartición de colas. (Si es necesario, puede especificar que un canal compartido se inicie en un iniciador de canal concreto.)

### **Estado de canal compartido**

Los iniciadores de canal en un grupo de compartición de colas mantienen una tabla de estado de canal compartido en Db2. Esta tabla registra qué canales están activos en qué iniciadores de canal. La tabla de estado de canal compartido se utiliza si se produce una anomalía en el iniciador de canal o en el sistema de comunicaciones. Indica qué canales deben reiniciarse en un iniciador de canal diferente en el grupo de compartición de colas.

### **clústeres y grupos de compartición de colas**

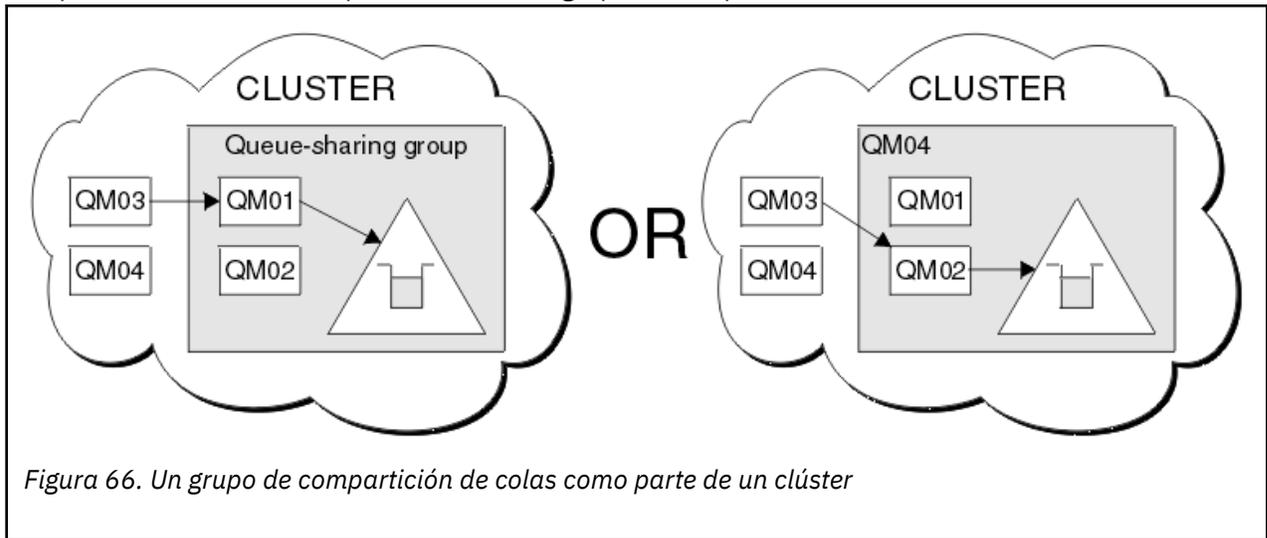
Utilice este tema si desea conocer cómo puede utilizar los grupos de compartición de colas con clústeres.

Puede poner las colas compartidas a disposición de un clúster en una única definición. Para ello, especifique el nombre del clúster al definir la cola compartida.

Los usuarios de la red ven la cola compartida como alojada por cada gestor de colas dentro del grupo de compartición de colas (la cola compartida no se anuncia como alojada por el grupo de compartición de

colas). Los clientes pueden iniciar sesiones con todos los miembros del grupo de compartición de colas para transferir mensajes a la misma cola compartida.

La Figura 66 en la página 217 muestra cómo los miembros de un clúster pueden acceder a una cola compartida a través de cualquier miembro del grupo de compartición de colas.



## **z/OS** Cómo influir en la distribución de la carga de trabajo con colas compartidas

Utilice este tema si desea conocer los factores que afectan la distribución de la carga de trabajo con colas compartidas en un grupo de compartición de colas.

IBM MQ no proporciona el equilibrio de carga de trabajo para colas compartidas. Sin embargo, la distribución de la carga de trabajo en un grupo de compartición de colas (QSG) puede estar influenciado en una *forma basada en la extracción*. La elección sobre qué gestor de colas debe servir a una cola (recibe un mensaje escrito en una cola común) se ve afectada por la capacidad de proceso disponible de cada gestor de colas del grupo de compartición de colas y los objetivos de la gestión de carga de trabajo definidas mediante Sysplex.

No obstante, resulta importante tener en cuenta que el gestor de colas que realiza la MQPUT de un mensaje también tiene una gran influencia al decidir qué gestor de colas obtiene el mensaje.

### **Es más probable que un gestor de colas local realice MQGET**

Para una aplicación que realice una MQPUT, se indica al gestor de colas que sea el gestor de colas con el que se conecta la aplicación.

Las siguientes consideraciones influyen exactamente sobre gestor de colas debe servir una MQPUT de un mensaje realizando una MQGET en nombre de una aplicación de obtención.

Cuando un mensaje se transfiere a una cola compartida vacía, normalmente se publica el gestor de colas local antes de notificar a cualquier otro gestor de colas del grupo de compartición de colas. Si el gestor de colas local está preparado para procesar el mensaje, recibe una notificación de transición de lista procedente del recurso de acoplamiento (CF) antes que cualquier otro gestor de colas del QSG. (Una notificación de transición de lista es una notificación cuyo estado ha sido modificado por la cola compartida de "vacía" a "no vacía".)

Los escenarios posibles, en este caso, son los siguientes:

1. MQPUT del mensaje no permanente fuera del punto de sincronismo y *colocación rápida para un método de obtención en espera*.

Si hay una aplicación con *MQGET con espera* en el gestor de colas local de la cola, el MQPUT del mensaje se pasa directamente al almacenamiento intermedio de la aplicación de recepción y no

se escribe en la cola. Esto ocurre en colas compartidas y no compartidas. Esta característica suele denominarse mecanismo de *colocación rápida para un método de obtención en espera*. En el caso de colas compartidas, no se notifica a ningún otro gestor de colas del QSG que la transición de cola vacía a no vacía no existe. Esto significa que, por ejemplo, siempre que este gestor de colas pueda atender todas las obtenciones de mensajes de esta aplicación y presuponiendo que no hay más aplicaciones que transfieran mensajes a la cola, ningún otro gestor de colas del grupo de compartición de colas ayuda a vaciar esta cola. No obstante, si en el gestor de colas local no hay ningún MQGET en espera y se transfiere un mensaje en la cola compartida, entonces el recurso de acoplamiento notifica a otros gestores de colas del grupo de compartición de cola, dependiendo de las reglas de notificación de transiciones de lista.

2. MQPUT de un mensaje persistente o en punto de sincronismo.

En este caso, si hay una aplicación con un *MQGET con espera* en el gestor de colas local, entonces el mensaje se transfiere a la cola compartida y el CF notifica a los demás gestores de colas del grupo de compartición de cola, en función de las reglas de notificación de transiciones de lista. No obstante, el gestor de colas local no espera una notificación de transición del CF, sino que primero reconoce cualquier *MQGET con espera* local y normalmente obtiene este mensaje en nombre de la aplicación, antes de que cualquier otro gestor de colas del grupo de compartición de colas pueda responder a una notificación de un CF. Ello depende de lo ocupado que esté el gestor de colas local. De lo contrario, cualquier gestor de colas al que el recurso de acoplamiento notifique la llegada del mensaje en la cola vacía intentará servir primero a la obtención. El primer gestor de colas que responda procesará el mensaje nuevo.

3. Por último, si no se drenan los mensajes de la cola, en la que el recurso de acoplamiento ha enviado una notificación de un cambio de estado de vacío o a no vacío para la cola, todos los gestores de colas conectados tendrán la oportunidad de ayudar en el proceso de la cola. En este caso, la carga de trabajo se dice que se *basa en la extracción*.

Este diseño permite mejorar el rendimiento mediante una distribución de la carga de trabajo basada exclusivamente en la extracción. El objetivo es aprovechar la alta disponibilidad ofrecida por las colas conservadas en el CF mientras que, a la vez, es posible que el gestor de colas, siempre que sea posible, realice la llamada MQGET sin tener que hacer referencia al CF y, por lo tanto, procesar la carga de trabajo de mensajes del modo más eficaz posible.

Se puede adoptar enfoques alternativos en los que enfatizar el equilibrio de la carga de trabajo sea más importante que las mejoras de rendimiento previamente descritas. Por ejemplo, puede asegurarse de que ninguna de las aplicaciones de obtención esté conectada al mismo gestor de colas al que esté conectada la aplicación de transmisión. La utilización de este diseño hace que todos los mensajes se coloquen en la cola y que se notifique a todos los gestores de colas en el QSG cuando la cola pase de "vacía" a "no vacía", de acuerdo con el algoritmo del recurso de acoplamiento para manejar tales transiciones. Además, el mecanismo *colocación rápida a método de obtención en espera* no se puede aplicar.

**z/OS** **Dónde encontrar más información sobre colas compartidas y grupos de compartición de colas**

Utilice la tabla de este tema para obtener más información sobre cómo IBM MQ for z/OS utiliza colas compartidas y grupos de compartición de colas.

<i>Tabla 19. Dónde encontrar más información sobre colas compartidas y grupos de compartición de colas</i>	
<b>Tema</b>	<b>Dónde buscar</b>
Recuperación del grupo de compartición de colas	<a href="#">"Recuperación y reinicio en z/OS"</a> en la página 261
Seguridad del grupo de compartición de colas	<a href="#">"Conceptos de seguridad en IBM MQ for z/OS"</a> en la página 279
Definiciones de objeto privado y global. Dirigiendo mandatos a distintos gestores de colas	<a href="#">Orígenes desde los que puede emitir mandatos en z/OS</a>

Tabla 19. Dónde encontrar más información sobre colas compartidas y grupos de compartición de colas (continuación)

Tema	Dónde buscar
Planificación del entorno de recurso entorno	<a href="#">Definición de recursos de acoplamiento</a>
Planificación del entorno SMDS	<a href="#">Planificación del entorno del conjunto de datos de mensajes compartidos (SMDS)</a>
Planificación del Entorno de Db2	<a href="#">Planificación del entorno de Db2</a>
Configuración de las colas compartidas. Parámetros del sistema	<a href="#">“Colas compartidas y grupos de compartición de colas” en la página 171</a>
Programas de utilidad Migración de colas	<a href="#">IBM MQ en z/OS</a>
Mensajes de la consola	<a href="#">Mensajes para IBM MQ for z/OS</a>
Mandatos MQSC	<a href="#">Mandatos MQSC</a>
Clústeres de IBM MQ	<a href="#">Configuración de un clúster de gestores de colas</a>
Colas distribuidas de IBM MQ Nombres de canal	<a href="#">Introducción a la gestión de colas distribuidas</a>
Escritura de aplicaciones	<a href="#">Visión general del diseño de aplicaciones</a>
Llamada MQCONN	<a href="#">MQCONN</a>

## z/OS Transferencia a colas entre grupos

En esta sección se describe la gestión de colas dentro del grupo, una función de IBM MQ for z/OS exclusiva de la plataforma z/OS. Esta función sólo está disponible para los gestores de colas definidos en un grupo de compartición de colas.

Para obtener información sobre grupos de compartición de colas, consulte [“Colas compartidas y grupos de compartición de colas” en la página 171](#).

### Conceptos de transferencia a colas dentro del grupo

Puede realizar una transferencia de mensajes rápida entre gestores de colas de un grupo de compartición de colas sin definir canales. Para ello se utiliza una cola del sistema denominada SYSTEM.QSG.TRANSMIT.QUEUE, que es una cola de transmisión compartida. Cada gestor de colas del grupo de compartición de colas inicia una tarea denominada "agente de transferencia a colas dentro del grupo", que espera hasta que lleguen mensajes en esta cola que estén destinados a su gestor de colas. Cuando se detectan tales mensajes, se eliminan de la cola y se colocan en la cola de destino correcta.

Se utilizan las reglas de resolución de nombres estándar pero, si se ha habilitado la transferencia a colas dentro del grupo (IGQ) y el gestor de colas de destino se encuentra dentro del grupo de compartición de colas, se utiliza SYSTEM.QSG.TRANSMIT.QUEUE para transferir el mensaje al gestor de colas de destino correcto en lugar de utilizar una cola de transmisión y un canal.

Habilite la transferencia a colas dentro del grupo mediante un atributo del gestor de colas. La transferencia a colas dentro del grupo mueve mensajes no permanentes fuera del punto de sincronización y los mensajes permanentes dentro del punto de sincronización. Si encuentra un problema en la entrega de mensajes a la cola de destino, la transferencia a colas dentro del grupo intenta colocarlos en la cola de mensajes no entregados. Si la cola de mensajes no entregados está llena o no se ha definido, los mensajes no permanentes se descartan pero se realiza una copia de seguridad de los mensajes permanente y se devuelven a SYSTEM.QSG.TRANSMIT.QUEUE y el agente de transferencia a colas dentro del grupo (IGQ) intenta entregar los mensajes hasta tener éxito.

Un canal compartido de entrada que recibe un mensaje destinado a una cola en un gestor de colas distinto en el grupo de compartición de colas puede utilizar la transferencia a colas dentro del grupo para *saltar* el mensaje e ir al destino correcto.

Puede haber ocasiones en las que desee que el gestor de colas local transfiera un mensaje directamente a la cola de destino, si la cola de destino es una cola compartida, en lugar del primer mensaje que se transfiere al gestor de colas de destino. Puede utilizar el atributo del gestor de colas SQQMNAME para controlarlo. Si establece el valor de SQQMNAME en USE, el mandato MQOPEN se ejecuta en el gestor de colas especificado por **ObjectQMgrName**.

Sin embargo, si la cola de destino es una cola compartida y establece el valor de SQQMNAME en IGNORE, y **ObjectQMgrName** es el de otro gestor de colas del grupo de compartición de colas, la cola compartida se abre en el gestor de colas local. Si el gestor de colas local no puede abrir la cola de destino, o transferir un mensaje a la cola, el mensaje se transfiere al **ObjectQMgrName** especificado a través de IGQ o de un canal IBM MQ .

La transferencia a colas dentro del grupo se puede utilizar para entregar, de forma más eficaz, mensajes pequeños a colas que residen en gestores de colas remotos dentro de un grupo de compartición de colas. La transferencia a colas dentro del grupo también da soporte a mensajes grandes, siendo el más grande de 100 MB *menos* la longitud de la cabecera de cola de transmisión.

**Nota:** Si utiliza esta característica, los usuarios deben tener el mismo acceso a las colas en cada gestor de colas del grupo de compartición de colas.

En el diagrama siguiente se muestra un ejemplo típico de transferencia a colas dentro del grupo.

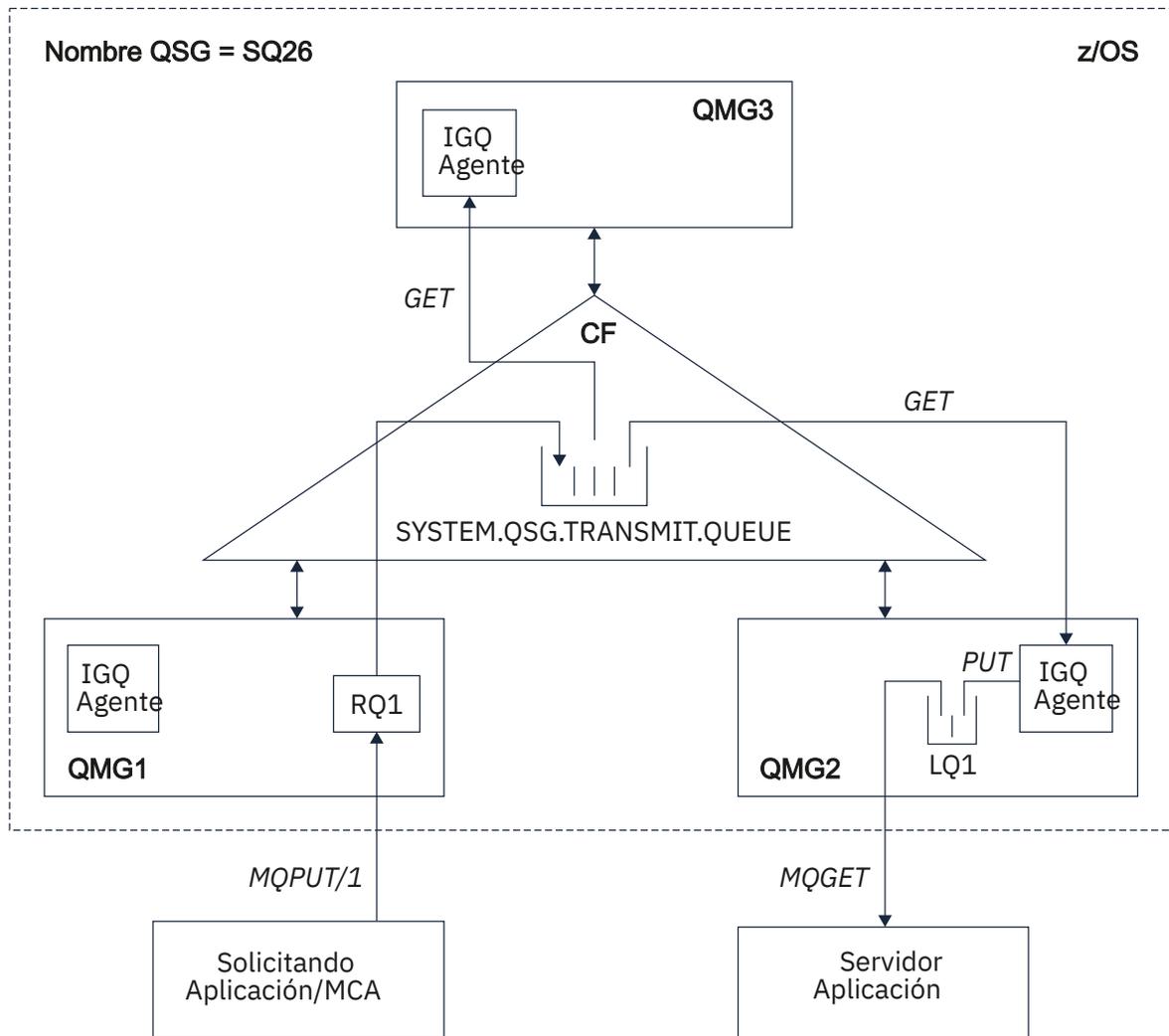


Figura 67. Ejemplo de transferencia a colas dentro del grupo

En el diagrama se muestran:

- Los agentes de IGQ que se ejecutan en tres gestores de colas (QMG1, QMG2 y QMG3) que se han definido para un grupo de compartición de colas denominado SQ26.
- La cola de transmisión compartida SYSTEM.QSG.TRANSMIT.QUEUE que se define en el recurso de acoplamiento (CF).
- Una definición de cola remota que se define en el gestor de colas QMG1.
- Una cola local que se define en el gestor de colas QMG2.
- Una aplicación solicitante (esta aplicación podría ser un Agente de canal de mensajes, MCA) que está conectada al gestor de colas QMG1.
- Una aplicación de servidor que está conectada al gestor de colas QMG2.
- Un mensaje de solicitud que se coloca en la cola SYSTEM.QSG.TRANSMIT.QUEUE.

### Transferencia a colas dentro del grupo y el agente de transferencia a colas dentro del grupo

Un agente de IGQ se inicia durante la inicialización del gestor de colas. Cuando las aplicaciones se abren y transfieren mensajes a colas remotas, el gestor de colas local determina si la transferencia a colas dentro del grupo se utiliza para la transferencia de mensajes. Si se va a utilizar la transferencia a colas dentro del

grupo, el gestor de colas local transfiere el mensaje a la cola SYSTEM.QSG.TRANSMIT.QUEUE. El agente de IGQ del gestor de colas remoto de destino recupera el mensaje y lo transfiere a la cola de destino.

## **Terminología de transferencia a colas dentro del grupo**

Explicaciones de la terminología: transferencia a colas dentro del grupo, cola de transmisión compartida para el uso de la transferencia a colas dentro del grupo y agente de transferencia a colas dentro del grupo.

### **Transferencia a colas entre grupos**

La transferencia a colas dentro del grupo puede ocasionar una transferencia potencialmente más rápida y menos costosa de mensajes entre gestores de colas de un grupo de compartición de colas, sin tener que definir canales.

### **Cola de transmisión compartida para utilizarla por la transferencia a colas dentro del grupo**

Cada grupo de compartición de colas tiene una cola de transmisión compartida denominada SYSTEM.QSG.TRANSMIT.QUEUE para el uso de la transferencia a colas dentro del grupo. Si la transferencia a colas dentro del grupo está habilitada, SYSTEM.QSG.TRANSMIT.QUEUE aparece en la vía de resolución de nombres cuando se abren colas remotas. Cuando las aplicaciones (incluyendo los agentes de canal de mensajes, MCA) ponen mensajes en una cola remota, el gestor de colas local determina si los mensajes reúnen los requisitos para transferirse de forma rápida y los coloca en SYSTEM.QSG.TRANSMIT.QUEUE.

### **Agente de transferencia a colas dentro del grupo**

El agente de IGQ es la tarea, iniciada durante la inicialización del gestor de colas, que espera los mensajes adecuados para llegar a la cola SYSTEM.QSG.TRANSMIT.QUEUE. El agente de IGQ recupera los mensajes adecuados de esta cola y los entrega a las colas de destino.

El agente de IGQ de cada gestor de colas siempre se inicia porque la transferencia a colas dentro del grupo la utiliza el propio gestor de colas para su propio proceso interno.

## **Ventajas de la transferencia a colas dentro del grupo**

Las ventajas de la transferencia a colas dentro del grupo son: menor número de definiciones del sistema, simplificación de la administración del sistema, mejora del rendimiento, soporte a la migración y la entrega de mensajes cuando se producen saltos múltiples entre gestores de colas de un grupo de compartición de colas.

Las ventajas de la transferencia a colas dentro del grupo son:

### **Menor número de definiciones del sistema**

La transferencia a colas dentro del grupo elimina la necesidad de definir canales entre gestores de colas de un grupo de compartición de colas.

### **Simplificación de la administración del sistema**

Como no hay canales definidos entre los gestores de colas de un grupo de compartición de colas, no hay necesidad de administrar canales.

### **Mejor rendimiento**

Puesto que sólo es necesario un agente de IGQ para la entrega de un mensaje a una cola de destino (en vez de dos agentes emisor y receptor intermedios), la entrega de mensajes utilizando la transferencia a colas dentro del grupo puede ser menos costosa que la entrega de mensajes utilizando canales. En la transferencia a colas dentro del grupo sólo hay un componente receptor ya que se ha eliminado la necesidad del componente emisor. Este ahorro se debe a que el mensaje está disponible para el agente de IGQ del gestor de colas de destino para su entrega a la cola de destino una vez que la operación de transferencia en el gestor de colas local se ha completado y, en el caso de los mensajes transferidos en el ámbito del punto de sincronización, que se ha confirmado.

## Soporte a la migración

Las aplicaciones externas a un grupo de compartición de colas pueden entregar mensajes a una cola que reside en cualquier gestor de colas del grupo de compartición de colas, mientras se estén conectadas únicamente a un determinado gestor de colas del grupo de compartición de colas. Esto se debe a que la llegada de mensajes en un canal receptor, destinado a una cola de un gestor de colas remoto, pueden enviarse de forma transparente a la cola de destino utilizando la transferencia a colas dentro del grupo. Este recurso permite que las aplicaciones se desplieguen entre el grupo de compartición de colas sin necesidad de cambiar los sistemas que son externos al grupo de compartición de colas.

Una configuración típica se muestra en el diagrama siguiente, en el que:

- Una aplicación solicitante conectada al gestor de colas QMG1 necesita enviar un mensaje a una cola local del gestor de colas QMG3.
- El gestor de colas QMG1 está conectado sólo al gestor de colas QMG2.
- Los gestores de colas de QMG2 y QMG3, que anteriormente estaban conectados mediante canales, ahora son miembros del grupo de compartición de colas SQ26.

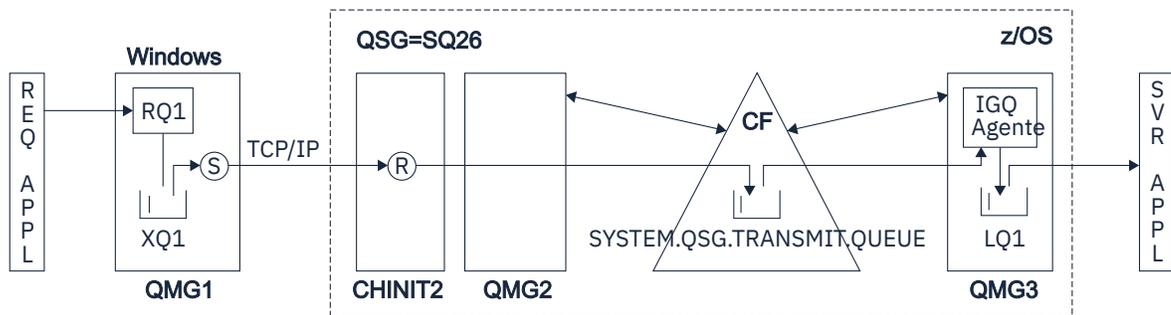


Figura 68. Ejemplo de soporte a la migración

El flujo de operaciones es el siguiente:

1. La aplicación solicitante coloca un mensaje, destinado a la cola local LQ1 del gestor de colas remoto QMG3, en la definición de cola remota RQ1.
2. El gestor de colas QMG1, que se ejecuta en una estación de trabajo Windows NT coloca el mensaje en la cola de transmisión XQ1.
3. El MCA emisor (S) en QMG1 transmite el mensaje, utilizando TCP/IP, al MCA receptor (R) en el iniciador de canal CHINIT2.
4. El MCA receptor (R) en el iniciador de canal CHINIT2 coloca el mensaje en la cola de transmisión compartida SYSTEM.QSG.TRANSMIT.QUEUE.
5. El agente de IGQ del gestor de colas QMG3 recupera el mensaje de la cola SYSTEM.QSG.TRANSMIT.QUEUE y lo coloca en la cola local de destino LQ1.
6. La aplicación de servidor recupera el mensaje de la cola local de destino y lo procesa.

## Entrega de mensajes cuando se producen saltos múltiples entre gestores de colas de un grupo de compartición de colas

El diagrama anterior en [Soporte a la migración](#), también muestra la entrega de mensajes cuando se producen saltos múltiples entre gestores de cola de un grupo de compartición de colas. Los mensajes que llegan a un gestor de colas del grupo de compartición de colas, pero que están destinados a una cola de otro gestor de colas del grupo de compartición de colas, pueden transmitirse fácilmente a la cola de destino del gestor de colas de destino, utilizando la transferencia a colas dentro del grupo.

## Limitaciones de la transferencia a colas dentro del grupo

Las limitaciones de la transferencia a colas dentro del grupo son: mensajes elegibles para transferir utilizando la transferencia a colas dentro del grupo, el número de agentes de transferencia a colas dentro del grupo por gestor de colas, e iniciar y detener el agente de transferencia a colas dentro del grupo.

En este tema se describen las limitaciones de la transferencia a colas dentro del grupo.

### Mensajes elegibles para transferir utilizando la transferencia a colas dentro del grupo

Como la transferencia a colas dentro del grupo utiliza una cola de transmisión compartida que se define en el recurso de acoplamiento (CF), la transferencia a colas dentro del grupo está limitada a la entrega de mensajes de la longitud máxima del mensaje admitida por las colas compartidas menos la longitud de una cabecera de la cola de transmisión (MQXQH).

### Número de agentes de transferencia a colas dentro del grupo por gestor de colas

En un grupo de compartición de colas sólo se inicia un agente de IGQ por cada gestor de colas.

### Iniciar y detener el agente de transferencia a colas dentro del grupo

El agente de IGQ se inicia durante la inicialización del gestor de colas y finaliza durante el cierre del gestor de colas. Está diseñada para ser una tarea larga, capaz de recuperarse automáticamente (en el caso de finalización anómala). Si hay un error en la definición de la cola SYSTEM.QSG.TRANSMIT.QUEUE (por ejemplo, si está inhibida para recuperaciones) el agente de IGQ continuará intentándolo. Si el agente de IGQ encuentra un error que tiene como consecuencia la finalización normal del agente mientras el gestor de colas sigue activo, puede reiniciarse emitiendo un mandato ALTER QMGR IGQ(ENABLED). Este mandato evita la necesidad de reciclar el gestor de colas.

### Configuración del atributo IGQ del gestor de colas en ENABLED o DISABLED

Si se establece el atributo IGQ del gestor de colas en ENABLED o DISABLED, es posible que los manejadores de objetos existentes se invaliden con el código de razón MQRC\_OBJECT\_CHANGED. Para obtener más información, consulte la sección [Iniciación a la transferencia a colas entre grupos](#).

## Cómo empezar a utilizar la transferencia a colas dentro del grupo

Puede habilitar, inhabilitar y utilizar la transferencia a colas dentro del grupo como se describe en este tema.

### Habilitación de la transferencia a colas dentro del grupo

Para habilitar la transferencia a colas dentro del grupo en los gestores de colas, necesita hacer lo siguiente:

- Defina una cola de transmisión compartida denominada SYSTEM.QSG.TRANSMIT.QUEUE. La definición de esta cola puede encontrarse en thlqual.SCSQPROC(CSQ4INSS), el ejemplo CSQINP2 de objetos del sistema (SYSTEM) para grupos de compartimiento de colas. Esta cola debe definirse con los atributos correctos, como se indica en thlqual.SCSQPROC(CSQ4INSS), para que la transferencia a colas dentro del grupo funcione correctamente.
- Como el agente de IGQ siempre se inicia durante la inicialización del gestor de colas, la transferencia a colas dentro del grupo siempre está disponible para el proceso de mensajes de entrada. El agente de IGQ procesa todos los mensajes que se ponen en la cola SYSTEM.QSG.TRANSMIT.QUEUE. Sin embargo, para habilitar la transferencia a colas dentro del grupo para el proceso de salida, el atributo del gestor de colas IGQ debe establecerse en ENABLED (habilitado).

**Importante:** Si se establece el atributo IGQ del gestor de colas en ENABLED, es posible que los manejadores de objetos existentes se invaliden con el código de razón MQRC\_OBJECT\_CHANGED. Consulte [“Propiedades específicas de la transferencia a colas dentro del grupo”](#) en la página 232 para obtener más información. Tal como se describe en la sección 'Respuesta del programador' para este código de razón, es necesario codificar las aplicaciones para manejar esta situación (consulte [2041 \(07F9\) \(RC2041\): MQRC\\_OBJECT\\_CHANGED](#) para obtener más detalles).

Además, dado que IGQ se ha diseñado como una tarea de autorecuperación y larga ejecución, que se inician durante la inicialización y finaliza durante la conclusión, consulte la sección [“Limitaciones de la transferencia a colas dentro del grupo”](#) en la página 224 para obtener más información.

## Inhabilitación de la transferencia a colas dentro del grupo

Para inhabilitar transferencia a colas dentro del grupo para la transferencia de mensajes de salida, establezca el atributo del gestor de colas IGQ en DISABLED (inhabilitado). Si la transferencia a colas dentro del grupo está inhabilitada para un determinado gestor de colas, el agente de IGQ de ese gestor de colas todavía puede procesar los mensajes de entrada que se han colocado en la cola SYSTEM.QSG.TRANSMIT.QUEUE de un gestor de colas que tenga habilitada la transferencia a colas dentro del grupo para la transferencia de salida.

**Importante:** Si se establece el atributo IGQ del gestor de colas en ENABLED, es posible que los manejadores de objetos existentes se invaliden con el código de razón MQRC\_OBJECT\_CHANGED. Consulte “Propiedades específicas de la transferencia a colas dentro del grupo” en la página 232 para obtener más información. Tal como se describe en la sección 'Respuesta del programador' para este código de razón, es necesario codificar las aplicaciones para manejar esta situación (consulte [2041 \(07F9\) \(RC2041\): MQRC\\_OBJECT\\_CHANGED](#) para obtener más detalles).

Además, dado que IGQ se ha diseñado como una tarea de autorecuperación y larga ejecución, que se inician durante la inicialización y finaliza durante la conclusión, consulte la sección “[Limitaciones de la transferencia a colas dentro del grupo](#)” en la página 224 para obtener más información.

## Utilización de la transferencia a colas dentro del grupo

Una vez habilitada la transferencia a colas dentro del grupo, puede utilizarse y un gestor de colas la utilizará siempre que sea posible. Es decir, cuando una aplicación coloca un mensaje en una definición de cola remota, en una cola remota totalmente calificada o en una cola de clúster, el gestor de colas determina si el mensaje es apto para ser entregado utilizando la transferencia a colas dentro del grupo y si es así, coloca el mensaje en la cola SYSTEM.QSG.TRANSMIT.QUEUE. No hay necesidad de cambiar las aplicaciones de usuario ni las colas de aplicación, porque el gestor de colas utiliza la cola SYSTEM.QSG.TRANSMIT.QUEUE para los mensajes aptos, en vez de cualquier otra cola de transmisión.

## configuraciones de transferencia a colas dentro del grupo

Además de la típica configuración de transferencia a colas dentro del grupo, pueden existir otras configuraciones.

“[Conceptos de transferencia a colas dentro del grupo](#)” en la página 219 describe la configuración típica.

### Conceptos relacionados

“[Gestión de colas distribuidas con transferencia a colas dentro del grupo \(varias rutas de entrega\)](#)” en la página 225

Para las aplicaciones que procesan mensajes cortos puede que sea factible configurar la transferencia a colas dentro del grupo sólo para suministrar mensajes entre gestores de colas de un grupo de compartición de colas.

“[Agrupación en clúster con transferencia a colas dentro del grupo \(varias rutas de entrega\)](#)” en la página 227

Es posible configurar gestores de colas para que estén en un clúster y en un grupo de compartición de colas.

“[Agrupación en clúster, transferencia a colas dentro del grupo y colas distribuidas](#)” en la página 230

Es posible configurar un gestor de colas que sea miembro de un clúster y de un grupo de compartición de colas y que esté conectado a un gestor de colas distribuidas mediante un par de canales emisor/receptor.

## *Gestión de colas distribuidas con transferencia a colas dentro del grupo (varias rutas de entrega)*

Para las aplicaciones que procesan mensajes cortos puede que sea factible configurar la transferencia a colas dentro del grupo sólo para suministrar mensajes entre gestores de colas de un grupo de compartición de colas.

La elección de transferencia a colas dentro del grupo a través de las comunicaciones del canal puede controlarse mediante el nivel del tipo CFSTRUCT (3 en lugar de 4 ó 5). La longitud máxima del mensaje tal como se ha establecido en SYSTEM.QSQ.TRANSMIT.QUEUE.

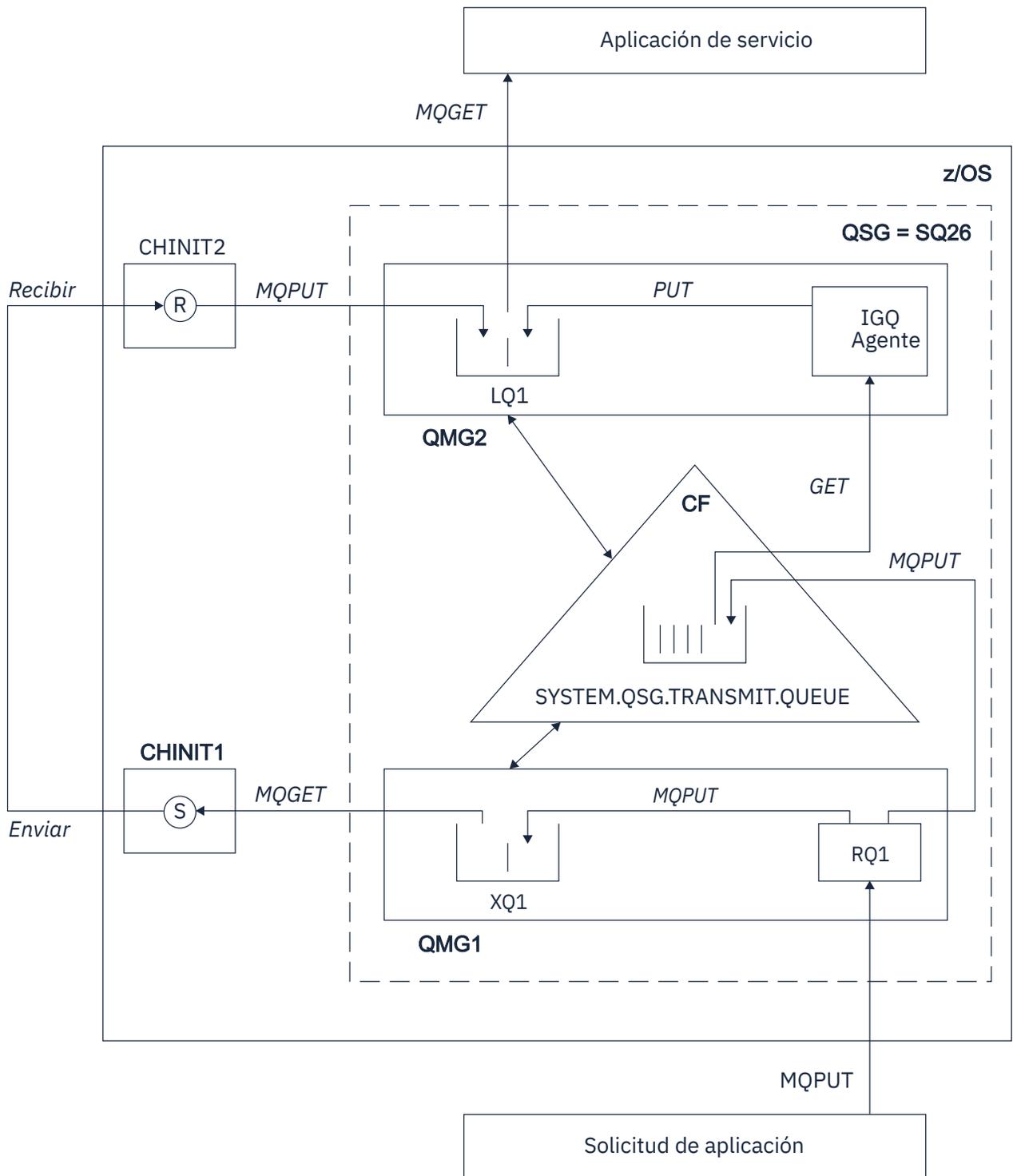


Figura 69. Configuración de ejemplo

### Proceso de apertura y transferencia

1. Es importante señalar que cuando la aplicación solicitante abra la cola remota RQ1, se lleva a cabo la resolución de nombres para la cola de transmisión no compartida XQ1 y la cola de transmisión compartida SYSTEM. QSG. TRANSMIT. QUEUE.
2. Cuando la aplicación solicitante coloca un mensaje en la cola remota, en función de si la transferencia a colas dentro del grupo está habilitada para transferencia de salida en el gestor de colas y de las características del mensaje, el mensaje se transfiere a la cola de transmisión XQ1 o a la

cola de transmisión SYSTEM.QSG.TRANSMIT.QUEUE. El gestor de colas coloca todos los mensajes grandes en la cola de transmisión XQ1 y todos los mensajes pequeños en la cola de transmisión SYSTEM.QSG.TRANSMIT.QUEUE.

3. Si la cola de transmisión XQ1 está llena o no está disponible, las peticiones de transferencia de mensajes grandes fallan de forma síncrona con un código de retorno y un código de razón adecuados. Sin embargo, las peticiones de transferencia de mensajes pequeños continúan realizándose con éxito y colocándose en la cola de transmisión SYSTEM.QSG.TRANSMIT.QUEUE.
4. Si la cola de transmisión SYSTEM.QSG.TRANSMIT.QUEUE está llena o no se pueden transferir mensajes a ella, las peticiones de transferencia de mensajes pequeños fallan de forma síncrona con un código de retorno y un código de razón adecuados. Sin embargo, las peticiones de transferencia de mensajes grandes continúan realizándose con éxito y colocándose en la cola de transmisión XQ1. En este caso, no se hace ningún intento de poner los mensajes pequeños en una cola de transmisión.

### **Flujo de mensajes grandes**

1. La aplicación solicitante coloca mensajes grandes en la cola remota RQ1.
2. El gestor de colas QMG1 pone los mensajes en la cola de transmisión XQ1.
3. EL MCA emisor (S) en el gestor de colas QMG1 recupera los mensajes de la cola de transmisión XQ1 y los envía al gestor de colas QMG2.
4. El MCA receptor (R) en el gestor de colas QMG2 recibe los mensajes y los coloca en la cola de destino LQ1.
5. La aplicación de servicio recupera y procesa los mensajes de la cola LQ1.

### **Flujo de mensajes pequeños**

1. La aplicación solicitante coloca mensajes pequeños en la cola remota RQ1.
2. El gestor de colas QMG1 pone los mensajes en la cola de transmisión SYSTEM.QSG.TRANSMIT.QUEUE.
3. IGQ en el gestor de colas QMG2 recupera los mensajes y los coloca en la cola de destino LQ1.
4. La aplicación de servicio recupera los mensajes de la cola LQ1.

### **Puntos que se deben tener en cuenta**

1. La aplicación solicitante no necesita conocer el mecanismo subyacente utilizado para la entrega de mensajes.
2. Puede conseguirse un mecanismo de entrega de mensajes potencialmente más rápido para mensajes pequeños.
3. Existen varias rutas para la entrega de mensajes (es decir, la ruta normal del canal y la ruta de la transferencia a colas dentro del grupo).
4. La ruta de la transferencia a colas dentro del grupo, potencialmente más rápida, se selecciona antes que la ruta normal del canal. En función de las características del mensaje, la entrega de mensajes podría dividirse entre las dos rutas. Por lo tanto, es posible que los mensajes se entreguen desordenados (aunque esta entrega también es posible si los mensajes se entregan utilizando sólo la ruta normal del canal).
5. Cuando se ha seleccionado una ruta y se han colocado mensajes en las colas de transmisión, sólo se utiliza la ruta seleccionada para la entrega de mensajes. Los mensajes no procesados en la cola SYSTEM.QSG.TRANSMIT.QUEUE no se desvían a la cola de transmisión XQ1.

### **z/OS Agrupación en clúster con transferencia a colas dentro del grupo (varias rutas de entrega)**

Es posible configurar gestores de colas para que estén en un clúster y en un grupo de compartición de colas.

Cuando se envían mensajes a una cola de clúster y los gestores de colas de destino locales y remotos están en el mismo grupo de compartición de colas, la transferencia a colas dentro del grupo se utiliza para la entrega de mensajes pequeños (utilizando la cola SYSTEM.QSG.TRANSMIT.QUEUE) y la entrega de mensajes grandes si la transferencia a colas dentro del grupo soporta el tamaño del mensaje. Además, la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE se utiliza para la entrega de mensajes a cualquier gestor de colas que esté en el clúster, pero fuera del grupo de compartición de colas. En el diagrama siguiente se ilustra esta configuración (los iniciadores de canal no se muestran).

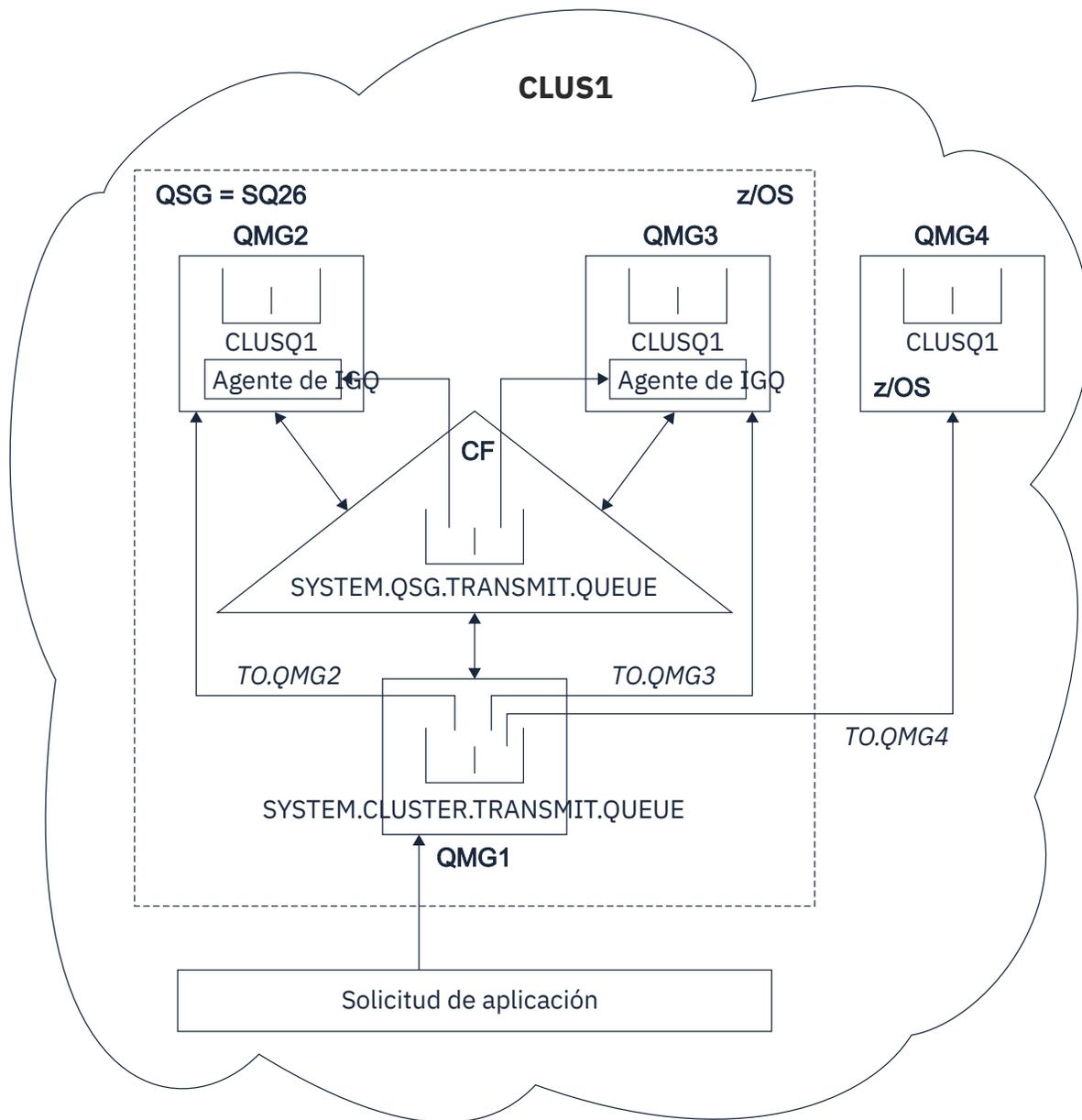


Figura 70. Ejemplo de agrupación en clúster con transferencia a colas dentro del grupo

En el diagrama se muestran:

- Cuatro gestores de colas de z/OS: QMG1, QMG2, QMG3, y QMG4 configurados en un clúster CLUS1.
- Los gestores de colas QMG1, QMG2 y QMG3 configurados en un grupo de compartición de colas SQ26.
- Los agentes de IGQ que se ejecutan en los gestores de colas QMG2 y QMG3.
- La cola SYSTEM.CLUSTER.TRANSMIT.QUEUE local definida en QMG1.

**Nota:** Para mayor claridad, no se muestra SYSTEM.CLUSTER.TRANSMIT.QUEUE en los otros gestores de colas.

- La cola SYSTEM.QSG.TRANSMIT.QUEUE compartida definida en el CF, que está en una estructura IBM MQ configurada con el atributo CFLEVEL (3) RECOVER (YES).
- Los canales de clúster TO.QMG2 (que conecta QMG1 con QMG2), TO.QMG3 (que conecta QMG1 con QMG3) y TO.QMG4 (que conecta QMG1 con QMG4).
- La cola de clúster CLUSQ1 que está alojada en los gestores de colas QMG2, QMG3 y QMG4.

Suponga que la aplicación solicitante abre la cola de clúster con la opción MQOO\_BIND\_NOT\_FIXED, de modo que el gestor de colas de destino para la cola de clúster está seleccionada en el momento de la transferencia.

Si el gestor de colas de destino seleccionado es QMG2:

- Todos los mensajes grandes transferidos por la aplicación solicitante:
  - Se colocan en la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE en QMG1, porque SYSTEM.QSG.TRANSMIT.QUEUE está en una estructura CFLEVEL(3); por lo tanto, sólo da soporte a mensajes con un tamaño de hasta 63 KB.
  - Se transfieren a la cola de clúster CLUSQ1 en QMG2 utilizando el canal de clúster TO.QMG2
- Todos los mensajes pequeños transferidos por la aplicación solicitante
  - Se colocan en la cola de transmisión compartida SYSTEM.QSG.TRANSMIT.QUEUE. Esta cola está en una estructura configurada con el atributo RECOVER(YES), por lo que se utiliza para mensajes pequeños, tanto persistentes como no persistentes.
  - Los recupera el agente de IGQ en QMG2
  - Se colocan en la cola de clúster CLUSQ1 en QMG2

Si el gestor de colas de destino seleccionado es QMG4:

- Debido a que QMG4 no es miembro del grupo de compartición de colas SQ26, todos los mensajes transferidos por la aplicación solicitante
  - Se colocan en la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE en QMG1
  - Se transfieren a la cola de clúster CLUSQ1 en QMG4 utilizando el canal de clúster TO.QMG4

## **Puntos que se deben tener en cuenta**

- La aplicación solicitante no necesita conocer el mecanismo subyacente utilizado para la entrega de mensajes.
- Se obtiene un mecanismo potencialmente más rápido para la transferencia de mensajes no permanentes pequeños entre gestores de colas en un grupo de compartición de colas (aunque los mismos gestores de colas estén en un clúster).
- Hay varias rutas disponibles para la entrega de mensajes (es decir, la ruta del clúster y la ruta de la transferencia a colas dentro del grupo).
- La ruta de la transferencia a colas dentro del grupo, potencialmente más rápida, se selecciona antes que la ruta del clúster. En función de las características del mensaje, la entrega de mensajes podría dividirse entre las dos rutas. Por lo tanto, es posible que los mensajes se entreguen desordenados. Es importante señalar que esta entrega es posible sea cual sea la opción MQOO\_BIND\_\* especificada por la aplicación. La transferencia a colas dentro del grupo distribuye los mensajes del mismo modo que la agrupación en clúster, dependiendo de si se ha especificado MQOO\_BIND\_NOT\_FIXED, MQOO\_BIND\_ON\_OPEN, MQOO\_BIND\_ON\_GROUP o MQOO\_BIND\_AS\_Q\_DEF en la apertura.
- Cuando se ha seleccionado una ruta y se han colocado mensajes en las colas de transmisión, sólo se utiliza la ruta seleccionada para la entrega de mensajes. Los mensajes no procesados en la cola SYSTEM.QSG.TRANSMIT.QUEUE no se desvían a la cola SYSTEM.CLUSTER.TRANSMIT.QUEUE.

## **Agrupación en clúster, transferencia a colas dentro del grupo y colas distribuidas**

Es posible configurar un gestor de colas que sea miembro de un clúster y de un grupo de compartición de colas y que esté conectado a un gestor de colas distribuidas mediante un par de canales emisor/receptor.

Esta configuración es una combinación de gestión de colas distribuidas con transferencia a colas dentro del grupo y agrupación en clúster con transferencia a colas dentro del grupo.

La transferencia a colas dentro del grupo se describe en [“Gestión de colas distribuidas con transferencia a colas dentro del grupo \(varias rutas de entrega\)”](#) en la página 225.

La agrupación en clúster con transferencia a colas dentro del grupo se describe en [“Agrupación en clúster con transferencia a colas dentro del grupo \(varias rutas de entrega\)”](#) en la página 227.

## **Mensajes de transferencia a colas dentro del grupo**

En esta sección se describen los mensajes transferidos a la cola SYSTEM.QSG.TRANSMIT.QUEUE.

### **Estructura de los mensajes**

Al igual que los demás mensajes que se transfieren a colas de transmisión, a los mensajes que se transfieren a la cola SYSTEM.QSG.TRANSMIT.QUEUE se les asigna como prefijo la cabecera de la cola de transmisión (MQXQH).

### **Persistencia de los mensajes**

En IBM WebSphere MQ 5.3 y anteriores, las colas compartidas soportan mensajes persistentes y no persistentes.

Si el gestor de colas termina mientras el agente de IGQ está procesando mensajes no permanentes, o si el agente de IGQ finaliza de forma anómala mientras está procesando mensajes, podrían perderse los mensajes no permanentes que se están procesando. Las aplicaciones deben hacer preparativos para la recuperación de los mensajes no permanentes si es necesaria su recuperación.

Si una petición de transferencia de un mensaje no persistente, emitida por el agente de IGQ, falla inesperadamente, el mensaje que se está procesando se pierde.

### **Entrega de los mensajes**

El agente de IGQ recupera y entrega todos los mensajes no permanentes fuera del ámbito del punto de sincronización y todos los mensajes permanentes dentro del ámbito del punto de sincronización. En este caso, el agente de IGQ actúa como el coordinador del punto de sincronización. Por lo tanto, el agente de IGQ procesa los mensajes no permanentes de la misma forma que un canal de mensajes procesa los mensajes rápidos no persistentes. Consulte [Mensajes rápidos no persistentes](#).

### **Proceso por lotes de los mensajes**

El agente de IGQ utiliza un tamaño fijo del lote de 50 mensajes. Los mensajes permanentes recuperados dentro de un lote se confirman a intervalos de 50 mensajes. El agente confirma un lote que consta de mensajes permanentes cuando ya no quedan más mensajes que recuperar en la cola SYSTEM.QSG.TRANSMIT.QUEUE.

### **Tamaño del mensaje**

El tamaño máximo de los mensajes que pueden transferirse a la cola SYSTEM.QSG.TRANSMIT.QUEUE es la longitud máxima del mensaje admitida por las colas compartidas menos la longitud de una cabecera de la cola de transmisión (MQXQH).

### **Persistencia y prioridad de mensajes predeterminadas**

Si la cola SYSTEM.QSG.TRANSMIT.QUEUE está en la vía de resolución de nombres de colas establecida en el momento de la apertura, para los mensajes que se transfieren con persistencia predeterminada y prioridad predeterminada (o con persistencia predeterminada o prioridad predeterminada), se aplican las reglas normales en la selección de la cola que tiene prioridad predeterminada y los valores de persistencia utilizados. (Consulte la sección [Mensajes de IBM MQ](#) para obtener más información sobre las reglas de selección de colas).

### **Conceptos relacionados**

[“Mensajes no entregados y no procesados”](#) en la página 231

En este tema se describe lo que sucede con los mensajes no entregados y no procesados en la cola SYSTEM.QSG.TRANSMIT.QUEUE.

“Mensajes de informe - Transferencia a colas dentro del grupo” en la página 231

En este tema se describen los mensajes de informe: confirmación de llegada, confirmación de entrega, informe de caducidad e informe de excepción.

### **Mensajes no entregados y no procesados**

En este tema se describe lo que sucede con los mensajes no entregados y no procesados en la cola SYSTEM.QSG.TRANSMIT.QUEUE.

Si un agente de IGQ no puede entregar un mensaje a la cola de destino, el agente de IGQ:

- Respetar la opción de informe MQRO\_DISCARD\_MSG (si el campo Opciones de informe del MQMD del mensaje no entregado indica que debe hacerlo) y elimina el mensaje no entregado.
- Intenta colocar el mensaje no entregado en la cola de mensajes no entregados del gestor de colas de destino, si el mensaje todavía no se ha descartado. El agente de IGQ añade al mensaje un prefijo con la cabecera de una cola de mensajes no entregados (MQDLH).

Si no se define una cola de mensajes no entregados o si se transfiere un mensaje no entregado a la cola de mensajes no entregados y el mensaje no entregado es:

- persistente, el agente de IGQ restituye el lote actual de mensajes permanentes que está procesando y entra en estado de reintento. Para obtener más información, consulte [“Propiedades específicas de la transferencia a colas dentro del grupo”](#) en la página 232.
- no persistente, el agente de IGQ descarta el mensaje y sigue procesando el siguiente mensaje.

Si un gestor de colas de un grupo de compartición de colas finaliza antes de que su agente de IGQ asociado haya tenido tiempo de procesar todos sus mensajes, los mensajes no procesados permanecerán en la cola SYSTEM.QSG.TRANSMIT.QUEUE hasta que el gestor de colas se vuelva a iniciar. A continuación, el agente de IGQ recupera y entrega los mensajes a las colas de destino.

Si el recurso de acoplamiento falla antes de haber procesado todos los mensajes de la cola SYSTEM.QSG.TRANSMIT.QUEUE, los mensajes no permanentes no procesados se perderán.

IBM recomienda que las aplicaciones no coloquen mensajes directamente en las colas de transmisión. Si una aplicación pone mensajes directamente en la cola SYSTEM.QSG.TRANSMIT.QUEUE, es posible que el agente de IGQ no pueda procesar estos mensajes, que permanecerán en la cola SYSTEM.QSG.TRANSMIT.QUEUE. En este caso, los usuarios tendrán que utilizar sus propios métodos para manejar estos mensajes no procesados.

### **Mensajes de informe - Transferencia a colas dentro del grupo**

En este tema se describen los mensajes de informe: confirmación de llegada, confirmación de entrega, informe de caducidad e informe de excepción.

#### **Mensajes de informe de confirmación de llegada (COA) y confirmación de entrega (COD)**

Los mensajes COA y COD los genera el gestor de colas cuando se utiliza la transferencia a colas dentro del grupo.

#### **Mensajes de informe de caducidad**

Los mensajes de informe de caducidad los genera el gestor de colas.

#### **Mensajes de informe de excepción**

Según la opción de informe MQRO\_EXCEPTION\_\* especificada en el campo *Opciones de informe* del descriptor de mensaje del mensaje no entregado, el agente de IGQ genera el informe de excepción necesario y lo coloca en la cola de respuestas especificada. La transferencia a colas dentro del grupo se pueden utilizar para entregar el informe de excepción en la cola de respuestas de destino.

La persistencia del mensaje de informe es la misma que la persistencia del mensaje no entregado. Si el agente de IGQ no puede resolver el nombre de la cola de respuestas de destino o si no puede poner el mensaje de respuesta en una cola de transmisión (para ser transferido posteriormente a la cola de respuestas de destino) intenta poner el informe de excepción en la cola de mensajes no entregados

del gestor de colas en el que se ha generado el mensaje de informe. Si no es posible y si el mensaje no entregado es:

- persistente, el agente de IGQ descarta el informe de excepción, restituye el lote actual de mensajes y entra en estado de reintento. Para obtener más información, consulte [“Propiedades específicas de la transferencia a colas dentro del grupo”](#) en la página 232.
- no persistente, el agente de IGQ descarta el informe de excepción y sigue procesando el siguiente mensaje en la cola SYSTEM.QSG.TRANSMIT.QUEUE.

## Seguridad de la transferencia a colas dentro del grupo

En este tema se describen las medidas de seguridad para la transferencia a colas dentro del grupo.

Pueden establecerse los atributos del gestor de colas IGQAUT (autorización IGQ) e IGQUSER (ID de usuario del agente de IGQ) para controlar el nivel de comprobación de seguridad que se debe realizar cuando el agente de IGQ abre las colas de destino.

### **Autorización de la transferencia a colas dentro del grupo (IGQAUT)**

Se puede establecer el atributo IGQAUT para indicar el tipo de comprobaciones de seguridad que deben realizarse y por lo tanto para determinar los ID de usuario que utilizará el agente de IGQ cuando establezca la autorización para poner mensajes en la cola de destino.

El atributo IGQAUT es semejante al atributo PUTAUT disponible en las definiciones de canal.

### **Identificador de usuario de la transferencia a colas dentro del grupo (IGQUSER)**

El atributo IGQUSER puede utilizarse para designar el ID de usuario que utilizará el agente de IGQ cuando establezca la autorización para poner mensajes en una cola de destino.

El atributo IGQUSER es semejante al atributo MCAUSER disponible en las definiciones de canal.

## Propiedades específicas de la transferencia a colas dentro del grupo

Esta sección describe las propiedades específicas de la transferencia a colas dentro del grupo como, por ejemplo, Invalidación de manejadores de objetos, Recuperación automática y Capacidad de reintento del agente de transferencia a colas dentro del grupo, y Agente de transferencia a colas dentro del grupo y serialización.

### **Invalidación de manejadores de objetos (MQRC\_OBJECT\_CHANGED)**

Si los atributos de un objeto han cambiado después de abrir el objeto, el gestor de colas invalida el manejador del objeto con MQRC\_OBJECT\_CHANGED en el próximo uso.

La transferencia a colas dentro del grupo presenta las siguientes reglas para la invalidación de manejadores de objetos:

- Si se ha incluido la cola SYSTEM.QSG.TRANSMIT.QUEUE en la vía de resolución de nombres durante el proceso de apertura porque la transferencia a colas dentro del grupo estaba ENABLED (habilitada) en el momento de la apertura, pero la transferencia a colas dentro del grupo está DISABLED (inhabilitada) en el momento de la transferencia, el gestor de colas invalida el manejador de objetos y la petición de transferencia falla con MQRC\_OBJECT\_CHANGED.
- Si no se ha incluido la cola SYSTEM.QSG.TRANSMIT.QUEUE en la vía de resolución de nombres durante el proceso de apertura porque la transferencia a colas dentro del grupo estaba DISABLED (inhabilitada) en el momento de la apertura, pero la transferencia a colas dentro del grupo está ENABLED (habilitada) en el momento de la transferencia, el gestor de colas invalida el manejador de objetos y la petición de transferencia falla con MQRC\_OBJECT\_CHANGED.
- Si se ha incluido la cola SYSTEM.QSG.TRANSMIT.QUEUE en la vía de resolución de nombres durante el proceso de apertura porque la transferencia a colas dentro del grupo estaba habilitada en el momento de la apertura, pero la definición de SYSTEM.QSG.TRANSMIT.QUEUE ha cambiado en el momento de la

transferencia, el gestor de colas invalida el manejador de objetos y la petición de transferencia falla con MQRC\_OBJECT\_CHANGED.

## Recuperación automática del agente de transferencia a colas dentro del grupo

Si el agente de IGQ finaliza de forma anómala, se emite un mensaje CSQM067E y el agente de IGQ se inicia de nuevo.

## Capacidad de reintento del agente de transferencia a colas dentro del grupo

Si el agente de IGQ se encuentra un problema al acceder a la cola SYSTEM.QSG.TRANSMIT.QUEUE (porque no está definida, por ejemplo, o se ha definido con atributos incorrectos, o está inhibida para recuperaciones, o por algún otro motivo), el agente de IGQ entra en el estado de reintento.

El agente de IGQ respeta un recuento de reintentos e intervalos cortos y largos. Los valores de estos recuentos e intervalos, que no se pueden cambiar, son los siguientes:

<i>Tabla 20. Valores de intervalo y recuentos de reintentos cortos y largos</i>	
<b>Constante</b>	<b>Valor</b>
Número de reintentos cortos	10
Intervalo de reintentos cortos	60 segundos = 1 min
Número de reintentos largos	999.999.999
Intervalo de reintentos largos	1200 segundos = 20 min

## Agente de transferencia a colas dentro del grupo y serialización

Un intento realizado por el agente de IGQ de serializar el acceso a las colas compartidas mientras la recuperación de iguales todavía está en curso podría fallar.

Si hay una anomalía en un gestor de colas de un grupo de compartición de colas mientras el agente de IGQ está abordando mensajes sin confirmar en colas compartidas, el agente de IGQ finaliza y se produce la recuperación de iguales de colas compartidas para el gestor de colas que ha fallado. Debido a que la recuperación de iguales de colas compartidas es una actividad asíncrona, existe la posibilidad de que el gestor de colas que ha fallado, así como el agente de IGQ de ese gestor de colas, se reinicien antes de que finalice la recuperación de iguales de colas compartidas. Lo que a su vez permite que algún mensaje confirmado se procese antes que los mensajes que aún se están recuperando, fuera de secuencia. Para garantizar que los mensajes no se procesan fuera de secuencia, el agente de IGQ serializa el acceso a las colas compartidas emitiendo la llamada de la API MQCONN.

Un intento realizado por el agente de IGQ de serializar el acceso a las colas compartidas mientras la recuperación de iguales todavía está en curso podría fallar. Se emite un mensaje de error y el agente de IGQ se coloca en estado de reintento. Cuando finalice la recuperación de iguales del gestor de colas, por ejemplo, en el momento de realizar el siguiente reintento, se puede iniciar el agente de IGQ.

z/OS

## Gestión de almacenamiento en z/OS

IBM MQ for z/OS requiere estructuras de datos permanentes y temporales y utiliza conjuntos de páginas y almacenamientos intermedios de memoria para almacenar estos datos. Estos temas proporcionan más detalles sobre cómo IBM MQ utiliza estos conjuntos de páginas y almacenamientos intermedios.

### Conceptos relacionados

[“Conjuntos de páginas para IBM MQ for z/OS” en la página 234](#)

Utilice este tema para comprender cómo IBM MQ for z/OS utiliza conjuntos de páginas para almacenar mensajes.

[“Clases de almacenamiento para IBM MQ for z/OS” en la página 235](#)

Una clase de almacenamiento es un concepto de IBM MQ for z/OS que permite al gestor de colas correlacionar colas con conjuntos de páginas. Puede utilizar clases de almacenamiento para controlar qué colas usan qué conjuntos de datos.

[“Almacenamientos intermedios y agrupaciones de almacenamiento intermedio para IBM MQ for z/OS” en la página 236](#)

IBM MQ for z/OS utiliza almacenamientos intermedios y agrupaciones de almacenamiento intermedio para almacenar temporalmente datos en memoria caché. Utilice este tema para comprender mejor cómo se organizan y se utilizan los almacenamientos intermedios.

### Referencia relacionada

[“Dónde encontrar más información sobre la gestión del almacenamiento para IBM MQ for z/OS” en la página 238](#)

Utilice este tema como referencia para buscar más información sobre la gestión de almacenamiento para IBM MQ for z/OS.

## Conjuntos de páginas para IBM MQ for z/OS

Utilice este tema para comprender cómo IBM MQ for z/OS utiliza conjuntos de páginas para almacenar mensajes.

Un *conjunto de páginas* es un conjunto de datos lineal VSAM que ha sido especialmente formateado para ser utilizado por IBM MQ. Los conjuntos de páginas se utilizan para almacenar la mayoría de mensajes y definiciones de objeto.

Las excepciones a esto son las definiciones globales que se almacenan en un repositorio compartido en Db2 y los mensajes en las colas compartidas. Estos no están almacenados en conjuntos de páginas del gestor de colas. Para obtener más información sobre las colas compartidas, consulte [“Colas compartidas y grupos de compartición de colas” en la página 171](#) y para obtener más información sobre las definiciones globales, consulte [Definiciones privadas y globales](#).

Los conjuntos de páginas de IBM MQ pueden tener hasta 64 GB de tamaño. Cada conjunto de páginas se identifica mediante un identificador de conjunto de páginas (PSID), un entero en el rango que va del 00 al 99. Cada gestor de colas debe tener sus propios conjuntos de páginas.

IBM MQ utiliza el conjunto de páginas cero (PSID=00) para almacenar definiciones de objeto y otra información importante relevante para el gestor de colas. Para el funcionamiento normal de IBM MQ es esencial que el conjunto de páginas cero no se llene; por consiguiente, no lo utilice para almacenar mensajes.

Para mejorar el rendimiento del sistema, también debería separar los mensajes de corta duración de mensajes de larga duración colocándolos en conjuntos de páginas distintos.

Debe formatear conjuntos de páginas y IBM MQ proporciona un programa de utilidad FORMAT para ello; consulte [Formato de conjuntos de páginas \(FORMAT\)](#). Los conjuntos de páginas también deben definirse en el subsistema IBM MQ.

IBM MQ for z/OS se puede configurar para expandir un conjunto de páginas dinámicamente si se llena. IBM MQ continúa expandiendo el conjunto de páginas si es necesario hasta que existan 123 extensiones lógicas, si hay suficiente espacio de almacenamiento en disco disponible. Las extensiones pueden expandir volúmenes si el conjunto de datos lineal se define de esta manera, sin embargo, IBM MQ no puede expandir los conjuntos de páginas más allá de 64 GB.

No puede utilizar conjuntos de páginas de un gestor de colas de IBM MQ en un gestor de colas IBM MQ distinto o cambiar el nombre del gestor de colas. Si desea transferir los datos de un gestor de colas a otro, debe descargar todos los objetos y mensajes del primer gestor de colas y volver a cargarlos en el otro.

No se pueden utilizar conjuntos de páginas que superen los 4 GB en un gestor de colas que se ejecute en un release anterior a la V6. Durante el periodo de migración, cuando tenga que volver a un release anterior de código:

- No modifique el conjunto de páginas 0 para que supere los 4 GB.

- Otros conjuntos de páginas superiores a 4 GB quedarán fuera de línea al reiniciar un gestor de colas con un release anterior.

Para obtener más información sobre la migración de conjuntos de páginas existentes capaces de ampliarse más allá de 4 GB, consulte [Definición de un conjunto de páginas para que supere los 4 GB](#).

Es posible para que un administrador pueda añadir dinámicamente conjuntos de páginas a un gestor de colas en ejecución, o para que pueda eliminar conjuntos de páginas de un gestor de colas en ejecución (excepto para el conjunto de páginas cero). El mandato DEFINE PSID sólo se puede ejecutar una vez se haya completado el reinicio del gestor de colas y si el mandato contiene la palabra clave DSN.

## Clases de almacenamiento para IBM MQ for z/OS

Una clase de almacenamiento es un concepto de IBM MQ for z/OS que permite al gestor de colas correlacionar colas con conjuntos de páginas. Puede utilizar clases de almacenamiento para controlar qué colas usan qué conjuntos de datos.

### Introducción a las clases de almacenamiento

Una *clase de almacenamiento* correlaciona una o varias colas con un conjunto de páginas. Ello significa que los mensajes para dicha cola se almacenan en dicho conjunto de páginas.

Las clases de almacenamiento permiten controlar dónde se almacenan datos de mensaje no compartidos con fines administrativos, de espacio de conjunto de datos y de gestión de carga o aislamiento de la aplicación. También puede utilizar clases de almacenamiento para definir el grupo XCF y el nombre del miembro de una región de IMS si utiliza el puente IMS (que se describe en [“IBM MQ y IMS”](#) en la [página 295](#)).

Las colas compartidas no utilizan clases de almacenamiento para obtener una correlación de conjunto de páginas porque los mensajes en ellas no se almacenan en conjuntos de páginas.

### Cómo funcionan las clases de almacenamiento

- Defina una clase de almacenamiento utilizando el mandato DEFINE STGCLASS y especificando un identificador de conjunto de páginas (PSID).
- Cuando defina una cola, especifique la clase de almacenamiento en el atributo STGCLASS.

En el ejemplo siguiente, la cola local QE5 se correlaciona con el conjunto de páginas 21 a través de la clase de almacenamiento ARC2.

```
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE QLOCAL(QE5) STGCLASS(ARC2)
```

Esto quiere decir que los mensajes que se colocan en la cola QE5 se almacenan en el conjunto de páginas 21 (si permanecen en la cola el tiempo suficiente como para ser escritos en el dispositivo de almacenamiento de acceso directo).

Puede utilizar la misma clase de almacenamiento más de una cola, y puede definir como tantas clases de almacenamiento como desee. Por ejemplo, puede ampliar el ejemplo anterior para incluir más clases de almacenamiento definiciones de colas, como se expone a continuación:

```
DEFINE STGCLASS(ARC1) PSID(05)
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE STGCLASS(MAXI) PSID(05)
DEFINE QLOCAL(QE1) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE2) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE3) STGCLASS(MAXI) ...
DEFINE QLOCAL(QE4) STGCLASS(ARC2) ...
DEFINE QLOCAL(QE5) STGCLASS(ARC2) ...
```

En [Figura 71 en la página 236](#), ambas clases de almacenamiento ARC1 y MAXI están asociadas con el conjunto de páginas 05. Por ello, las colas QE1, QE2 y QE3 se correlacionan con el conjunto de páginas 05. De forma parecida, la clase de almacenamiento ARC2 asocia las colas QE4 y QE5 con el conjunto de páginas 21.

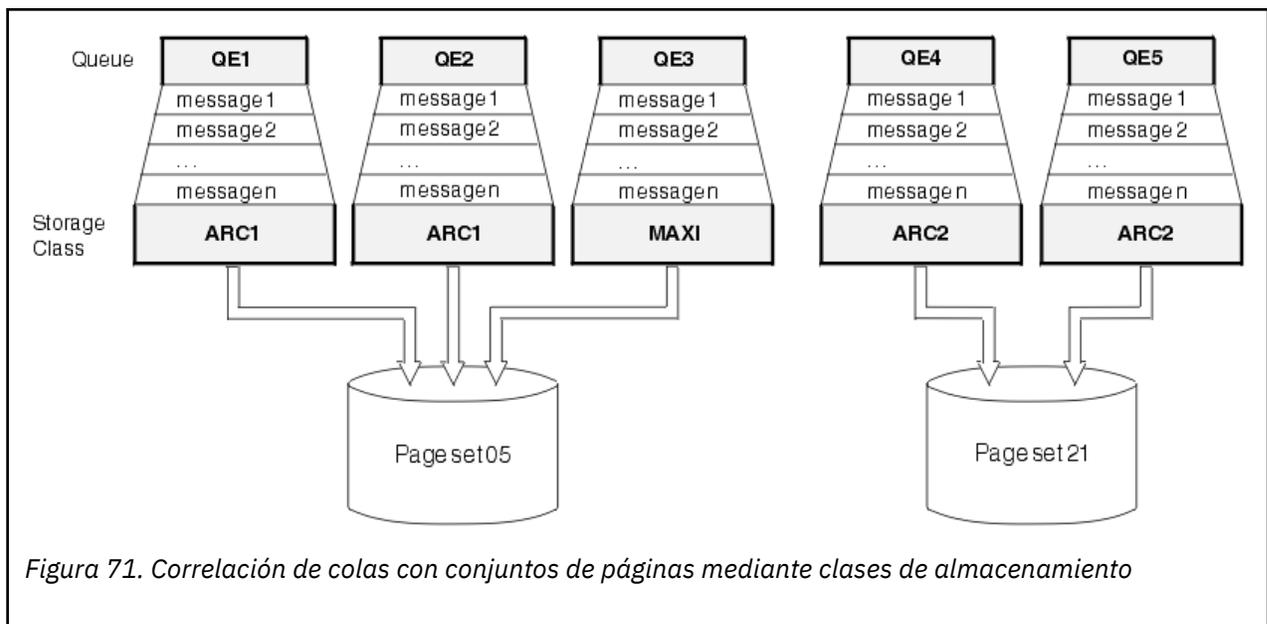


Figura 71. Correlación de colas con conjuntos de páginas mediante clases de almacenamiento

Si se define una cola sin especificar una clase de almacenamiento, IBM MQ utiliza una clase de almacenamiento predeterminada.

Si se pone un mensaje en una cola que nombra una clase de almacenamiento no existente, la aplicación recibe un error. Debe modificar la definición de cola para otorgarle un nombre de clase de almacenamiento existente o crear la clase de almacenamiento denominada por la cola.

Sólo puede cambiar una clase de almacenamiento cuando:

- Todas las colas que utilicen esta clase de almacenamiento estén vacías y no tengan ninguna actividad no confirmada.
- Todas las colas que utilizan esta clase de almacenamiento estén cerradas.

## **z/OS Almacenamientos intermedios y agrupaciones de almacenamiento intermedio para IBM MQ for z/OS**

IBM MQ for z/OS utiliza almacenamientos intermedios y agrupaciones de almacenamiento intermedio para almacenar temporalmente datos en memoria caché. Utilice este tema para comprender mejor cómo se organizan y se utilizan los almacenamientos intermedios.

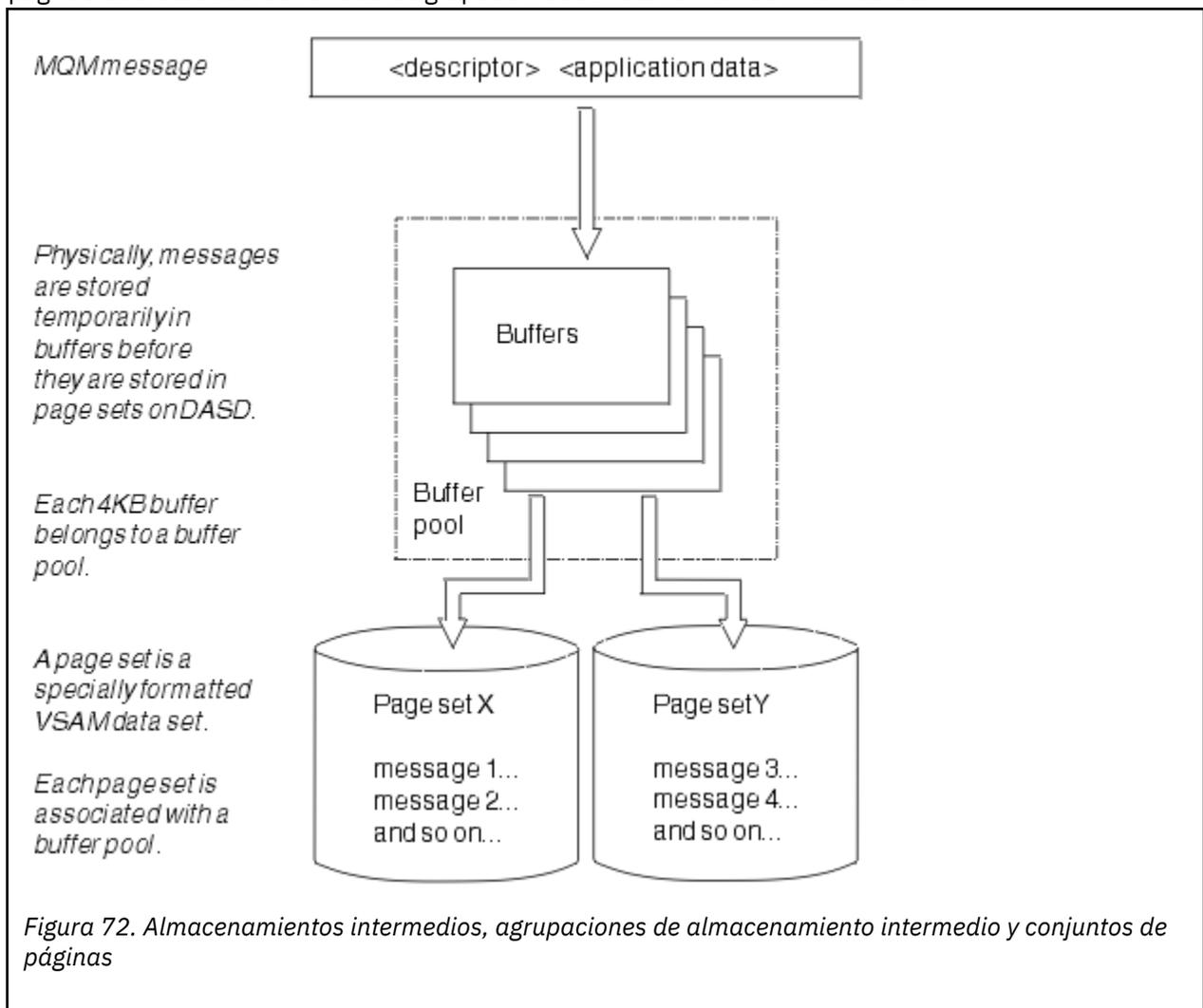
Para mayor eficiencia, IBM MQ utiliza una forma de almacenamiento en memoria caché mediante la cual los mensajes (y definiciones de objeto) se almacenan temporalmente en almacenamientos intermedios antes de almacenarse en conjuntos de páginas en DASD. Los mensajes de corta duración, es decir, los mensajes que se recuperan de una cola poco después de que se reciben, es posible que sólo se almacenen en almacenamientos intermedios. Esta actividad de almacenamiento en memoria caché está controlada por un gestor de almacenamiento intermedio, que es un componente de IBM MQ.

Los almacenamientos intermedios están organizados en *agrupaciones de almacenamiento intermedio*. Puede definir hasta 100 agrupaciones de almacenamiento intermedio (de 0 a 99) para cada gestor de colas.

Se recomienda utilizar el número mínimo de agrupaciones de almacenamientos intermedios coherentes con el objeto y la separación de tipos de mensajes que se describe en [Figura 72 en la página 237](#),

y cualquier requisito de aislamiento de datos que pueda tener la aplicación. Cada almacenamiento intermedio tiene 4 KB de longitud. Las agrupaciones de almacenamientos intermedios utilizan un almacenamiento de 31 bits de forma predeterminada; en este modo, el número máximo de almacenamientos intermedios viene determinado por la cantidad de almacenamiento de 31 bits disponible en el espacio de direcciones del gestor de colas. No utilice más de alrededor del 70% para los almacenamientos intermedios. De forma alternativa, la asignación de almacenamiento de agrupación de almacenamiento intermedio se puede realizar desde un almacenamiento de 64 bits (utilice el atributo LOCATION del mandato **DEFINE BUFFPOOL** ). La utilización de LOCATION(ABOVE) para que se emplee el almacenamiento de 64 bits tiene dos ventajas. En primer lugar, hay mucho más almacenamiento de 64 bits disponible, lo que permite que las agrupaciones de almacenamientos intermedios puedan ser mucho mayores; en segundo lugar, el almacenamiento de 31 bits está a disposición de otras funciones. Normalmente, cuanto más almacenamientos intermedios tenga, más eficiente será el almacenamiento intermedio y mejor el rendimiento de IBM MQ.

Figura 72 en la página 237 muestra la relación entre los mensajes, almacenamientos intermedios, agrupaciones de almacenamientos intermedios y conjuntos de páginas. Una agrupación de almacenamientos intermedios está asociada a uno o varios conjuntos de páginas; cada conjunto de páginas está asociado con una sola agrupación de almacenamientos intermedios.



Puede emitir dinámicamente mandatos para modificar el tamaño y la ubicación de la agrupación de almacenamiento intermedio utilizando el mandato **ALTER BUFFPOOL** . Se pueden añadir conjuntos de páginas dinámicamente utilizando el mandato **DEFINE PSID**, o suprimirse utilizando el mandato **DELETE PSID**.

Si una agrupación de almacenamiento intermedio es demasiado pequeña, IBM MQ emite el mensaje [CSQP020E](#). A continuación, puede añadir dinámicamente más almacenamientos intermedios a la

agrupación de almacenamientos intermedios afectada (tenga en cuenta que es posible que tenga que eliminar almacenamientos intermedios de otras agrupaciones de almacenamientos intermedios para hacerlo).

Especifique el número de almacenamientos intermedios en una agrupación con el mandato **DEFINE BUFFPOOL** y redimensione dinámicamente agrupaciones de almacenamientos intermedios con el mandato **ALTER BUFFPOOL**. Determine el número de almacenamientos intermedios actuales en una agrupación dinámicamente visualizando un conjunto de páginas que utilice la agrupación de almacenamiento intermedio, utilizando el mandato **DISPLAY USAGE**.

Por razones de rendimiento, no coloque mensajes ni definiciones de objeto en la misma agrupación de almacenamiento intermedio. Utilice una agrupación de almacenamiento intermedio (es decir, número cero) sólo y exclusivamente para el conjunto de páginas cero, donde se conservan las definiciones de objeto. Del mismo modo, mantenga los mensajes de corta duración y de larga duración en distintas agrupaciones de almacenamiento intermedio y, por ello, en diferentes conjuntos de páginas y colas.

El mandato **DEFINE BUFFPOOL** puede ser utilizado después del reinicio para crear una nueva agrupación de almacenamientos intermedios. En su lugar, si un mandato **DEFINE PSID** utiliza la palabra clave DSN, puede identificar explícitamente una agrupación de almacenamiento intermedio que no está definida actualmente. A continuación se creará la nueva agrupación de almacenamiento intermedio.

## z/OS **Dónde encontrar más información sobre la gestión del almacenamiento para IBM MQ for z/OS**

Utilice este tema como referencia para buscar más información sobre la gestión de almacenamiento para IBM MQ for z/OS.

Puede encontrar más información sobre los temas de esta sección en las siguientes fuentes:

<i>Tabla 21. Dónde encontrar más información sobre la gestión de almacenamiento</i>	
<b>Tema</b>	<b>Dónde buscar</b>
Cantidad de almacenamiento que necesita	<a href="#">Planificación de los requisitos de almacenamiento y rendimiento en z/OS</a>
Qué tamaño utilizar para los conjuntos de páginas y las agrupaciones de almacenamientos intermedios	<a href="#">Planifique los conjuntos de páginas y agrupaciones de almacenamiento intermedio</a>
Gestión de conjuntos de páginas	<a href="#">Gestión de conjuntos de páginas</a>
Mandatos MQSC	<a href="#">Los mandatos MQSC</a>

## z/OS **Registro en IBM MQ for z/OS**

IBM MQ mantiene *registros* de cambios de datos y sucesos significativos a medida que se producen. Estos registros se pueden utilizar para recuperar los datos a un estado anterior si es necesario.

El *conjunto de datos de rutina de carga* (BSDS) almacena información sobre los conjuntos de datos que contienen los registros.

El registro no contiene información para estadísticas, rastreos o evaluación del rendimiento. Para obtener más detalles sobre la información estadística y de supervisión que IBM MQ recopila, consulte [Supervisión y estadísticas](#).

Para obtener más información sobre el registro cronológico, consulte los temas siguientes :

- [“Archivos de registro en IBM MQ for z/OS” en la página 239](#)
- [“Cómo se estructura el registro” en la página 243](#)

- [“Cómo se graban los registros de IBM MQ for z/OS” en la página 244](#)
- [“RBA \(Dirección de byte relativa\) de registro mayor” en la página 247](#)
- [“Conjunto de datos de arranque” en la página 248](#)

### Tareas relacionadas

[Planificación del entorno de registro cronológico](#)

[Establecer registros utilizando el módulo de parámetros del sistema](#)

 [Administración de z/OS](#)

### Referencia relacionada

 [Mensajes para IBM MQ for z/OS](#)

## Archivos de registro en IBM MQ for z/OS

Los archivos de registro contienen la información necesaria para la recuperación de transacciones. Los archivos de anotaciones activos pueden ser archivados para que puedan mantener los datos de registro durante un largo periodo.

### Qué es un archivo de registro

IBM MQ registra todos los sucesos significativos a medida que se producen en un *registro activo*. El registro contiene la información necesaria para recuperar:

- Mensajes persistentes
- Objetos de IBM MQ, como, por ejemplo, colas
- El gestor de colas de IBM MQ

El registro activo consta de una colección de conjuntos de datos (hasta 310) que se utilizan de forma cíclica.

Puede habilitar el archivado de registros de modo que cuando un registro activo se llene, se realice una copia en un conjunto de datos de archivo. La utilización del archivo permite mantener datos de registro para un periodo ampliado. Si no utiliza el archivado, los registros se resetean y los datos anteriores se sobrescriben. Para recuperar un conjunto de páginas o recuperar datos de una estructura CF, necesita datos de registro de cuando se ha tomado la copia de seguridad del conjunto de páginas o estructura. Un registro de archivado puede crearse en disco o en cinta.

### Archivado

Dado que el registro activo tiene un tamaño fijo, IBM MQ copia el contenido de cada conjunto de datos de registro periódicamente en un *registro de archivado*, que normalmente es un conjunto de datos en un dispositivo de almacenamiento de acceso directo (DASD) o una cinta magnética. Si hay un error de subsistema o transacción, IBM MQ utiliza el registro activo y, si es necesario, el registro de archivado para la recuperación.

El registro de archivado puede contener hasta 1.000 conjuntos de datos secuenciales. Puede catalogar cada conjunto de datos utilizando Integrated Catalog Facility (recurso de catálogo integrado) de z/OS.

El archivado es un componente esencial de la recuperación de IBM MQ. Si una unidad de recuperación es de larga ejecución, los registros de archivo dentro de dicha unidad de recuperación se pueden encontrar en el registro de archivado. En este caso, la recuperación requiere datos del registro de archivado. Sin embargo, si el archivado está inhabilitado, se recorta el registro activo con los nuevos registros de anotaciones sobrescribiendo los registros de anotaciones anteriores. Esto significa que es posible que IBM MQ no pueda restituir la unidad de recuperación y que se pierdan mensajes. A continuación, el gestor de colas termina de forma anómala.

Por lo tanto, en un entorno de producción, **no desactive nunca el archivo**. Si lo hace, corre el riesgo de perder datos tras una anomalía del sistema o transacción. Sólo puede considerar el desactivado del

archivo si realiza la ejecución en un entorno de prueba. Si necesita hacerlo, utilice la macro CSQ6LOGP, que está explicada en [Utilización de CSQ6LOGP](#).

Para ayudar a prevenir problemas con unidades de trabajo de larga ejecución no planificadas, IBM MQ emite un mensaje ([CSQJ160I](#) o [CSQJ161I](#)) si se detecta una unidad de trabajo de larga ejecución durante el proceso de descarga de registro activo.

## Registro dual

En el registro dual, cada registro dual se escribe en dos conjuntos de datos de registro activo distintos para minimizar la probabilidad de tener problemas de pérdida de datos durante el reinicio.

Puede configurar IBM MQ para que se ejecute con unas *anotaciones simples* o con un *registro dual*. Con una registro simple, los registros de anotaciones se escriben una única vez en un conjunto de datos de registro activo. Cada conjunto de datos de registro activo es un conjunto de datos lineal (LDS) VSAM de un solo alcance. Con el registro dual, cada registro dual se escribe en dos conjuntos de datos de registro activo distintos. El registro dual minimiza la probabilidad de tener problemas de pérdida de datos durante el reinicio.

## Derivación de registro

La derivación de registro hace que los registros de anotaciones de algunas unidades de trabajo se escriban más abajo del registro. Esto reduce la cantidad de datos de registro que deben leerse al reiniciar el gestor de colas, o restituirlo, para unidades de trabajo pendientes de larga ejecución o de largo plazo.

Cuando una unidad de trabajo se considera que es larga, se graba una representación de cada registro de anotaciones más abajo en el registro. Esta técnica se conoce como *derivación*. Cuando se ha procesado toda la unidad de trabajo, la unidad de trabajo pasa al estado *derivada*. Cualquier actividad de copia de seguridad o reinicio relacionada con la unidad de trabajo derivada puede utilizar registros de anotaciones derivados en lugar de utilizar los registros de anotaciones de la unidad de trabajo originales.

La detección de una unidad de trabajo de larga ejecución es una función del proceso de punto de comprobación. Durante el punto de comprobación, se comprueba cada unidad de trabajo activa para establecer si necesita ser derivada. Si la unidad de trabajo ha pasado por dos puntos de comprobación anteriores desde su creación, o desde su última derivación, la unidad de trabajo será adecuada para ser derivada. Esto significa que una sola unidad de trabajo puede ser derivada más de una vez. Se conoce como una unidad de trabajo *de varias derivaciones*.

Una unidad de trabajo se desvía cada tres puntos de comprobación. Sin embargo, el punto de comprobación se realiza de asíncrona para el conmutador de registro (o la grabación del registro que ha provocado la superación de LOGLOAD).

Solo un único punto de comprobación tiene lugar en cada momento, por lo tanto pueden haber varios conmutadores de registro antes de que se complete el punto de comprobación.

Esto significa que si no hay suficientes registros activos o si son demasiado pequeños, es posible que la derivación de una unidad de trabajo de gran tamaño no se complete antes de que se llenen todos los registros.

Se genera el mensaje [CSQR027I](#) si no se puede completar la derivación.

Si se ha desactivado el archivado de registros, se genera ABEND 5C6 con la razón 00D1032A si se intenta restituir la unidad de trabajo para la que ha fallado la derivación. Para evitar este problema debe utilizar OFFLOAD=YES.

La derivación de registros siempre está activa y se ejecuta tanto si el archivo de registro está habilitado como si no.

**Nota:** Aunque se derivan todos los registros de anotaciones para una unidad de trabajo, el contenido completo de cada registro no se deriva, sólo la parte que es necesaria para la restitución. Esto significa que la cantidad de datos de registro grabados se mantiene al mínimo, y que los registros derivados no se pueden utilizar si se produce una anomalía en el conjunto de páginas. Una unidad de trabajo de larga

ejecución es aquella que se ha estado ejecutando durante más de tres puntos de control de gestor de colas.

Para obtener más información acerca de la derivación del registro, consulte [Gestión de los registros](#).

## Compresión de anotaciones

Puede configurar IBM MQ for z/OS para comprimir y descomprimir los registros de anotaciones, ya que se escriben y leen desde el conjunto de datos de registro.

La compresión del registro se puede utilizar para reducir la cantidad de datos escrito en el registro para mensajes permanentes en colas privadas. La cantidad de compresión que se consiga dependerá del tipo de datos contenidos en los mensajes. Por ejemplo, la codificación de longitud de ejecución (RLE) funciona instancias repetidas de bytes que pueden dar buenos resultados para datos estructurados u orientados a registros.



**Atención:** Los mensajes persistentes que se están colocando en una cola compartida no están sujetos a la compresión del registro.

Puede utilizar campos de la sección Gestor de registros de los registros del Recurso de gestión del sistema 115 (SMF) para supervisar la cantidad de compresión de datos que se ha logrado. Para obtener más información sobre SMF, consulte [Utilización del recurso de gestión del sistema](#) y [Mensajes contables y estadísticos](#).

La compresión del registro aumenta la utilización del procesador del sistema. Sólo debería considerar utilizar la compresión si el rendimiento del gestor de colas está restringido por el ancho de banda de E/S al escribir en los conjuntos de datos de registro o si está restringido por el almacenamiento de disco necesario para mantener los conjuntos de datos de registro. Si está utilizando colas compartidas, el ancho de banda de entrada/salida se puede aligerar añadiendo gestores de cola adicionales al grupo de compartición de cola y distribuyendo la carga de trabajo entre varios gestores de colas.

La opción de compresión de registros puede habilitarse e inhabilitarse, según sea necesario, sin necesidad de detener y reiniciar el gestor de colas. El gestor de colas puede leer todos los registros de anotaciones comprimidos independientemente del valor de la compresión del registro actual.

El gestor de colas admite 3 valores para la compresión de registros.

### NINGUNO

No se utiliza ninguna compresión de datos de registro. Éste es el valor predeterminado.

### RLE

La compresión de datos de registro se realiza utilizando la codificación de longitud de ejecución (RLE).

### CUALQUIERA

Permite que el gestor de colas seleccione el algoritmo de compresión que indica el grado más alto de la compresión del registro de anotaciones. Esta opción da como resultado una compresión RLE.

Puede controlar la compresión de registros de anotaciones utilizando una de las siguientes posibilidades:

- Los mandatos SET y DISPLAY LOG en el MQSC; consulte [SET LOG](#) y [DISPLAY LOG](#)
- Las funciones Establecer registro y Consultar registro en la interfaz de formato de mandato programable (PCF); consulte [Establecer registro](#) y [Consultar registro](#)
- La macro CSQ6LOGP en el módulo de parámetros de sistema; consulte [Utilización de CSQ6LOGP](#)

Además, el programa de utilidad Impresión de registro CSQ1LOGP admite la expansión y la compresión de registros de anotaciones.

## Datos de registro

El registro puede contener hasta 18 millones de millones ( $1,8 \times 10^{19}$ ) de bytes. Cada byte puede ser direccionado por su desplazamiento desde el comienzo del registro, y dicho desplazamiento se conoce como su *dirección de byte relativa* (RBA).

La dirección de byte relativa (RBA) está referenciada por un campo de 6 u 8 bytes, lo que da un rango direccionable total de  $2^{48}$  bytes o  $2^{64}$  bytes, según si se utilizan RBA de registro de 6 u 8 bytes.

Sin embargo, cuando IBM MQ detecta que el rango utilizado está más allá de F00000000000 (si las RBA de 6 bytes están en uso) o FFFF800000000000 (si las RBA de registro de 8 bytes están en uso), se emiten los mensajes [CSQI045](#), [CSQI046](#), [CSQI047](#) y [CSQJ032](#), avisándole para que restablezca la RBA de registro.

Si el valor de RBA alcanza FFF800000000 (si se utilizan RBA de registro de 6 bytes) o FFFFFFFC0000000000 (si se utilizan RBA de registro de 8 bytes), el gestor de colas finaliza con el código de razón [00D10257](#).

Una vez que se emiten los mensajes de aviso acerca del rango de registro utilizado, debe planificar una interrupción del gestor de colas durante el cual el gestor de colas puede convertirse para utilizar RBA de registro de 8 bytes o puede restablecerse el registro. El procedimiento para restablecer el registro está documentado en [Restablecimiento del registro del gestor de colas](#).

Si el gestor de colas utiliza RBA de registro de 6 bytes, considere convertir el gestor de colas para que utilice RBA de registro de 8 bytes en lugar de restablecer el registro del gestor de colas siguiendo el proceso descrito en [Implementación de la dirección de byte relativa de registro más grande](#).

El registro consta de *registros de anotaciones*, cada uno de los cuales es un conjunto de datos de registro que se utiliza como una única unidad. Un registro de anotaciones se identifica mediante la dirección de byte relativa (RBA) del primer byte de su cabecera o mediante su número de secuencia de registro (LRSN). La dirección de byte relativa o el número de secuencia de registro sólo identifican un registro que empieza por un punto concreto del registro.

Si utiliza la dirección de byte relativa o si utiliza el número de secuencia de registro para identificar puntos de registro dependerá de si se utilizan de grupos de compartición de colas. En un entorno de compartición de cola, no puede utilizar la dirección de byte relativa para identificar un punto de registro unívocamente, porque varios gestores de colas pueden actualizar la misma cola al mismo tiempo y cada uno tiene su propio registro. Para resolver esto, el número de secuencia de registro se deriva de un valor de indicación de fecha y hora y no representa necesariamente el desplazamiento físico del registro de anotaciones en el registro.

Cada registro de anotaciones tiene una cabecera que da su tipo, el subcomponente IBM MQ que ha realizado el registro y, para la unidad de registros de recuperación, un identificador de unidad de recuperación.

Hay cuatro tipos de registros de anotaciones descritos bajo las siguientes cabeceras:

- [Registros de anotaciones de unidad de recuperación](#)
- [Registros de punto de comprobación](#)
- [Registros de control de conjunto de páginas](#)
- [Registros de copia de seguridad de estructura CF](#)

## Registros de anotaciones de unidad de recuperación

La mayoría de los registros de anotaciones describen cambios en las colas de IBM MQ. Todos estos cambios se realizan dentro de unidades de recuperación.

IBM MQ utiliza técnicas de registro especiales que implican *deshacer/rehacer* y la *compensación de registros de anotación* para reducir los tiempos de reinicio y mejorar la disponibilidad del sistema.

Un efecto de esto es que se limita el tiempo de reinicio. Si se produce una anomalía durante un reinicio de modo que el gestor de colas tenga que reiniciarse una segunda vez, toda la actividad de recuperación que se haya completado en el punto de la anomalía en el primer reinicio no necesitará volver a aplicarse durante un segundo reinicio. Esto significa que los sucesos reinicios no tardan cada vez más tiempo en completarse.

## Registros de punto de comprobación

Para reducir el tiempo de reinicio, IBM MQ toma puntos de comprobación periódicos durante la operación normal. Esto se produce de la siguiente manera:

- Cuando se ha escrito un número predefinido de registros de anotaciones. Se trata del número definido por el operando de frecuencia del punto de comprobación denominado LOGLOAD de la macro del parámetro de sistema CSQ6SYSP, descrito en [Utilización de CSQ6SYSP](#).
- Al final de un reinicio satisfactorio.
- Durante una terminación normal.
- Siempre que IBM MQ cambie al siguiente conjunto de datos de registro activo en el ciclo.

En el momento en que se toma un punto de comprobación, IBM MQ emite el mandato DISPLAY CONN (descrito en [DISPLAY CONN](#)) internamente para que una lista de conexiones actualmente dudosa se grabe en el registro de la consola de z/OS.

## Registros de control de conjunto de páginas

Estos registros registran los conjuntos de páginas y las agrupaciones de almacenamiento intermedio conocidas por el gestor de colas de IBM MQ en cada punto de comprobación y registran información sobre los rangos de registro necesarios para realizar la recuperación de soportes del conjunto de páginas en el momento del punto de comprobación.

Algunos cambios dinámicos en conjuntos de páginas y agrupaciones de almacenamiento intermedio también se pueden escribir como registros de control del conjunto de páginas, a fin de que los cambios se puedan recuperar y reintegrar automáticamente durante el siguiente reinicio del gestor de colas.

## Registros de copia de seguridad de estructura CF

Estos registros contienen datos leídos de una estructura de lista del recurso de acoplamiento en respuesta a un mandato BACKUP CFSTRUCT. En el improbable caso de que se produzca una anomalía en la estructura del recurso de acoplamiento, estos registros se utilizan, junto con la unidad de los registros de recuperación, por parte del mandato RECOVER CFSTRUCT para realizar la recuperación del soporte de la estructura del recurso de acoplamiento en el punto de la anomalía.

### Tareas relacionadas

[Implementación de la RBA \(Relative Byte Address\)](#)

## **Cómo se estructura el registro**

Utilice este tema para comprender la terminología utilizada para describir registros de anotaciones.

Cada conjunto de datos de registro activo debe ser un conjunto de datos lineal (LDS) VSAM. La unidad de salida física grabada en el conjunto de datos del registro activo es un intervalo de control (CI) de 4 KB. Cada intervalo de control contiene un registro VSAM.

## Registros de anotaciones físicos y lógicos

Un intervalo de control VSAM es un registro *físico*. La información registrada en un momento determinado forma un registro *lógico* con una longitud que cambia independientemente del espacio disponible en el intervalo de control. Por lo tanto, un registro físico puede contener:

- Varios registros lógicos
- Uno o varios registros lógicos y parte de otro registro lógico
- Sólo parte de un registro lógico

El término *registro de anotaciones* hace referencia al registro *lógico*, independientemente de la cantidad de registros *físicos* que sean necesarios para almacenarlo.

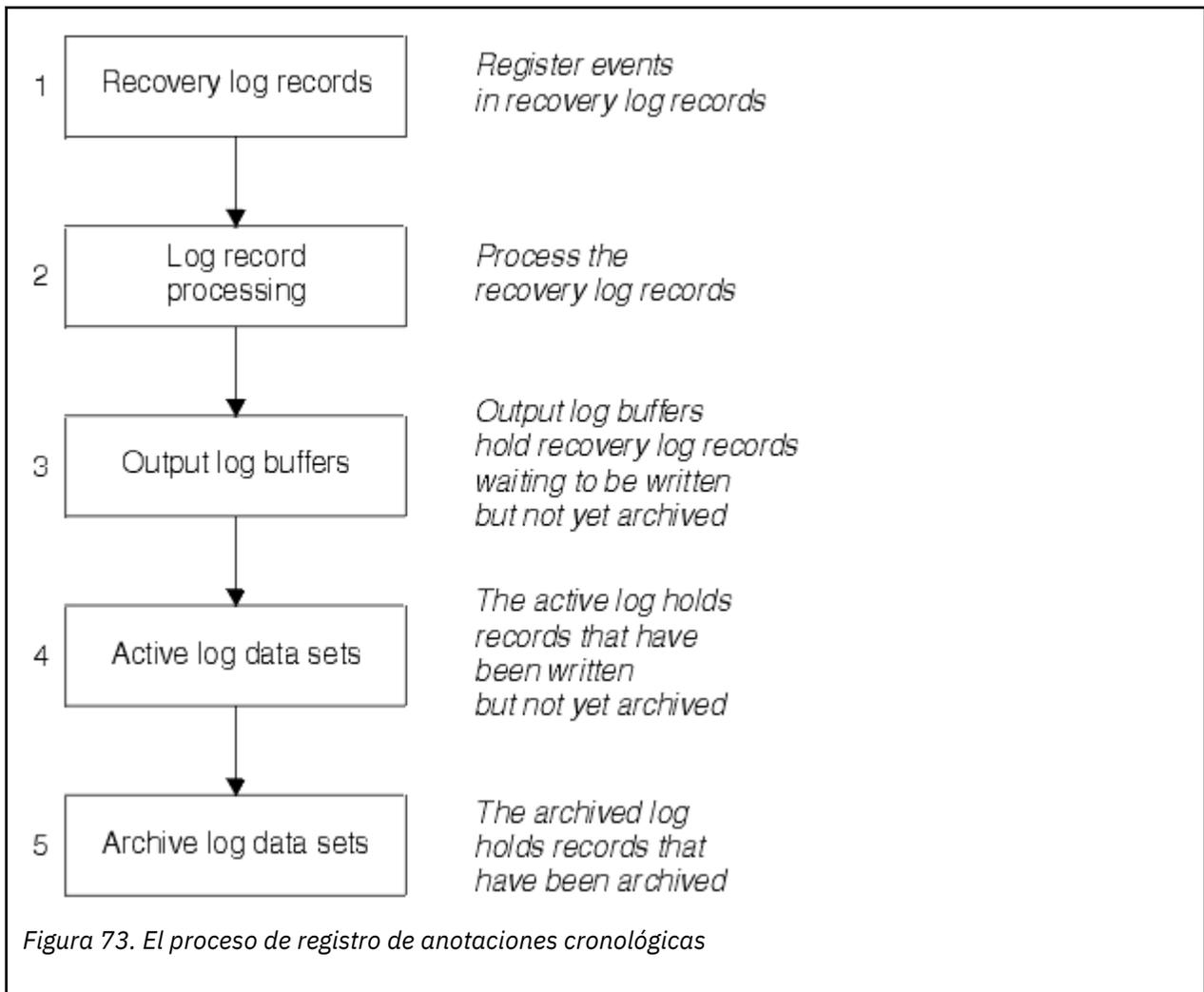
## z/OS **Cómo se graban los registros de IBM MQ for z/OS**

Utilice este tema para comprender cómo IBM MQ procesa los registros cronológicos de instalación.

IBM MQ graba cada registro de anotaciones en un conjunto de datos DASD denominado *registro activo*. Cuando el registro activo está lleno, IBM MQ copia su contenido en un conjunto de datos de DASD o cinta denominado *registro de archivado*. Este proceso se denomina *descarga*.

Figura 73 en la página 244 ilustra el proceso de registro de anotaciones cronológicas. Los registros de anotaciones normalmente giran en torno al siguiente ciclo:

1. IBM MQ anota los cambios en los datos y los sucesos significativos en los registros de recuperación.
2. IBM MQ procesa los registros de recuperación y los divide en segmentos, si es necesario.
3. Los registros de anotaciones se colocan secuencialmente en *almacenamientos intermedios de registro de salida*, que están formateados como intervalos de control (CI) VSAM. Cada registro de anotaciones se identifica mediante una dirección de byte relativa en el rango del cero al  $2^{64}-1$ .
4. Los intervalos de control se escriben en un conjunto de conjuntos de datos de registro activo (DASD) predefinidos, que se utilizan secuencialmente y se reciclan.
5. Si el archivado está activo, cuando cada conjunto de datos de registro activo se llena, su contenido se descarga automáticamente en un nuevo conjunto de datos de registro de archivado.



### **Cuando se escribe el registro activo**

Los almacenamientos intermedios de registro en almacenamiento se escriben en un conjunto de datos de registro activo siempre que produce una de las siguientes situaciones:

- Los almacenamientos intermedios de registro se llenan
- Se alcanza el umbral de escritura (como se especifica en la macro CSQ6LOGP).
- Se producen determinados sucesos significativos, como por ejemplo un punto de confirmación, o cuando se emite un mandato IBM MQ BACKUP CFSTRUCT.

Cuando se inicializa el gestor de colas, los conjuntos de datos de registro activos especificados en el BSDS se asignan dinámicamente para el uso exclusivo del gestor de colas y permanecen asignados exclusivamente a IBM MQ hasta que el gestor de colas termine.

## Adición dinámica de conjuntos de datos de registro

Se pueden definir dinámicamente conjuntos de datos de registro activo nuevos mientras se está ejecutando el gestor de colas. Esta característica mitiga el problema de que se cuelgue el gestor de colas cuando el archivado no puede descargar los registros activos debido a un problema transitorio. Consulte el mandato [DEFINE LOG](#) para obtener más información.

**Nota:** Para definir o eliminar registros activos, debe finalizar y reiniciar el gestor de colas.

## IBM MQ y subsistema de gestión de almacenamiento

Los parámetros de IBM MQ permiten especificar clases de almacenamiento del subsistema de gestión de almacenamiento (MVS/DFP SMS) cuando se asignan dinámicamente conjuntos de datos de registro de archivado de IBM MQ. IBM MQ inicia el archivado de conjunto de datos de registro, pero puede utilizar SMS para realizar la asignación del conjunto de datos de archivado.

### Referencia relacionada

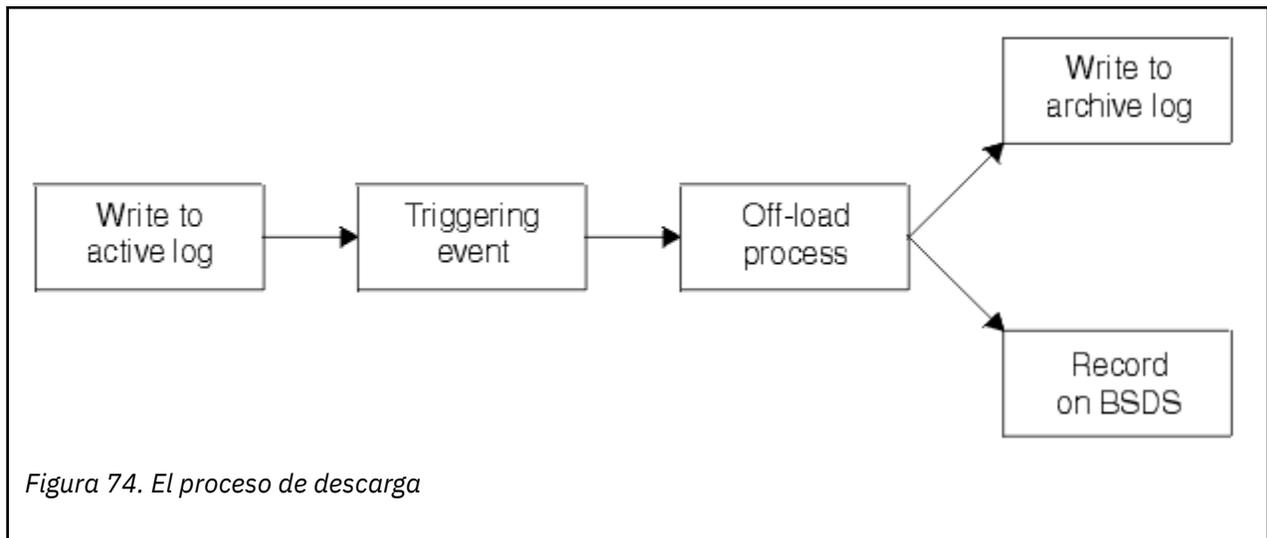
“Cuando se graba el registro de archivado de IBM MQ for z/OS” en la página 245

Utilice este tema para obtener información sobre el proceso de copia de registros activos en registros de archivado, y acerca de cuándo se produce el proceso.

### **Cuando se graba el registro de archivado de IBM MQ for z/OS**

Utilice este tema para obtener información sobre el proceso de copia de registros activos en registros de archivado, y acerca de cuándo se produce el proceso.

El proceso de copiar registros activos en registros de archivado se denomina *descarga*. La relación de descarga en otros sucesos de anotaciones se muestra esquemáticamente en [Figura 74](#) en la página 246.



## Desencadenamiento del proceso de descarga

El proceso de descarga de un registro activo en un registro de archivado puede desencadenarse mediante varios sucesos. Por ejemplo:

- Llenando un conjunto de datos de registro activo.
- Utilizando el mandato MQSC ARCHIVE LOG.
- Se ha producido un error al escribir en un conjunto de datos de registro activo.

El conjunto de datos se trunca antes del punto de anomalía, y el registro que no se ha escrito pasa a ser el primer registro del nuevo conjunto de datos. La descarga se desencadena para el conjunto de datos truncado tal como lo haría para un conjunto de datos de registro completo normal. Si existen registros activos duales, ambas copias se truncan de forma que las dos copias siguen sincronizadas.

El mensaje CSQJ110E se emite cuando el último registro activo disponible está un 5% lleno y, en los sucesivos incrementos del 5%, va indicando el porcentaje de capacidad del registro que se está utilizando. Si todos los registros activos se llenan, IBM MQ detiene el proceso, hasta que se produce la descarga, y emite este mensaje:

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

## El proceso de descarga

Cuando todos los registros activos se llenan, IBM MQ ejecuta el proceso de descarga y detiene el proceso hasta que se haya completado el proceso de descarga. Si el proceso de descarga falla cuando los registros activos están llenos, IBM MQ termina de forma anómala.

Cuando un registro activo está preparado para descargarse, se envía una solicitud al operador de la consola de z/OS para montar una cinta o preparar una unidad DASD. El valor de la opción de registro ARCWTOR (para obtener más información, consulte [Utilización de CSQ6ARVP](#)) determina si se ha recibido la solicitud. Si está utilizando una cinta para la descarga, especifique ARCWTOR=YES. Si el valor es YES, la solicitud viene precedida por un WTOR (número de mensaje CSQJ008E) que indica al operador que prepare un conjunto de datos de registro de archivado que deba ser asignado.

No es necesario que el operador responda a este mensaje inmediatamente. No obstante, retardar la respuesta retarda el proceso de descarga. No afecta al rendimiento de IBM MQ a menos que el operador haya retardado la respuesta durante tanto tiempo que IBM MQ se quede sin registros activos.

El operador puede responder cancelando el proceso de descarga. En este caso, si la asignación es para la primera copia de conjuntos de datos de archivo duales, el proceso de descarga simplemente se retarda hasta que se llena el siguiente conjunto de datos de archivo. Si la asignación es para la segunda copia, el proceso de archivado conmuta a la modalidad de una sola copia, pero sólo para este conjunto de datos.

## Interrupciones y errores durante la descarga

Una solicitud para detener al gestor de colas no tendrá efecto hasta que termine el proceso de descarga. Si IBM MQ falla mientras la descarga está en curso, la descarga empieza de nuevo cuando se reinicia el gestor de colas.

## Mensajes durante el proceso de descarga

Los mensajes descargados se envían a la consola de z/OS mediante IBM MQ y el proceso de descarga. Puede utilizar estos mensajes para buscar los rangos de RBA en los distintos conjuntos de datos de registro.

### RBA (Dirección de byte relativa) de registro mayor

Esta función mejora la disponibilidad del gestor de colas al aumentar el periodo de tiempo antes de restablecer el registro.

Los datos de recuperación se graban en el registro para que los mensajes permanentes estén disponibles cuando se reinicie el gestor de colas. El término dirección de byte relativa de registro (RBA de registro) se utiliza para hacer referencia a la ubicación de los datos como desplazamiento desde el principio del registro.

Antes de IBM MQ 8.0, la RBA de registro de 6 bytes podía direccionar hasta 256 terabytes de datos. Antes de que se grabe esta cantidad de registros de anotaciones, debe restablecer las anotaciones del gestor de colas siguiendo el procedimiento que se describe en [Restablecimiento de las anotaciones del gestor de colas](#).

El restablecimiento de los registros de los gestores de colas no es un proceso rápido y puede requerir un tiempo de interrupción ampliado, debido a la necesidad de restablecer los conjuntos de páginas como parte del proceso. Para un gestor de colas de alta utilización, esta operación puede realizarse generalmente una vez al año.

Desde IBM MQ 8.0, la RBA de registro puede tener 8 bytes de longitud y ahora el gestor de colas puede direccionar más de 64.000 veces más datos (16 exabytes) antes de que se restablezca la RBA de registro. El impacto de utilizar la RBA de registro más grande es que el tamaño de los datos de registro grabados aumenta en unos cuantos bytes.

## ¿Cuándo se habilita esta función?

 Los gestores de colas creados en IBM MQ 9.3.0 o versiones posteriores ya tienen esta función habilitada.

Si la RBA de registro actual se está acercando al final del rango de RBA de registro, considere convertir el gestor de colas para utilizar una RBA de registro de 8 bytes en lugar de restablecer el registro del gestor de colas. Convertir un gestor de colas para utilizar RBA de registro de 8 bytes requiere una interrupción menor que restablecer el registro, y aumenta significativamente el periodo de tiempo antes de restablecer el registro.

El mensaje `CSQJ034I`, emitido durante la inicialización del gestor de colas, indica el final del rango de RBA de registro para el gestor de colas tal como se ha configurado y se puede utilizar para determinar si se están utilizando RBA de registro de 6 u 8 bytes.

## ¿Cómo se habilita esta función?

La RBA de registro de 8 bytes se habilita al iniciar el gestor de colas con un BSDS en formato de la versión 2. En resumen, esto se consigue mediante:

1. Comprobando que todos los gestores de colas en el grupo de compartición de colas satisfagan los requisitos para habilitar la RBA de registro de 8 bytes
2. El cierre correcto del gestor de colas

3. La ejecución del [programa de utilidad de conversión BSDS](#) para crear una copia del BSDS en formato de la versión 2.
4. El reinicio del gestor de colas con el BSDS convertido.

Una vez que un gestor de colas se ha convertido para utilizar RBA de registro de 8 bytes, no puede volver a utilizar RBA de registro de 6 bytes.

Consulte [Implementación de la dirección de byte relativa de registro más grande](#) para conocer el procedimiento detallado sobre cómo habilitar las direcciones de byte relativo de registro de 8 bytes.

### Tareas relacionadas

[Planificación del aumento del rango máximo de registros direccionables](#)

### Referencia relacionada

[El programa de utilidad de conversión BSDS \(CSQJUCNV\)](#)

## Conjunto de datos de arranque

IBM MQ requiere el conjunto de datos de rutina de carga como mecanismo para hacer referencia a conjuntos de datos de registro y registros de anotaciones. Esta información es necesaria durante el proceso normal y la recuperación de reinicio.

### Para qué es el conjunto de datos de arranque

El *conjunto de datos de arranque* (BSDS) es un conjunto de datos de secuencia de claves VSAM (KSDS) que contiene la información necesaria para IBM MQ. Contiene lo siguiente:

- Un inventario de todos los conjuntos de datos de registro de archivado y conjuntos de datos de registro activos conocidos en IBM MQ. IBM MQ utiliza este inventario para:
  - Realizar un seguimiento de los conjuntos de datos de registro de archivado y conjuntos de datos de registro activos
  - Ubicar registros de anotaciones de forma que pueda satisfacer a las solicitudes de lectura de registro durante el proceso normal
  - Ubicar registros de anotaciones de forma que pueda manejar el proceso de reinicio

IBM MQ almacena información en el inventario cada vez que se define un conjunto de datos de registro de archivado o se reutiliza un conjunto de datos de registro activo. En los registros activos, el inventario muestra cuáles están vacíos y cuales están disponibles para volver a ser utilizados. El inventario contiene la dirección de byte relativa (RBA) de cada parte del registro mantenido en el conjunto de datos en cuestión.

- Un inventario de *ajuste automático* de toda la actividad reciente de IBM MQ. Resulta necesario si tiene que reiniciar el gestor de colas.

El conjunto de datos de arranque (BSDS) es necesario si el gestor de colas tiene un error y tiene que reiniciarlo. IBM MQ **debe** tener un BSDS. Para minimizar la probabilidad de problemas durante un reinicio, puede configurar IBM MQ con BSDS duales, cada uno de los cuales graba la misma información. La utilización de BSDS duales se conoce como ejecución en *modalidad dual*. Si es posible, coloque las copias en volúmenes separados. Ello reduce el riesgo de que se pierdan ambos si el volumen está dañado o destruido. Utilice el BSDS dual en lugar de la escritura dual en DASD.

El BSDS se configura cuando se personaliza IBM MQ y el inventario se puede gestionar utilizando el programa de utilidad de inventario de cambios (CSQJU003). Para obtener más información sobre este programa de utilidad, consulte [Administrar IBM MQ for z/OS](#). Una sentencia DD hace referencia al mismo en el procedimiento de arranque del gestor de colas.

Normalmente, IBM MQ mantiene copias duplicadas del BSDS. Si se produce un error de E/S, anula la asignación de la copia anómala y sigue con un único BSDS. Puede restaurar la operación en modalidad dual, tal como se describe en [Administrar IBM MQ for z/OS](#).

Los registros activos se registran por primera vez en el BSDS cuando se instala IBM MQ. No puede sustituir los registros activos sin terminar y reiniciar el gestor de colas.

Los conjuntos de datos de registro de archivado se asignan automáticamente. Cuando se asigna uno, el nombre del conjunto de datos se registra en el BSDS. La lista de conjuntos de datos de registro de archivado se expande a medida que se van añadiendo archivos y se ajusta cuando se alcanza un número de entradas determinado por el usuario. El número máximo de entradas es de 1.000 para el registro de archivado individual y de 2.000 para el registro dual.

Puede utilizar un sistema de gestión de cintas para suprimir los conjuntos de datos de registro de archivado ( IBM MQ no tiene un método automatizado). Por ello, la información sobre un conjunto de datos de arranque puede permanecer largo tiempo en el BSDS después de que el administrador del sistema haya suprimid el conjunto de datos de registro de archivado.

Por el contrario, el número máximo de conjuntos de datos de registro de archivado podría haberse sobrepasado y los datos BSDS se habrían descartado mucho tiempo antes de que el conjunto de datos alcanzara la fecha de caducidad.

Puede utilizar el siguiente mandato MQSC para determinar el alcance del registro y el nombre del conjunto de datos de registro de archivado o del registro activo que conservan la dirección de byte relativa antigua que resulta necesaria para diversos tipos de soporte o de recuperación del gestor de colas:

```
DISPLAY USAGE TYPE(DATASET)
```

Si el módulo de parámetros del sistema especifica que los conjuntos de datos del registro de archivado deben catalogarse al asignarse, el BSDS apunta al ICF (Integrated Catalog Facility) para obtener la información necesaria para las asignaciones posteriores. De lo contrario, las entradas del BSDS de cada volumen registran el número de serie de volumen y la información de la unidad necesarios para posteriores asignaciones.

## La versión del BSDS

El formato del BSDS varía según su versión. Aumentar la versión del BSDS permite utilizar nuevas características. IBM MQ da soporte a las siguientes versiones de BSDS:

### Versión 1

Soportado por todos los releases de IBM MQ. Un BSDS de la versión 1 admite valores de RBA de registro de 6 bytes.

### Versión 2

Soportado por IBM MQ 8.0 y posteriores. Un BSDS de la versión 2 permite valores de RBA de registro de 8 bytes y hasta 310 conjuntos de datos en cada copia del registro activo.

 **V 9.3.0** Está habilitado de forma predeterminada para los gestores de colas creados en IBM MQ 9.3.0 y posteriores.

### Versión 3

Soportado por IBM MQ 8.0 y posteriores. El BSDS se convierte automáticamente a la versión 3, desde la versión 2, cuando se añaden más de 31 conjuntos de datos a una copia de registro activa.

Puede determinar la versión de un BSDS ejecutando el programa de utilidad de correlación de registro de impresión ([CSQJU004](#)). Para convertir un BSDS de la versión 1 a la versión 2, ejecute el programa de utilidad de conversión de BSDS ([CSQJUCNV](#)).

Consulte “[RBA \(Dirección de byte relativa\) de registro mayor](#)” en la [página 247](#) para obtener más información los RBA de registro de 6 y 8 bytes.

## Conjuntos de datos de registro de archivado y copias de BSDS

Cada vez que se crea un conjunto de datos del registro de archivado nuevo, también se crea una copia en el BSDS. Si el registro de archivado se encuentra en una cinta, el conjunto de datos de arranque es el primer conjunto de datos en el primer volumen de salida. Si el registro de archivado se encuentra en DASD, el conjunto de datos de arranque se encontrará en un conjunto de datos separado.

Los nombres del conjunto de datos del registro de archivado y la copia BSDS son los mismos, excepto que el calificador de nivel inferior del nombre del registro de archivado empieza por A y la copia BSDS empieza por B, por ejemplo:

**Nombre del registro de archivado**

CSQ.ARCHLOG1.E00186.T2336229. A 0000001

**Nombre de la copia BSDS**

CSQ.ARCHLOG1.E00186.T2336229. B 0000001

Si se produce un error al copiar el conjunto de datos de arranque, la copia no se crea, se emite el mensaje [CSQJ125E](#) y la descarga en el nuevo conjunto de datos de registro de archivado sigue sin la copia del conjunto de datos de arranque.

## **Definición de sistema en z/OS**

IBM MQ for z/OS utiliza muchas definiciones de objeto predeterminadas y proporciona JCL de ejemplo para crear esos objetos predeterminados. Utilice este tema para comprender estos objetos predeterminados y el JCL ejemplo.

### **Establecer los parámetros del sistema**

En IBM MQ for z/OS, un módulo de parámetros del sistema controla los entornos de registro, archivado, rastreo y conexión que IBM MQ utiliza en sus operaciones. Los parámetros del sistema se especifican mediante tres macros de ensamblador, que son las siguientes:

**CSQ6SYSP**

Parámetros del sistema, incluyendo el establecimiento del entorno de conexión y rastreo.

**CSQ6LOGP**

Parámetros de registro cronológico.

**CSQ6ARVP**

Parámetros de archivado de registro.

Los módulos de parámetros predeterminados se suministran con IBM MQ for z/OS. Si no contienen los valores que desea utilizar, puede crear sus propios módulos de parámetros utilizando el ejemplo que se suministra con IBM MQ. El ejemplo es `th1qua1.SCSQPROC (CSQ4ZPRM)`.

Puede modificar algunos parámetros del sistema mientras un gestor de colas está en ejecución. Consulte los mandatos SET SYSTEM, SET LOG y SET ARCHIVE en [Los mandatos MQSC](#).

Para obtener más información sobre la definición, consulte los temas siguientes:

- [“Definición de objetos del sistema para IBM MQ for z/OS” en la página 250](#)
- [“Ajuste del gestor de colas en IBM MQ for z/OS” en la página 255](#)
- [“Definiciones de ejemplo suministradas con IBM MQ for z/OS” en la página 256](#)

### **Conceptos relacionados**

[Personalizar los conjuntos de datos de entrada de inicialización de ejemplo](#)

[Orígenes desde los que puede emitir mandatos MQSC y PCF en IBM MQ for z/OS](#)

### **Tareas relacionadas**

[Administración de z/OS](#)

[Configuración de clústeres](#)

[Supervisión de IBM MQ](#)

## **Definición de objetos del sistema para IBM MQ for z/OS**

IBM MQ for z/OS requiere objetos predefinidos adicionales para aplicaciones de publicación/suscripción, clúster y control de canal y otras funciones de administración del sistema.

Los objetos del sistema necesarios para IBM MQ for z/OS se pueden dividir en las categorías siguientes:

- [Objetos de publicación/suscripción](#)
- [Objetos predeterminados del sistema](#)
- [Objetos de mandato del sistema](#)
- [Objetos de administración del sistema](#)
- [Colas de canales](#)
- [Colas de clúster](#)
- [Colas del grupo de compartición de colas](#)
- [Clases de almacenamiento](#)
- [Definición de la cola de mensajes no entregados de objeto del sistema](#)
- [Cola de transmisión predeterminada](#)
- [Colas internas](#)
- [“Cola de autenticación de canal” en la página 255](#)

## Objetos de publicación/suscripción

Hay varios objetos del sistema que debe definir para poder utilizar aplicaciones de publicación/suscripción con IBM MQ for z/OS. Las definiciones de ejemplo se suministran con IBM MQ para ayudarle a definir estos objetos. Estos ejemplos están descritos en [CSQ4INSG](#).

Para utilizar la publicación/suscripción debe definir los siguientes objetos:

- Una cola local denominada SYSTEM.RETAINED.PUB.QUEUE, que se utiliza para mantener una copia de cada publicación retenida en el gestor de colas. Cada nombre de tema completo podría tener hasta una publicación retenida almacenada en esta cola. Si sus aplicaciones van a utilizar publicaciones retenidas en muchos temas diferentes o si sus mensajes de publicación retenidos son mensajes muy grandes, los requisitos de almacenamiento de esta cola deberían planificarse con cuidado, incluyendo su asignación a su propio conjunto de páginas si los requisitos de almacenamiento para el mismo son muy grandes. Para mejorar el rendimiento, debe definir esta cola con un tipo de índice de MSGID (como se muestra en la definición de cola de ejemplo proporcionada).
- Una cola local denominada SYSTEM.DURABLE.SUBSCRIBER.QUEUE, que se utiliza para mantener una copia persistente de suscripciones duraderas en el gestor de colas. Para mejorar el rendimiento, debe definir esta cola con un tipo de índice de CORRELID (como se muestra en la definición de cola de ejemplo proporcionada).
- Una cola local denominada SYSTEM.DURABLE.MODEL.QUEUE, que se utiliza como modelo para suscripciones duraderas gestionadas.
- Una cola local denominada SYSTEM.NDURABLE.MODEL.QUEUE, que se utiliza como modelo para suscripciones no duraderas gestionadas.
- Una lista de nombres denominada SYSTEM.QPUBSUB.QUEUE.NAMELIST, que contiene una lista de nombres de cola supervisados por la interfaz de publicación/suscripción en cola.
- Una lista de nombres denominada SYSTEM.QPUBSUB.SUBPOINT.NAMELIST, que contiene una lista de objetos de tema utilizados por la interfaz de publicación/suscripción en cola para coincidir con los objetos de tema para los puntos de suscripción.
- Un tema denominado SYSTEM.BASE.TOPIC, que se utiliza como un tema base para resolver atributos.
- Un tema denominado SYSTEM.BROKER.DEFAULT.STREAM, que la secuencia predeterminada utilizada por la interfaz de publicación/suscripción en cola.
- Un tema denominado SYSTEM.BROKER.DEFAULT.SUBPOINT, que es el punto de suscripción RFH2 predeterminado utilizado por la interfaz de publicación/suscripción en cola.
- Un tema denominado SYSTEM.BROKER.ADMIN.STREAM, que la secuencia admin. utilizada por la interfaz de publicación/suscripción en cola.

- Una suscripción denominada SYSTEM.DEFAULT.SUB, que es un objeto de suscripción predeterminado utilizado para proporcionar valores predeterminados en mandatos DEFINE SUB.

## Objetos predeterminados del sistema

Los objetos predeterminados del sistema se utilizan para proporcionar los atributos predeterminados al definir un objeto y no especifican el nombre de otro objeto en el que basar la definición.

Los nombres de las definiciones de objeto del sistema predeterminado empiezan por los caracteres "SYSTEM.DEFAULT" o "SYSTEM.DEF." Por ejemplo, la cola local predeterminada del sistema se denomina SYSTEM.DEFAULT.LOCAL.QUEUE.

Estos objetos definen los valores predeterminados del sistema para los atributos de estos objetos de IBM MQ:

- Colas locales
- Colas modelo
- Colas alias
- Colas remotas
- todos los Procesos
- Listas de nombres
- Canales
- Clases de almacenamiento
- Información de autenticación

Las colas compartidas son un tipo especial de cola local, por lo que al definir una cola compartida, la definición se basa en SYSTEM.DEFAULT.LOCAL.QUEUE. Debe recordar indicar un valor para el nombre de la estructura de recurso de acoplamiento porque no se especifica ninguno en la definición predeterminada. Alternativamente, puede definir su propia definición de cola compartida predeterminada para utilizarla como base para las colas compartidas de modo que todas hereden los atributos necesarios. Recuerde que sólo necesita definir una cola compartida en un gestor de colas del grupo de compartición de colas.

## Objetos de mandato del sistema

Los nombres de los objetos de mandatos de sistema empiezan por los caracteres SYSTEM.COMMAND. Debe definir estos objetos para poder utilizar las operaciones y los paneles de control de IBM MQ con el fin de emitir mandatos a un subsistema IBM MQ.

Hay dos objetos de mandato de sistema:

1. La cola de entrada del mandato del sistema es una cola local en la que se colocan los mandatos antes de que los procese el procesador de mandatos de IBM MQ. Su nombre debe ser SYSTEM.COMMAND.INPUT. También debe definirse un alias denominado SYSTEM.ADMIN.COMMAND.QUEUE, para compatibilidad con IBM MQ for Multiplatforms y para que lo utilicen IBM MQ Console y administrative REST API.
2. SYSTEM.COMMAND.REPLY.MODEL es una cola modelo que define la cola de respuesta de mandatos del sistema.

Hay dos objetos adicionales para el uso de IBM MQ Explorer:

- cola SYSTEM.MQEXPLORER.REPLY.MODEL
- canal SYSTEM.ADMIN.SVRCONN

SYSTEM.REST.REPLY.QUEUE es la cola de respuestas que la IBM MQ administrative REST API utiliza.

Los mandatos normalmente se envían utilizando mensajes no permanentes, por lo que los objetos de mandato del sistema deben tener el atributo DEFPSIST(NO) de forma que las aplicaciones que los utilicen

(incluyendo las aplicaciones suministradas como el programa de utilidad y las operaciones y los paneles de control) obtengan mensajes no permanentes de forma predeterminada. Si tiene una aplicación que utilice mensajes permanentes para mandatos, establezca el atributo DEFTYPE(PERMDYN) para la cola de respuesta, porque los mensajes de respuesta a tales mandatos son permanentes.

## Objetos de administración del sistema

Los nombres de los objetos de administración del sistema empiezan por los caracteres SYSTEM.ADMIN.

Hay siete objetos de administración del sistema:

- La cola SYSTEM.ADMIN.CHANNEL.EVENT
- La cola SYSTEM.ADMIN.COMMAND.EVENT
- La cola SYSTEM.ADMIN.CONFIG.EVENT
- La cola SYSTEM.ADMIN.PERFM.EVENT
- La cola SYSTEM.ADMIN.QMGR.EVENT
- La cola SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- La cola SYSTEM.ADMIN.ACTIVITY.QUEUE

## Colas de canales

Para utilizar colas distribuidas, debe definir los objetos siguientes:

- Una cola local con el nombre SYSTEM.CHANNEL.SYNCQ, que se utiliza para mantener números de secuencia y unidades lógicas de identificadores de trabajo (LUWID) de los canales. Para mejorar el rendimiento del canal, debe definir esta cola con un tipo de índice de MSGID (como se muestra en la definición de cola de ejemplo proporcionada).
- Una cola local con el nombre SYSTEM.CHANNEL.INITQ, que se utiliza para mandatos de canal.

No puede definir estas colas como colas compartidas.

## Colas de clúster

Para utilizar clústeres de IBM MQ, debe definir los objetos siguientes:

- Una cola local denominada SYSTEM.CLUSTER.COMMAND.QUEUE, que se utiliza para comunicar cambios del repositorio entre gestores de colas. Los mensajes escritos en esta cola contienen actualizaciones de los datos del repositorio que deben aplicarse a la copia local del repositorio, o solicitudes para los datos de repositorio.
- Una cola local denominada SYSTEM.CLUSTER.REPOSITORY.QUEUE, que se utiliza para retener una copia persistente del repositorio.
- Una cola local denominada SYSTEM.CLUSTER.TRANSMIT.QUEUE, que es la cola de transmisión de todas las destinaciones del clúster. Por razones de rendimiento, debe definir esta cola con un tipo de índice de CORRELID (como se muestra en la definición de cola de ejemplo).

Estas colas normalmente contienen un número elevado de mensajes.

No puede definir estas colas como colas compartidas.

## Colas del grupo de compartición de colas

Para utilizar canales compartidos y transferencia a colas dentro del grupo, debe definir los siguientes objetos:

- Una cola compartida con el nombre SYSTEM.QSG.CHANNEL.SYNCQ, que se utilice para mantener información de sincronización para canales compartidos.

- Una cola compartida con el nombre SYSTEM.QSG.TRANSMIT.QUEUE, que se utilice como la cola de transmisión para la transferencia a colas dentro del grupo. Si realiza la ejecución en un grupo de compartición de colas, debe definir esta cola, incluso aunque no utilice la transferencia a colas dentro del grupo.

## Clases de almacenamiento

Es recomendable definir las seis siguientes clases de almacenamiento. Debe definir cuatro de ellos porque son necesarios para IBM MQ. Las otras definiciones de clase de almacenamiento son recomendables porque se utilizan en las definiciones de cola de ejemplo.

### **DEFAULT (necesaria)**

Esta clase de almacenamiento se utiliza para todas las colas de mensajes que no tienen un rendimiento crítico y que no se ajustan a ninguna de las otras clases de almacenamiento. También es la clase de almacenamiento predeterminada suministrada si no especifica una al definir la cola.

### **NODEFINE (necesaria)**

Esta clase de almacenamiento se utiliza si la clase de almacenamiento especificada al definir una cola no está definida.

### **REMOTE (necesaria)**

Esta clase de almacenamiento se utiliza principalmente para colas de transmisión, es decir, colas relacionadas con el sistema con mensajes de rendimiento crítico de vida corta.

### **SYSNLGLV**

Esta clase de almacenamiento se utiliza para mensajes de rendimiento crítico y de larga duración.

### **SYSTEM (necesaria)**

Esta clase de almacenamiento se utiliza para colas de mensajes relacionadas con el sistema y críticas de rendimiento, por ejemplo, SYSTEM.CHANNEL.SYNQ y SYSTEM.CLUSTER.\* colas.

### **SYSVOLAT**

Esta clase de almacenamiento se utiliza para mensajes de rendimiento crítico y de vida corta.

Puede modificar sus atributos y añadir otras definiciones de clase de almacenamiento, según sea necesario.

## Definición de la cola de mensajes no entregados de objeto del sistema

La cola de mensajes no entregados se utiliza si el destino del mensaje no es válido. IBM MQ coloca estos mensajes en una cola local denominada "cola de mensajes no entregados". Aunque no es obligatorio tener una cola de mensajes no entregados, debe considerarla como esencial, especialmente si utiliza colas distribuidas o uno de los puentes de IBM MQ.

**No** defina la cola de mensajes no entregados como cola compartida. Una transferencia a una cola local en un gestor de colas puede colocarse en la cola de mensajes no entregados. Si la cola de mensajes no entregados era una cola compartida, un manejador de colas de mensajes no entregados podría procesar el mensaje y colocarlo en una cola con el mismo nombre, pero como se encuentra en un gestor de colas diferente, sería la cola errónea o tendría un perfil de seguridad distinto. Si la cola no existía, no conseguiría reprocesarla.

Si decide definir una cola de mensajes no entregados, también debe indicar su nombre al gestor de colas. Para ello, utilice el mandato ALTER QMGR DEADQ(*nombre-cola*). Para obtener más información, consulte [Visualización y modificación de los atributos del gestor de colas](#).

## Cola de transmisión predeterminada

La cola de transmisión predeterminada se utiliza cuando no hay ninguna otra cola de transmisión disponible para enviar mensajes a otro gestor de colas. Si define una cola de transmisión predeterminada,

también debe definir un canal para que sirva a la cola. Si no lo hace, los mensajes que se colocan en la cola de transmisión predeterminada no se transmiten al gestor de colas remoto y permanecen en la cola.

Si decide definir una cola de transmisión predeterminada, también debe indicar su nombre al gestor de colas. Para hacerlo, utilice el mandato ALTER QMGR.

## Colas internas

### • Cola de datos pendientes

- Una cola definida para uso interno, SYSTEM.PENDING.DATA.QUEUE, da soporte al uso de suscripciones duraderas en un entorno de publicación/suscripción de JMS .

### • Cola de transferencia de retardo de entrega de JMS 2.0

- Si se utiliza la funcionalidad de retardo de entrega proporcionada por JMS 2.0 , se utiliza una cola de transferencia interna, SYSTEM.DDELAY.LOCAL.QUEUE, debe estar definido. El gestor de colas utiliza dicha cola para almacenar temporalmente los mensajes enviados con un retardo de entrega distinto de cero hasta que finaliza el retardo de entrega y el mensaje se coloca en su destino final. En CSQ4INSG se proporciona una definición de ejemplo comentada.
- Cuando se define la cola SYSTEM.DDELAY.LOCAL.QUEUE, deberán definirse los atributos STGCLASS, MAXMSGL y MAXDEPTH para el número anticipado de mensajes que se vayan a enviar con retardo de entrega. Además, al definir la cola SYSTEM.DDELAY.LOCAL.QUEUE, hay que asegurarse de que solo el gestor de colas pueda poner mensajes en ella. Hay que procurar que ningún identificador de usuario tenga autoridad para poner mensajes en esta cola.

## Cola de autenticación de canal

La cola SYSTEM.CHLAUTH.DATA.QUEUE es obligatoria para el uso interno de la autenticación de canal. Las definiciones de ejemplo se suministran con IBM MQ para ayudarle a definir estos objetos. Este ejemplo se describe en CSQ4INSA, que también define algunas reglas predeterminadas.

## Ajuste del gestor de colas en IBM MQ for z/OS

Hay unos pocos pasos sencillos que puede adoptar para asegurarse de que el gestor de colas se ajuste a fin de evitar problemas de rendimiento básicos.

Hay varias formas de mejorar el rendimiento del gestor de colas. Estas formas están controladas por el conjunto de atributos del gestor de colas mediante el mandato ALTER QMGR. Esta sección contiene información sobre cómo puede hacerlo estableciendo el número máximo de mensajes permitido en el gestor de colas o realizando "mantenimiento" en el gestor de colas. [IBM MQ SupportPac MP16 -IBMMQ paraz/OS Planificación y ajuste de capacidad](#) proporciona más información sobre el rendimiento y el ajuste.

## Puntos de sincronismo

Uno de los roles del gestor de colas es el control del punto de sincronización dentro de una aplicación. Una aplicación construye una unidad de trabajo que contiene cualquier número de llamadas MQPUT o MQGET terminadas con una llamada MQCMIT.

A medida que aumenta el número de llamadas MQPUT o MQGET dentro del ámbito de un MQCMIT, el coste de rendimiento de la confirmación aumenta significativamente. Las aplicaciones, en general, deben estar diseñadas para no ejecutar MQPUT/MQGET en un gran número de mensajes en un único punto de sincronismo.

Puede limitar administrativamente el número de mensajes dentro de un único punto de sincronismo utilizando el atributo del gestor de colas MAXUMSGS. Si una aplicación excede este límite, recibe

MQRC\_SYNCPOINT\_LIMIT\_ALCANZADO en la llamada MQPUT, MQPUT1o MQGET que excede el límite. A continuación, la aplicación debe emitir MQCMIT o MQBACK según corresponda.

El valor predeterminado de MAXUMSGS es 10000. Este valor se puede reducir si desea aplicar un límite inferior, que también puede ayudar a proteger las aplicaciones en bucle. Antes de reducir MAXUMSGS, cerciórese de que comprende las aplicaciones existentes para asegurarse de que no superan el límite o pueden tolerar el código de retorno MQRC\_SYNCPOINT\_LIMIT\_REACHED

## Mensajes caducados

Los mensajes que han caducado son descartados por la siguiente llamada MQGET adecuada. Sin embargo, si se produce tal llamada, los mensajes caducados no se descartan y, para algunas colas, especialmente aquellas en las que la recuperación de mensajes la realiza mediante MessageId, CorrelId o GroupId y la cola está indexada para el rendimiento, pueden acumularse muchos mensajes caducados. El gestor de colas puede explorar periódicamente cualquier cola de mensajes caducados, que se suprimen a continuación. Puede elegir la frecuencia con la que debe llevarse a cabo esta exploración, si es que debe realizarse. Hay dos formas de llevarla a cabo:

### Solicitud explícita

Puede controlar qué colas deben explorarse y cuándo. Emita el mandato REFRESH QMGR TYPE(EXPIRY), especificando la cola o colas que desea que se exploren.

### Exploración periódica

Puede especificar un intervalo de caducidad en el objeto de gestor de colas utilizando el atributo EXPRYINT. El gestor de colas mantiene información sobre los mensajes caducados en cada cola, y sabe a qué hora vale la pena realizar una exploración de los mensajes caducados. Cada vez que se alcanza el intervalo EXPRYINT, el gestor de colas busca colas de candidatos en las que valga la pena explorar mensajes caducados, y sólo explora aquellas colas dignas de consideración. No explora todas las colas. Esto evita que el procesador pierda tiempo en exploraciones innecesarias.

Las colas compartidas sólo son exploradas por un gestor de colas del grupo de compartición de colas. Normalmente, el primer gestor de colas que realiza el reinicio o el primero que tiene el conjunto EXPRYINT realiza la exploración.

**Nota:** Debe establecer el mismo valor EXPRYINT para todos los gestores de colas dentro de un grupo de compartición de colas.

## Definiciones de ejemplo suministradas con IBM MQ for z/OS

Utilice este tema como referencia para el JCL de ejemplo y el código suministrado con IBM MQ for z/OS.

Las siguientes definiciones de ejemplo se suministran con IBM MQ en la biblioteca thlqual.SCSQPROC. Puede utilizarlas para definir los objetos del sistema y personalizar sus propios objetos. Puede incluir algunos de ellos en los conjuntos de datos de entrada de inicialización (descritos en [Comandos de inicialización](#)).

<i>Tabla 22. Definiciones de ejemplo de IBM MQ para objetos del sistema</i>	
<b>Conjunto de datos de entrada de inicialización</b>	<b>Nombre de ejemplo</b>
<u>CSQINP1</u>	CSQ4INP1 CSQ4INPR

Tabla 22. Definiciones de ejemplo de IBM MQ para objetos del sistema (continuación)

Conjunto de datos de entrada de inicialización	Nombre de ejemplo
<u>CSQINP2</u>	CSQ4INSA CSQ4INYS <sup>1</sup> CSQ4INSX CSQ4INSG CSQ4INSR CSQ4INSS CSQ4INSJ CSQ4INSM CSQ4INYG CSQ4INYR CSQ4INYC CSQ4INYD CSQ4INSC
<u>CSQINPT</u>	CSQ4INST CSQ4INYT
<u>Otros</u>	CSQ4DISP CSQ4INPX CSQ4IVPQ CSQ4IVPG CSQ4MSTR CSQ4MSRR CSQ4QMIN

**Nota:**

1. El orden de estas definiciones de ejemplo es importante: se produce un error si INYS, INSX, e INSG están ordenados incorrectamente.

**Ejemplos de CSQINP1**

Utilice el conjunto de datos CSQINP1 de ejemplo `thlqual.SCSQPROC(CSQ4INP1)` cuando utilice un conjunto de páginas para cada clase de mensaje, o `thlqual.SCSQPROC(CSQ4INPR)` cuando utilice múltiples conjuntos de páginas para clases más importantes del mensaje. Contiene definiciones de agrupaciones de almacenamiento intermedio, un conjunto de páginas para asociaciones de agrupaciones de almacenamiento intermedio y un mandato ALTER SECURITY. Incluya el ejemplo en la concatenación CSQINP1 del procedimiento de tarea iniciada por el gestor de colas.

**Ejemplos de CSQINP2**

**Ejemplo de objeto del sistema CSQ4INSG**

El conjunto de datos CSQINP2 de ejemplo `thlqual.SCSQPROC(CSQ4INSG)` contiene definiciones de los siguientes objetos de sistema de uso general:

- Objetos predeterminados del sistema
- Objetos de mandato del sistema
- Objetos de administración del sistema

- Otros objetos para uso del sistema

Debe definir los objetos en este ejemplo, pero sólo debe hacerlo una vez cuando el subsistema se ha iniciado por primera vez. La inclusión de definiciones en el conjunto de datos CSQINP2 constituye la mejor forma de hacerlo. Se mantienen aunque se produzca un reinicio y una conclusión del gestor de colas. No debe cambiar los nombres de objeto, pero puede cambiar sus atributos si es necesario.

Cuando se cumplen las condiciones siguientes, se coloca un mensaje en la cola SYSTEM.DURABLE.SUBSCRIBER.QUEUE (incluso si la publicación/suscripción no está activa):

- El atributo de gestor de colas PSMODE está establecido en DISABLED
- La sentencia CSQ4INST del objeto de ejemplo DEFINE SUB('SYSTEM.DEFAULT.SUB') está presente.

Para evitarlo, suprima o comente la sentencia DEFINE SUB('SYSTEM.DEFAULT.SUB').

La cola de transferencia de retardo de entrega JMS 2.0, SYSTEM.DDELAY.LOCAL.QUEUE sólo debe definirse si se utiliza el retardo de entrega de JMS 2.0. De forma predeterminada, la definición de cola aparece comentada, pero puede descomentarse si es necesario.

### **Ejemplo de autenticación y objeto del sistema CSQ4INSA**

El conjunto de datos CSQINP2 de ejemplo thlqual.SCSQPROC(CSQ4INSA) contiene la definición de la cola de sistema de autenticación de canal. Esta cola contiene los registros de autenticación de canal. También contiene las reglas de autenticación de canal predeterminadas.

Debe definir los objetos de este ejemplo si CHLAUTH es ENABLED en el gestor de colas y desea ejecutar canales, o desea establecer o visualizar el registro (SET o DISPLAY CHLAUTH). Sólo deberá definirlos una vez cuando se inicie por primera vez el subsistema. La inclusión de definiciones en el conjunto de datos CSQINP2 constituye la mejor forma de hacerlo. Se mantienen a través de un cierre y reinicio del gestor de colas. No debe cambiar el nombre de la cola.

### **Ejemplo de objeto del sistema CSQ4INSS**

Puede definir objetos del sistema adicionales si utiliza grupos de compartición de colas.

El conjunto de datos de ejemplo thlqual.SCSQPROC(CSQ4INSS) contiene mandatos que deben utilizarse para estructuras CF y un conjunto de definiciones para los objetos del sistema necesarios para canales compartidos y transferencia a colas dentro del grupo.

No puede utilizar este ejemplo tal cual; debe personalizarlo antes de utilizarlo. A continuación, puede incluir este miembro en la concatenación CSQINP2 DD del procedimiento de arranque del gestor de colas, o puede utilizarlo como entrada de la función COMMAND del programa de utilidad CSQUTIL para emitir los mandatos necesarios.

Cuando se definen objetos compartidos o de grupo, deberá incluirlos en la concatenación CSQINP2 DD para un único gestor de colas del grupo de compartición de colas.

### **Ejemplo de objeto del sistema de CSQ4INSX**

Debe definir objetos de sistema adicionales si está utilizando colas distribuidas y agrupaciones en clúster.

El conjunto de datos de ejemplo thlqual.SCSQPROC(CSQ4INSX) contiene las definiciones de cola necesarias. Puede incluir este miembro en la concatenación CSQINP2 DD del procedimiento de arranque del gestor de colas, o puede utilizarlo como entrada en la función de MANDATO en el programa de utilidad CSQUTIL para emitir los mandatos DEFINE necesario.

Hay dos tipos de definiciones de objeto:

- SYSTEM.CHANNEL.xx, necesaria para cualquier cola distribuida
- SYSTEM.CLUSTER.xx, necesaria para la agrupación en clúster

## **Ejemplo de objeto CSQ4INSJ del sistema JMS**

Define colas utilizadas en el dominio de publicación/suscripción de JMS.

## **Ejemplo de objeto del sistema CSQ4INSM**

Si se está utilizando la seguridad de mensajes avanzada, deberán definirse objetos de sistema adicionales. El conjunto de datos de ejemplo thlqual.SCSQPROC(CSQ4INSM) contiene las definiciones de cola necesarias.

## **Ejemplo de objeto CSQ4INSR**

Define colas utilizadas por WebSphere Application Server e intermediarios.

## **Ejemplo de objeto CSQ4INYD**

Si utiliza colas distribuidas y debe establecer sus propias colas, procesos y canales.

El conjunto de datos de ejemplo thlqual.SCSQPROC(CSQ4INYD) contiene definiciones de muestra que se pueden utilizar para personalizar los objetos de gestión de colas distribuidas. Incluye:

- Un conjunto de definiciones para el extremo emisor
- Un conjunto de definiciones para el extremo receptor
- Un conjunto de definiciones para utilizar clientes

No puede utilizar este ejemplo tal cual; debe personalizarlo antes de utilizarlo. A continuación, puede incluir este miembro en la concatenación CSQINP2 DD del procedimiento de arranque del gestor de colas, o puede utilizarlo como entrada de la función COMMAND del programa de utilidad CSQUTIL para emitir los mandatos DEFINE necesarios. (Esto es preferible porque significa que no tendrá que redefinir estos objetos cada vez que reinicie el gestor de colas).

## **Ejemplo de objeto CSQ4INYC**

Si utiliza la agrupación en clúster, las definiciones equivalentes a las definiciones de canal y las definiciones de colas remotas de la gestión de colas distribuidas se crean automáticamente, cuando resulta necesario. Sin embargo, algunas definiciones de canal manuales resultan necesarias (un canal de clúster receptor para el clúster y una definición de remitente de clúster para como mínimo un gestor de colas de repositorio de clúster).

El conjunto de datos de muestra: thlqual.SCSQPROC(CSQ4INYC) contiene las siguientes definiciones de ejemplo que se pueden utilizar para personalizar los objetos de agrupación en clúster:

- Definiciones para el gestor de colas
- Definiciones para el canal receptor
- Definiciones para el canal emisor
- Definiciones para colas de clúster
- Definiciones para listas de clústeres

No puede utilizar este ejemplo tal cual; debe personalizarlo antes de utilizarlo. A continuación, puede incluir este miembro en la concatenación CSQINP2 DD del procedimiento de arranque del gestor de colas, o puede utilizarlo como entrada de la función COMMAND del programa de utilidad CSQUTIL para emitir los mandatos DEFINE necesarios. Es preferible porque significa que no tiene que volver a definir estos objetos cada vez que reinicie IBM MQ.

## **Ejemplo de objeto CSQ4INYG**

El conjunto de datos de ejemplo: thlqual.SCSQPROC(CSQ4INYG) contiene las siguientes definiciones de ejemplo que se pueden utilizar para personalizar objetos propios para uso general:

- Cola de mensajes no entregados
- Cola de transmisión predeterminada
- Objetos de adaptador de CICS

No puede utilizar este ejemplo tal cual; debe personalizarlo antes de utilizarlo. A continuación, puede incluir este miembro en la concatenación CSQINP2 DD del procedimiento de arranque del gestor de colas, o puede utilizarlo como entrada de la función COMMAND del programa de utilidad CSQUTIL para emitir los mandatos DEFINE necesarios. Es preferible porque significa que no tiene que volver a definir estos objetos cada vez que reinicie IBM MQ.

Además de las definiciones de ejemplo aquí, puede utilizar las definiciones de objeto del sistema como base para sus propias definiciones de recursos. Por ejemplo, puede hacer una copia de trabajo de SYSTEM.DEFAULT.LOCAL.QUEUE y denominarla MY.DEFAULT.LOCAL.QUEUE. A continuación, puede cambiar todos los parámetros que sea necesario en esta copia. Podría emitir el mandato DEFINE con cualquier método que elija, siempre y cuando tenga la suficiente autoridad para crear recursos de dicho tipo.

### **Cola de transmisión predeterminada**

Lea la descripción de la [Cola de transmisión predeterminada](#) antes de decidir si desea definir una cola de transmisión predeterminada.

- Si decide que desea definir una cola de transmisión predeterminada, recuerde que también debe definir un canal para que lo sirva.
- Si decide que no desea definir una, recuerde eliminar la sentencia DEFEXMITQ del mandato ALTER QMGR en el ejemplo.

### **Objetos de adaptador de CICS**

El ejemplo define una cola de inicio denominada CICS01.INITQ. Esta cola la utiliza la transacción CKTI suministrada por IBM MQ. Puede cambiar el nombre de esta cola; sin embargo, debe coincidir con el nombre especificado en la tabla de inicialización del sistema (SIT) de CICS o bien SYSIN en la sentencia INITPARM.

## **Ejemplos de objeto CSQ4INYS/CSQ4INYR**

Definiciones de clase de almacenamiento para utilizar:

- un conjunto de páginas para cada clase de mensaje
- varios conjuntos de páginas para clases de mensaje principales

Por ejemplo, SYSTEM.COMMAND.INPUT utiliza STGCLASS ('SYSVOLAT') y SYSTEM.CLUSTER.TRANSMIT.QUEUE utiliza STGCLASS ('REMOTE'). En CSQ4INYS, ambas clases de almacenamiento utilizan el mismo conjunto de páginas. En CSQ4INYR, estas clases de almacenamiento utilizan conjuntos de páginas diferentes para reducir el impacto del llenado de la cola de transmisión.

## **Ejemplos de CSQINPT**

### **CSQ4INST**

El conjunto de datos de ejemplo: thlqual.SCSQPROC(CSQ4INST) contiene la definición de la suscripción predeterminada del sistema.

Debe definir el objeto de este ejemplo, pero sólo es necesario una vez cuando se inicie por primera vez el motor de publicación/suscripción. Para ello, lo mejor es incluir la definición en el conjunto de datos CSQINPT. Los datos mantienen aunque se concluya y se reinicie el gestor de colas. No debe cambiar el nombre de objeto, pero puede cambiar los atributos si es necesario.

### **CSQ4INYT**

El conjunto de datos de ejemplo: thlqual.SCSQPROC(CSQ4INYT) contiene un conjunto de mandatos que tal vez desee ejecutar cuando se inicie el motor de publicación/suscripción. Este ejemplo muestra información sobre el Tema y la Suscripción.

## **Otro**

### **Ejemplo de visualización de CSQ4DISP**

El conjunto de datos de ejemplo: thlqual.SCSQPROC(CSQ4DISP) contiene un conjunto de mandatos DISPLAY genéricos que muestran todos los recursos definidos en el gestor de colas. Esto incluye las definiciones de todos los objetos y definiciones de IBM MQ como, por ejemplo, clases de almacenamiento y rastreo. Estos mandatos pueden generar una gran cantidad de salida. Puede utilizar este ejemplo en el conjunto de datos CSQINP2 o como entrada de la función COMMAND del programa de utilidad CSQUTIL.

### **Ejemplo de CSQ4INPX**

El conjunto de datos de ejemplo: thlqual.SCSQPROC(CSQ4INPX) contiene un conjunto de mandatos que el usuario puede ejecutar cada vez que se inicie el iniciador de canal. Debe personalizar este ejemplo antes de utilizarlo y luego puede incluirlo en el conjunto de datos CSQINPX para el iniciador de canal.

### **Ejemplos de CSQ4IVPQ y CSQ4IVPG**

Los conjuntos de datos de ejemplo: thlqual.SCSQPROC(CSQ4IVPQ) y thlqual.SCSQPROC(CSQ4IVPG) contienen conjuntos de mandatos DEFINE que son necesarios para ejecutar los programas de verificación de instalación (IVP).

Puede incluir estos ejemplos en el conjunto de datos CSQINP2. Cuando haya ejecutado los IVP correctamente, no será necesario ejecutarlos de nuevo cada vez que se reinicie el gestor de colas. Por lo tanto, no es necesario mantener estos ejemplos en la concatenación CSQINP2 permanentemente.

### **Ejemplos de CSQ4MSTR y CSQ4MSRR**

Son procedimientos de la tarea iniciada de ejemplo para el gestor de colas: thlqual.SCSQPROC (CSQ4MSTR) y thlqual.SCSQPROC (CSQ4MSRR).

CSQ4MSRR utiliza CSQ4INYP en la concatenación CSQINP2 para que las colas importantes se distribuyan entre distintos conjuntos de páginas.

Puede eliminar los comentarios para poder utilizar la tarjeta CSQMINI para gestores de colas recién creados si es necesario.

### **Ejemplo de CSQ4QMIN**

Un conjunto de datos QMINI de ejemplo, thlqual.SCSQPROC (CSQ4QMIN).

Consulte [Conjunto de datos QMINI](#) para obtener más detalles del conjunto de datos QMINI y la stanza **TransportSecurity**.

## **z/OS Recuperación y reinicio en z/OS**

Utilice los enlaces de este tema para obtener información sobre las características de IBM MQ for z/OS para el reinicio y la recuperación.

IBM MQ for z/OS ofrece potentes características para el reinicio y la recuperación. Para obtener información sobre cómo se recupera un gestor de colas después de haberse detenido, y lo que sucede cuando se reinicia, consulte los subtemas siguientes:

- [“Cómo se realizan los cambios en los datos en IBM MQ for z/OS” en la página 262](#)
- [“Cómo se mantiene la coherencia en IBM MQ for z/OS” en la página 263](#)
- [“Qué sucede durante la terminación en IBM MQ for z/OS” en la página 266](#)
- [“Qué sucede durante el reinicio y la recuperación en IBM MQ for z/OS” en la página 267](#)
- [“Cómo se resuelven las unidades pendientes de recuperación” en la página 269](#)
- [“Recuperación de la cola compartida” en la página 272](#)

### **Conceptos relacionados**

**z/OS** [Acciones de recuperación de IBM MQ for z/OS](#)

## Tareas relacionadas

Planificación de la copia de seguridad y la recuperación

z/OS Administración de z/OS

## Referencia relacionada

z/OS Mensajes para IBM MQ for z/OS

## z/OS Cómo se realizan los cambios en los datos en IBM MQ for z/OS

IBM MQ for z/OS debe interactuar con otros subsistemas para mantener todos los datos coherentes. Este tema contiene información sobre *unidades de recuperación*, lo que son y cómo se utilizan en *restituciones*.

## Unidades de recuperación

Una *unidad de recuperación* es el proceso realizado por un único gestor de colas para un programa de aplicación, que cambia los datos de IBM MQ entre un punto de coherencia y otro. Un *punto de coherencia* (también llamado *punto de sincronismo* o *punto de confirmación*) es un punto en el tiempo, en el que todos los datos recuperables a los que un programa de aplicación tiene acceso, son coherentes.

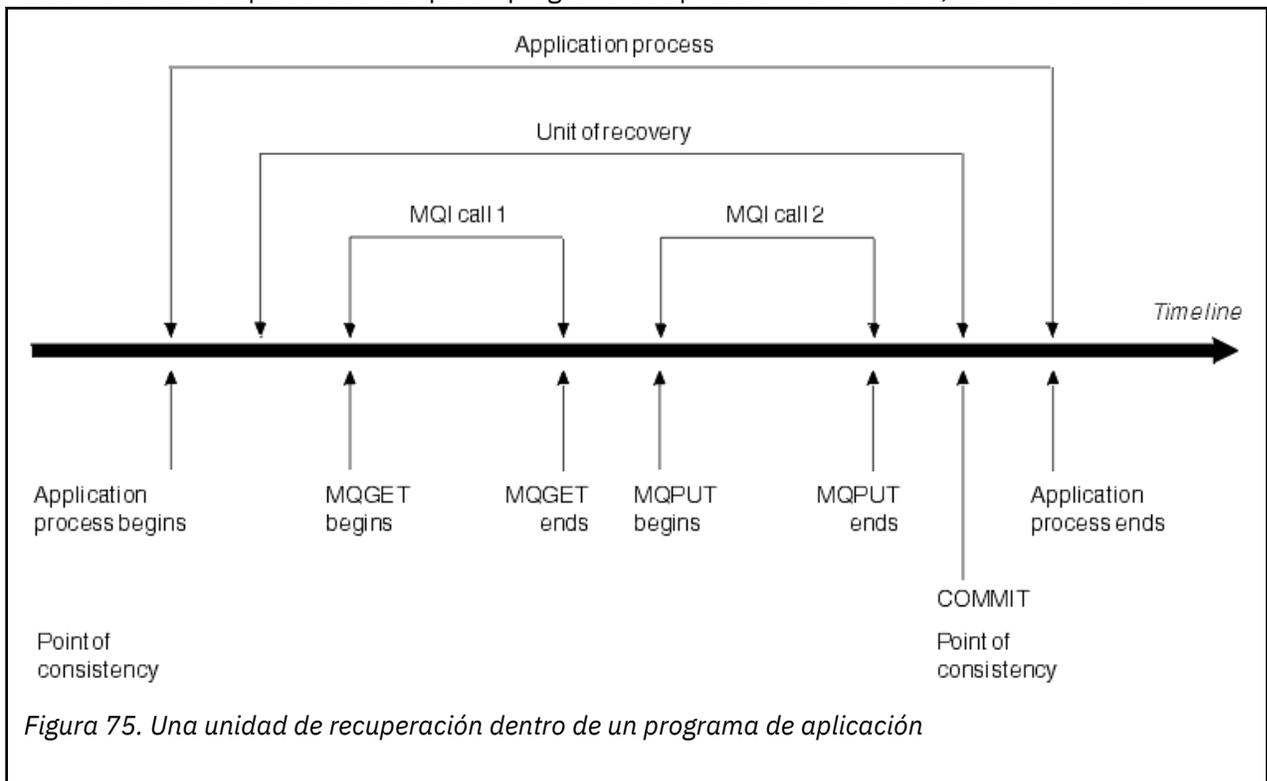


Figura 75. Una unidad de recuperación dentro de un programa de aplicación

Una unidad de recuperación empieza por el primer cambio en los datos después de principio del programa o siguiendo el punto de coherencia anterior; finaliza en un punto de coherencia posterior. [Figura 75 en la página 262](#) muestra la relación entre unidades de recuperación, el punto de coherencia y un programa de aplicación. En este ejemplo, el programa de aplicación realiza cambios en las colas a través de las llamadas MQI 1 y 2. El programa de aplicación puede incluir más de una unidad de recuperación o tan solo una. Sin embargo, cualquier unidad completa de recuperación finaliza en un punto de confirmación.

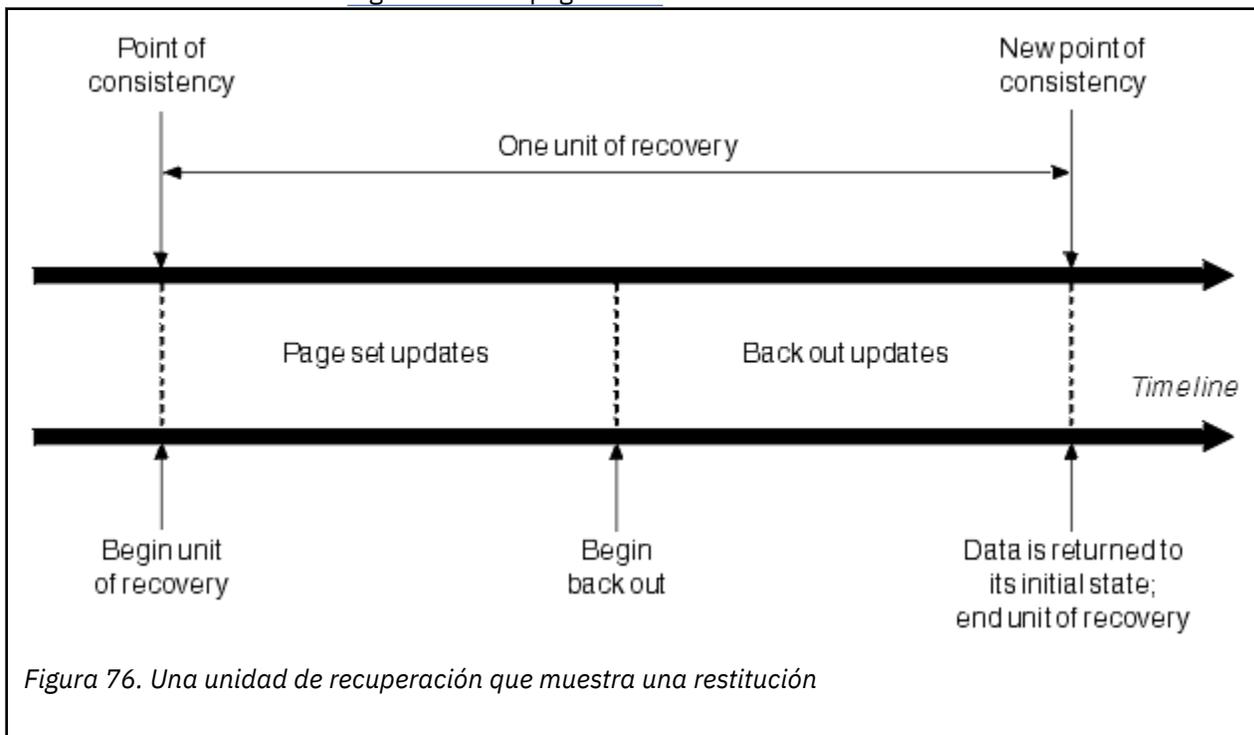
Por ejemplo, una transacción bancaria transfiere fondos de una cuenta a otra. En primer lugar, el programa resta el importe de la primera cuenta, la cuenta A. A continuación, añade el importe a la segunda cuenta, B. Después de restar el importe de A, las dos cuentas no son coherentes y IBM MQ no puede confirmar. Son coherentes cuando se añade el importe a la cuenta B. Cuando ambos pasos se han

completado, el programa puede anunciar un punto de coherencia a través de una confirmación, haciendo que los cambios sean visibles para otros programas de aplicación.

La terminación normal de un programa de aplicación desencadena automáticamente un punto de coherencia. Algunas solicitudes de programa en CICS y programas IMS también causarán un punto de coherencia, por ejemplo, EXEC CICS SYNCPOINT.

## Restitución de trabajo

Si se produce un error dentro de una unidad de recuperación, IBM MQ elimina los cambios en los datos, devolviéndolos su estado al principio de la unidad de recuperación; es decir, IBM MQ restituye el trabajo. Los sucesos se muestran en Figura 76 en la página 263.



## z/OS Cómo se mantiene la coherencia en IBM MQ for z/OS

Los datos de IBM MQ for z/OS deben ser coherentes con el proceso por lotes, CICS, IMS o TSO. Todos los datos que se modifiquen en un sistema deben ser correlacionados con un cambio en el otro.

Antes de que un sistema confirme los datos cambiados, debe saber si el otro sistema puede realizar el cambio correspondiente. Por ello, los sistemas deben comunicarse.

Durante una *confirmación en dos fases* (por ejemplo, bajo CICS), un subsistema coordina el proceso. Dicho subsistema se denomina *coordinador*; el otro es el *participante*. CICS o IMS siempre es el coordinador en las interacciones con IBM MQ, e IBM MQ siempre es el participante. En el entorno de proceso por lotes o TSO, IBM MQ puede participar en protocolos de confirmación de dos fases coordinados por z/OS RRS.

Durante una *confirmación en una sola fase* (por ejemplo, bajo TSO o por lotes), IBM MQ siempre es el coordinador en las interacciones y controla completamente el proceso de confirmación.

En un entorno WebSphere Application Server, la semántica del objeto de sesión JMS determina si se utiliza la coordinación de la confirmación en una sola fase o en dos fases.

## Coherencia con CICS o con IMS

La conexión entre IBM MQ y CICS o IMS es compatible con los siguientes protocolos de punto de sincronización:

- Confirmación en dos fases: para transacciones que actualizan los recursos que son propiedad de más de un gestor de recursos.

Se trata del protocolo de punto de sincronización distribuido estándar. Implica más anotaciones cronológicas y flujos de mensajes que una confirmación en una sola fase.

- Confirmación en una sola fase: para las transacciones que actualizan los recursos que son propiedad de un único gestor de recursos (IBM MQ).

Este protocolo está optimizado para las anotaciones cronológicas y flujos de mensajes.

- Omisión del punto de sincronización: para transacciones en las que participa IBM MQ, pero que no hacen nada en el gestor de colas que requiere un punto de sincronización (por ejemplo, examinar una cola).

En cada caso, CICS o IMS actúa como el gestor de puntos de sincronización.

Las etapas de la confirmación en dos fases que IBM MQ utiliza para comunicarse con CICS o IMS son las siguientes:

1. En la fase 1, cada sistema determina de forma independiente si ha registrado suficiente información de recuperación en su registro y si puede confirmar su trabajo.  
Al final de la fase, los sistemas se comunican. Si están de acuerdo, cada uno empieza la siguiente fase.
2. En la fase 2, los cambios son permanentes. Si uno de los sistemas termina de forma anómala durante la fase 2, la operación la completa el proceso de recuperación durante el reinicio.

### Ilustración del proceso de confirmación en dos fases

Figura 77 en la página 264 ilustra el proceso de confirmación en dos fases. Los sucesos del coordinador CICS o IMS se muestran en la línea superior, los sucesos de IBM MQ, en la línea inferior.

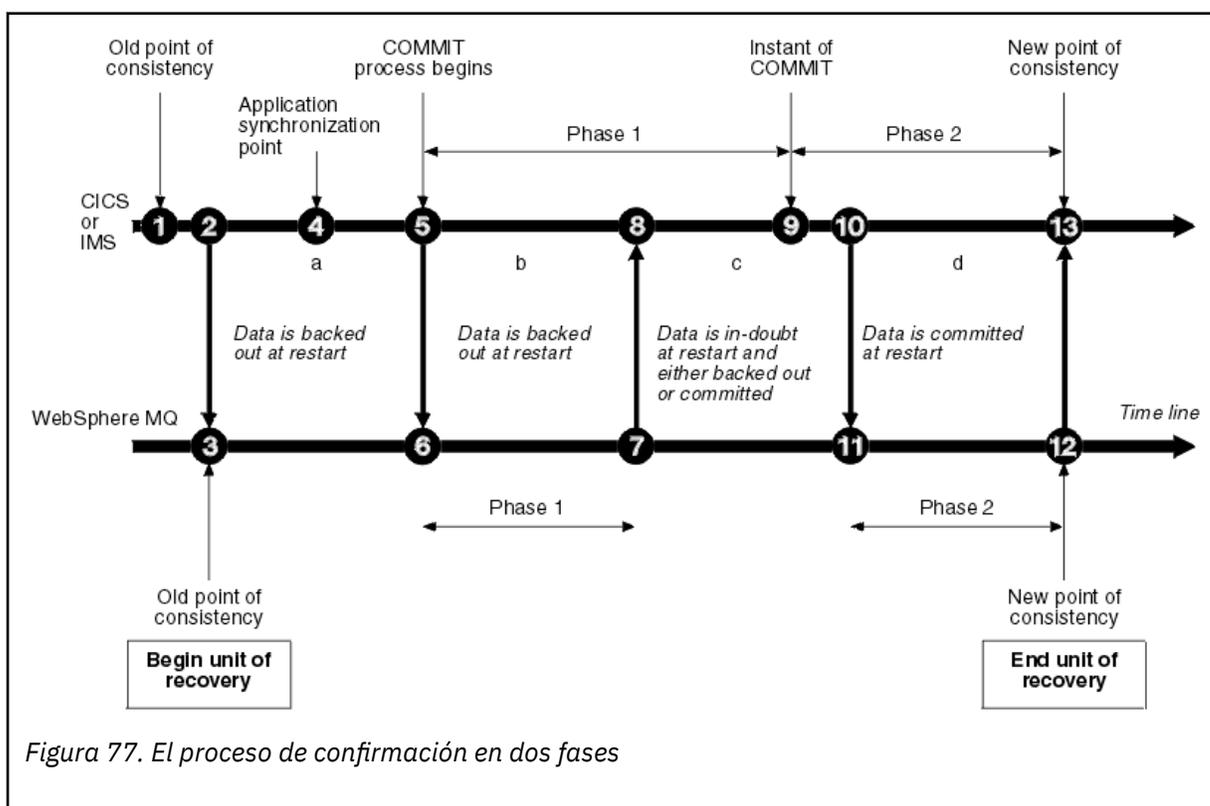


Figura 77. El proceso de confirmación en dos fases

Los números de la sección siguiente están vinculados a los que se muestran en la figura.

1. Los datos del coordinador se encuentran en un punto de coherencia.
2. Un programa de aplicación en el coordinador llama a IBM MQ para actualizar una cola añadiendo un mensaje.
3. Esto inicia una unidad de recuperación en IBM MQ.
4. El proceso continúa en el coordinador de hasta que se alcance un punto de sincronización de aplicación.
5. A continuación, el coordinador inicia el proceso de confirmación. Los programas de CICS utilizan un mandato SYNCPOINT o una terminación de aplicación normal para iniciar la confirmación. Los programas de IMS pueden iniciar la confirmación utilizando una llamada CHKP, una llamada SYNC, una llamada GET UNIQUE a IOPCB o bien una terminación de aplicación normal. Se inicia la fase 1 del proceso de confirmación.
6. A medida que el coordinador inicia el proceso de la fase 1, IBM MQ también lo hace.
7. IBM MQ completa con éxito la fase 1, escribe este hecho en su registro y lo notifica al coordinador.
8. El coordinador recibe la notificación.
9. El coordinador completa satisfactoriamente su proceso de la fase 1. Ahora los dos subsistemas están de acuerdo en confirmar los cambios de datos, porque ambos han completado la fase 1 y podrían recuperarse de cualquier error. El coordinador registra el momento de la confirmación en su registro; la decisión irrevocable de dos subsistemas de realizar los cambios.  
Ahora el coordinador empieza la fase 2 del proceso: la confirmación real.
10. El coordinador notifica a IBM MQ que comience su fase 2.
11. IBM MQ registra el inicio de la fase 2.
12. La fase 2 se ha completado satisfactoriamente y ahora constituye un nuevo punto de coherencia para IBM MQ. A continuación, IBM MQ notifica al coordinador que ha finalizado el proceso de la fase 2.
13. El coordinador finaliza el proceso de la fase 2. Ahora, los datos controlados por ambos subsistemas son coherentes y están disponibles para otras aplicaciones.

## **Cómo se mantiene la consistencia después de una terminación anómala**

Cuando un gestor de colas se reinicia tras una terminación anómala, debe determinar si confirmar o si restituir las unidades de recuperación que estaban activas en el momento de la terminación. Para algunas unidades de recuperación, IBM MQ tiene suficiente información para tomar la decisión. En el caso de otras, no, y debe obtener información del coordinador cuando se restablece la conexión.

En la [Figura 77 en la página 264](#) se muestran cuatro periodos dentro de las dos fases: a, b, c y d. El estado de una unidad de recuperación depende del periodo en el que se haya producido la terminación. El estado puede ser uno de los siguientes:

### **En curso**

El gestor de colas ha finalizado antes de finalizar la fase 1 (periodo a o b); durante el reinicio, IBM MQ restituye las actualizaciones

### **Dudoso**

El gestor de colas ha finalizado después de que finalizara la fase 1 y antes de que se iniciara la fase 2 (periodo c); sólo el coordinador sabe si el error se ha producido antes o después de la confirmación (punto 9). Si se ha producido antes, IBM MQ debe restituir sus cambios; si se produce después, IBM MQ debe realizar sus cambios y confirmarlos. Durante el reinicio, IBM MQ espera información del coordinador antes de procesar esta unidad de recuperación.

### **En confirmación**

El gestor de colas finaliza después de que haya comenzado su propio proceso de la fase 2 (periodo d); realiza los cambios confirmados.

## En retroceso

El gestor de colas ha finalizado después de que se haya iniciado una copia de seguridad de una unidad de recuperación, pero antes de que el proceso se haya completado (no se muestra en la figura) durante el reinicio, IBM MQ continúa realizando una copia de seguridad de los cambios.

## z/OS Qué sucede durante la terminación en IBM MQ for z/OS

Un gestor de colas termina de forma normal en respuesta al mandato STOP QMGR. Si un gestor de colas se detiene por cualquier otro motivo, la terminación es anómala.

Tenga en cuenta que durante la terminación del gestor de colas, IBM MQ emite internamente el mandato

```
DISPLAY CONN(*) TYPE(CONN) ALL WHERE (APPLTYPE NE SYSTEMAL)
```

de forma que sepa las hebras que pueden impedir que el gestor de colas complete el proceso de apagado.

SYSTEMAL coincide con APPLTYPES de SYSTEM o de CHINIT, por lo que el mandato DISPLAY CONN, que filtra tipos de aplicación que no coinciden con SYSTEMAL, vuelve a la información de registro de trabajo sobre las hebras que podrían impedir el cierre normal.

### Terminación normal

En una terminación normal, IBM MQ detiene toda la actividad de forma ordenada. Puede detener IBM MQ utilizando la modalidad de desactivación temporal, forzada o de reinicio. Los efectos se proporcionan en [Tabla 23](#) en la [página 266](#).

<i>Tabla 23. Terminación utilizando QUIESCE, FORCE y REINICIO</i>			
<b>Tipo de hebra</b>	<b>QUIESCE</b>	<b>FORCE</b>	<b>RESTART</b>
Hebras activas	Ejecutar hasta terminación	Restituir	Restituir
Hebras nuevas	Pueden iniciarse	No permitido	No permitido
Conexiones nuevas	No permitido	No permitido	No permitido

A las aplicaciones por lotes se les notifica si se produce una terminación mientras la aplicación sigue conectada.

Con CICS, una hebra actual solamente se ejecuta al final de la unidad de recuperación. Con CICS, al detener un gestor de colas en modalidad de desactivación temporal, se detiene el adaptador de CICS y así sucesivamente, si una tarea activa contiene más de una unidad de recuperación, la tarea no se ejecutará hasta el final.

Si detiene un gestor de colas en modalidad forzada o de reinicio, no se asignan hebras nuevas y el trabajo en hebras conectadas se retrotrae. La utilización de estos nodos puede crear unidades pendientes de recuperación para hebras que se encuentran entre fases de proceso de confirmación. Se resuelven cuando IBM MQ se vuelve a conectar con el subsistema de control CICS, IMS o RRS.

Cuando se detiene un gestor de colas, en cualquier modalidad, los pasos son:

1. Las conexiones finalizan.
2. IBM MQ deja de aceptar mandatos.
3. IBM MQ garantiza que se hayan completado las actualizaciones pendientes de los conjuntos de páginas.
4. El mandato DISPLAY USAGE emitido internamente por IBM MQ para que la RBA de reinicio se grabe en el registro de consola de z/OS.
5. Se toma el punto de comprobación de conclusión y el conjunto de datos de arranque se actualiza.

Las finalizaciones que especifican la modalidad de desactivación temporal no afectan a las unidades pendientes de recuperación. Toda unidad que esté en duda seguirá en duda.

### **Terminación anómala**

Una terminación anómala puede dejar datos en un estado incoherente, por ejemplo:

- Se ha interrumpido una unidad de recuperación antes de alcanzar el punto de coherencia.
- No se han escrito datos confirmados en conjuntos de páginas.
- Se han escrito datos no confirmados en conjuntos de páginas.
- Se ha interrumpido un programa de aplicación entre la fase 1 y la fase 2 del proceso de confirmación, dejando la unidad de recuperación pendiente.

IBM MQ resuelve las incoherencias de datos que surgen de la terminación anómala durante el reinicio y la recuperación.

## **Qué sucede durante el reinicio y la recuperación en IBM MQ for z/OS**

IBM MQ utiliza su registro de recuperación y el conjunto de datos de rutina de carga (BSDS) para determinar cuándo se debe recuperar cuando se reinicia. El BSDS identifica los conjuntos de datos de registro activos y de archivado así como la ubicación del punto de comprobación IBM MQ más reciente en el registro.

### **Introducción al reinicio y a la recuperación**

Después de inicializar IBM MQ, el proceso de reinicio del gestor de colas tiene lugar de la siguiente manera:

- Inicialización del registro
- Recreación del estado actual
- Recuperación de registro hacia adelante
- Recuperación inversa del registro
- Reconstrucción del índice de colas

Cuando se ha completado la recuperación:

- Los cambios validados se reflejan en los datos.
- La actividad pendiente se refleja en los datos. Sin embargo, los datos están bloqueados y no se pueden utilizar hasta que IBM MQ reconozca y actúe en la decisión dudosa.
- Los cambios interrumpidos durante su curso y los cambios con terminaciones anómalas se han eliminado de las colas. Los mensajes son coherentes y se pueden utilizar.
- Se ha adoptado un nuevo punto de comprobación.
- Se han compilado nuevos índices para las colas indexadas que contienen mensajes permanentes (descritos en [“Reconstrucción de índices de colas”](#) en la página 269).

Si se utilizan dos BSDS duales, IBM MQ comprueba la coherencia de las indicaciones de fecha y hora en el BSDS:

- Si ambas copias del BSDS son actuales, IBM MQ prueba si las dos indicaciones de fecha y hora son iguales. Si no lo son, IBM MQ emite el mensaje CSQJ120E y termina. Esto puede ocurrir cuando las dos copias del conjunto de datos de arranque se actualizan en volúmenes de DASD separados y uno de los volúmenes se ha restaurado mientras el gestor de colas estaba detenido. IBM MQ detecta la situación durante el reinicio.
- Si se ha reasignado una copia del conjunto de datos de arranque y el registro ha seguido en un único conjunto de datos de arranque, es posible que surja un problema. Si ambas copias del BSDS se mantienen en un solo volumen y éste se ha restaurado o si ambas copias de BSDS se han

restaurado por separado, es posible que IBM MQ no detecte la restauración. En tal caso, los registros de anotaciones no anotados en el conjunto de datos de arranque serían desconocidos para el sistema.

o se realiza ninguna notificación a las aplicaciones por lotes cuando se produce un reinicio *después* de que la aplicación haya solicitado una conexión.

## El rango de registros necesario para la recuperación

Durante el reinicio, el rango de datos de registro que debe leerse depende de muchos factores:

- Cuando se produce una terminación anómala, normalmente hay muchas unidades de trabajo incompletas en el sistema. Como se ha descrito anteriormente, el proceso de reinicio hará que el sistema pase a un estado de consistencia, lo que puede implicar la restitución de unidades de trabajo en curso o la recuperación de bloqueos de unidades de trabajo pendientes. La recuperación de la unidad de trabajo requiere que todos los registros de anotaciones de unidad de trabajo para unidades en curso y pendientes estén disponibles. IBM MQ 'derivará' unidades antiguas de trabajo, de forma que la recuperación de la unidad de trabajo se pueda realizar utilizando un rango de datos de registro mucho menor.
- Cuando se produce una terminación anómala, normalmente hay muchas actualizaciones persistentes que sólo se conservan en la memoria caché de la agrupación de almacenamiento intermedio. Aún no se han escrito en disco. Estos cambios deben leerse desde el registro y deben volverse a aplicar a los datos conservados en conjuntos de páginas. Las direcciones de byte relativas de recuperación del conjunto de páginas en el punto de comprobación describen la dirección de byte relativa de registro más baja para actualizar los conjuntos de páginas en un estado coherente.
- Si se han introducido conjuntos de páginas antiguos en el sistema, por ejemplo, se ha introducido una copia de seguridad del conjunto de páginas para poder recuperarse de una anomalía de soporte, deberán leerse todos los cambios del registro desde la hora en que se realizó de la copia de seguridad. Estos cambios se aplican a los datos incluidos en el conjunto de páginas que se está recuperando. Las direcciones de byte relativas de recuperación de conjunto de páginas conservadas en la página 0 del conjunto de páginas describen la dirección de byte relativa de registro más baja para la recuperación del soporte de un conjunto de páginas.
- Si utiliza mensajes permanentes en colas compartidas, necesitará un rango de datos de registro para recuperar CFSTRUCT que conserven mensajes permanentes. Los datos de registro más recientes que serían necesarios para realizar una recuperación de CFSTRUCT son los correspondientes aproximadamente a la hora de la antigua CFSTRUCT BACKUP.

Durante la ejecución normal, el mandato DISPLAY USAGE TYPE(DATASET) puede utilizarse para ver el rango de registro de recuperación asociado con estos factores (no puede proporcionar información debido a la reintroducción de conjuntos de páginas antiguos, por supuesto). Para evitar todos los problemas que pudieran prolongar el reinicio de un gestor de colas si se produce una terminación anómala, supervise regularmente la salida de valores de DISPLAY USAGE TYPE(DATASET).

Además, el gestor de colas emite mensajes informativos relacionados con estos factores:

- CSQJ160I y CSQJ161I advierten sobre unidades de trabajo de larga ejecución.
- CSQR026I y CSQR027I proporcionan información sobre si estas unidades de trabajo de larga ejecución se han derivado satisfactoriamente.
- CSQE040I y CSQE041E advierten que las copias de seguridad de la estructura se están quedando obsoletas y, por consiguiente, una operación RECOVER CFSTRUCT se prolongaría mucho.

## Cómo determinar qué aplicación tiene una unidad de trabajo de larga ejecución

Se puede determinar la aplicación con la unidad de trabajo de larga ejecución. Para hacerlo, utilice el mandato DISPLAY CONN.

El mandato DISPLAY CONN devuelve información de conexión de todas las aplicaciones conectadas al gestor de colas, junto con información adicional que le ayuda a determinar qué aplicación(es) actualmente tienen una unidad de trabajo de larga ejecución. La información devuelta por el mandato DISPLAY CONN es similar a la información que devuelve el mandato DISPLAY QSTATUS, pero la diferencia

principal es que DISPLAY CONN muestra información sobre objetos e información transaccional para una conexión en particular, en lugar de detalles sobre qué conexiones están asociados a un objeto determinado.

Para cada aplicación conectada, el mandato DISPLAY CONN devuelve la siguiente información:

- Información básica que incluye el ID de conexión y el PID.
- Información sobre transacciones para dicha conexión, incluyendo la hora y la fecha de creación de la transacción (es decir, cuando se ha realizado la primera MQGET/PUT bajo el punto de sincronismo) y cuando se ha escrito la transacción por primera vez en registro.
- Información del tiempo de registro que indique qué aplicación aún tiene una unidad de trabajo de larga ejecución.
- Una lista de todos los objetos que la conexión actualmente tiene abierta. Se devuelven detalles sobre cada objeto como un mensaje separado, con el ID de conexión utilizado como una clave. Puesto que existen diferentes tipos de objeto como colas y gestores de colas, la información visualizada con el objeto es específico de su tipo de objeto.

## Reconstrucción de índices de colas

Para aumentar la velocidad de las operaciones de MQGET en una cola en la que los mensajes no se recuperan secuencialmente, puede especificar que desea que IBM MQ mantenga un índice del mensaje o los identificadores de correlación o el ID de grupo para todos los mensajes de esa cola.

Cuando se reinicia un gestor de colas, estos índices se reconstruyen para cada cola. Esto sólo se aplica a los mensajes permanentes; los mensajes no permanentes se suprimen durante el reinicio. Si las colas indexadas contienen un número elevado de mensajes permanentes, se aumenta el tiempo necesario para reiniciar el gestor de colas.

Puede elegir tener índices reconstruidos de forma asíncrona en el reinicio del gestor de colas utilizando el parámetro QINDXBLD de la macro CSQ6SYSP. Si establece QINDXBLD=NOWAIT, IBM MQ se reinicia sin esperar a que se reconstruyan los índices.

## **Cómo se resuelven las unidades pendientes de recuperación**

Si IBM MQ pierde su conexión con otro gestor de recursos, normalmente intenta recuperar todos los objetos incoherentes durante el reinicio.

Si IBM MQ pierde su conexión con CICS, IMS o RRS, normalmente intenta recuperar todos los objetos incoherentes durante el reinicio. La información necesaria para resolver unidades pendientes de recuperación debe proceder del sistema de coordinación. En las siguientes secciones se describe el proceso de resolución en distintos entornos.

- [Cómo se resuelven las unidades pendientes de recuperación de CICS](#)
- [Cómo se resuelven las unidades pendientes de recuperación de IMS](#)
- [Cómo se resuelven las unidades pendientes de recuperación de RRS](#)
- [Cómo se resuelven las unidades pendientes de recuperación con una disposición de unidad de recuperación GROUP](#)

## **Cómo se resuelven las unidades pendientes de recuperación de CICS**

En algunas circunstancias, CICS no puede ejecutar el proceso IBM MQ para resolver unidades de recuperación dudosas. Cuando esto sucede, IBM MQ envía uno de los mensajes siguientes:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E

seguido del mensaje CSQC408I.

Para obtener información detallada de lo que significan estos mensajes, consulte el manual [IBM MQ for z/OS mensajes, códigos de finalización y de motivo](#).

La resolución de unidades pendientes no afecta a los recursos de CICS. CICS tiene el control de la coordinación de la recuperación y, cuando se reinicia, confirma o restituye automáticamente cada unidad, en función de si había un registro de anotaciones marcando el inicio de la confirmación. La existencia de objetos dudosos no bloquea los recursos de CICS mientras que IBM MQ se está reconectando.

Una de las funciones del adaptador de CICS es mantener los datos sincronizados entre CICS y IBM MQ. Si un gestor de colas termina de forma anómala mientras está conectado a CICS, es posible que CICS confirme o restituye una copia de seguridad del trabajo sin que IBM MQ sea consciente de ello. Cuando el gestor de colas se reinicia, ese trabajo se denomina *pendiente*.

IBM MQ no puede resolver estas unidades dudosas de recuperación (es decir, confirmar o restituir los cambios realizados en los recursos de IBM MQ) hasta que se reinicie o se vuelva a conectar CICS.

Se inicia un proceso para resolver las unidades pendientes de recuperación durante el inicio del adaptador de CICS. El proceso se inicia cuando el adaptador solicita una lista de las unidades pendientes de recuperación. A continuación:

- El adaptador recibe una lista de unidades dudosas de recuperación para este ID de conexión de IBM MQ y las pasa a CICS para su resolución.
- CICS compara las entradas de esta lista con las entradas de su propio registro. CICS determina, a partir de su propia lista, qué acción hay que emprender para cada unidad pendiente de recuperación.

En el caso de todas las unidades resueltas, IBM MQ actualiza las colas según sea necesario y libera los bloqueos correspondientes. Las unidades no resueltas pueden permanecer después del reinicio. Resuélvalas con los métodos que se describen en [Administrar IBM MQ for z/OS](#).

## **Cómo se resuelven las unidades pendientes de recuperación de IMS**

La resolución de unidades pendientes de recuperación en IMS no afecta a los recursos DL/I. IMS tiene el control de la coordinación de la recuperación y, cuando se reinicia, confirma automáticamente o restituye el trabajo de Data Language/I incompleto. La decisión de confirmar o restituir regiones en línea (vía de acceso no rápida) se toma con la presencia o la ausencia de los tipos de registros de anotaciones X'3730' y X'3801' de IMS. La existencia de unidades de recuperación dudosas no implica que los registros DL/I estén bloqueados hasta que se conecte IBM MQ.

Durante el reinicio del gestor de colas, IBM MQ crea una lista de las unidades pendientes de recuperación. IMS crea su propia lista de entradas de recuperación residuales (RRE). Las RRE se registran en los puntos de comprobación de IMS hasta que se resuelvan todas las entradas.

Durante la reconexión de una región IMS a IBM MQ, IMS indica a IBM MQ si se deben confirmar o restituir unidades de trabajo marcadas en IBM MQ como dudosas.

Cuando se resuelven las unidades pendientes:

1. Si IBM MQ reconoce que ha marcado una entrada para su confirmación y IMS la ha marcado para su restitución, IBM MQ emite el mensaje CSQQ010E. IBM MQ emite este mensaje para todas las incoherencias de este tipo entre IBM MQ e IMS.
2. Si IBM MQ tiene todas las unidades dudosas restantes, el adaptador emite el mensaje CSQQ008I.

En el caso de todas las unidades resueltas, IBM MQ actualiza las colas según sea necesario y libera los bloqueos correspondientes.

IBM MQ mantiene bloqueos sobre el trabajo dudoso que no se ha resuelto. Esto puede provocar un retraso en el sistema si se mantienen bloqueos importantes. La conexión permanece activa para que pueda resolver las RRE de IMS. Recupere las hebras pendientes mediante los métodos descritos en [Administrar IBM MQ for z/OS](#).

Todo el trabajo pendiente debe resolverse a menos que haya problemas de software u operativos, tal como un arranque en frío de IMS. La resolución pendiente por parte de la región de control de IMS se produce en dos casos:

1. Al principio de la conexión con IBM MQ, durante la cual la resolución se realiza de forma síncrona.
2. Cuando un programa termina de forma anómala, durante el cual la resolución se hace de forma asíncrona.

## **Cómo se resuelven las unidades pendientes de recuperación de RRS**

Una de las funciones del adaptador RRS es mantener los datos sincronizados entre IBM MQ y otros gestores de recursos que participan en RRS. Si se produce una anomalía cuando IBM MQ ha completado la fase uno de la confirmación y está esperando una decisión de RRS (el coordinador de confirmación), la unidad de recuperación entra en el estado dudoso.

Cuando se restablece la comunicación entre RRS y IBM MQ, RRS confirma automáticamente o restituye cada unidad de recuperación, en función de si había un registro de anotaciones marcando el principio de la confirmación. IBM MQ no puede resolver estas unidades pendientes de recuperación (es decir, confirmar o restituir los cambios realizados en los recursos de IBM MQ) hasta que se restablezca la conexión con RRS.

En determinadas circunstancias, los servicios de recuperación de recursos no pueden resolver las unidades pendientes de recuperación. Cuando esto sucede, IBM MQ envía uno de los siguientes mensajes a la consola de z/OS:

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Para obtener información detallada de lo que significan estos mensajes, consulte el manual [IBM MQ for z/OS mensajes, códigos de finalización y de motivo](#).

En el caso de todas las unidades de recuperación resueltas, IBM MQ actualiza las colas según sea necesario y libera los bloqueos correspondientes. Las unidades de recuperación no resueltas pueden permanecer después del reinicio. Resuélvalas con el método que se describe en [Administrar IBM MQ for z/OS](#).

## **Cómo se resuelven las unidades pendientes de recuperación con una disposición de unidad de recuperación GROUP**

Las transacciones pendientes que tienen una disposición de unidad de recuperación GROUP pueden ser resueltas por el coordinador de transacción mediante cualquier gestor de colas en el grupo de compartición de colas (QSG) donde está habilitado el atributo del gestor de colas GROUPUR. Siempre que un coordinador de transacción se reconecta, normalmente solicita una lista de todas las transacciones dudosas pendientes y, a continuación, las resuelve utilizando la información de sus registros.

Cuando un coordinador de transacción, que se ha conectado con una disposición de unidad de recuperación GROUP, solicita la lista de transacciones pendientes, la lista devuelta incluye todas las transacciones pendientes con una disposición de unidad de recuperación GROUP que existen en todo el grupo de compartición de colas. Esta lista no depende del gestor de colas en el que se hayan iniciado dichas transacciones pendientes. Un gestor de colas que procese dicha solicitud compila la lista comunicándose con todos los otros gestores de colas activos del grupo de compartición de colas utilizando SYSTEM.QSG.UR.RESOLUTION.QUEUE. A continuación, el gestor de colas lee los registros de todos los gestores de colas inactivos (desde su último punto de comprobación) para identificar cualquier transacción pendiente adicional de la que se haya informado de que ha estado activa.

Cuando un coordinador de transacción solicita la resolución de una transacción pendiente, el gestor de colas al que está conectado identifica si la transacción se ha originado en sí misma y, si es así, la resuelve de la misma manera que las transacciones con una disposición de unidad de recuperación QMGR. Si la transacción se ha originado en otro gestor de colas activo en el QSG, se envía a dicho gestor de colas una solicitud para completar la resolución utilizando SYSTEM.QSG.UR.RESOLUTION.QUEUE. En el caso en el que la transacción se haya originado en un gestor de colas inactivo en el QSG, todo trabajo de cola compartida se resuelve inmediatamente y se coloca en SYSTEM.QSG.UR.RESOLUTION.QUEUE una solicitud para resolver cualquier trabajo de cola privada pendiente. El gestor de colas inactivo procesa dicha solicitud al reiniciarse antes de aceptar un trabajo nuevo. En este escenario, los registros del gestor de colas original todavía reflejan que la unidad de recuperación está pendiente hasta que haya reiniciado y ha procesado la solicitud.

## Recuperación de la cola compartida

Utilice este tema para comprender la recuperación de IBM MQ y la resiliencia de diversos componentes en el entorno de grupo de compartición de colas.

- [“Recuperación transaccional” en la página 272](#)
- [“Recuperación de iguales” en la página 272](#)
- [“Definiciones de cola compartida” en la página 273](#)
- [“Registro” en la página 273](#)
- [“Anomalías de estructura y del recurso de acoplamiento” en la página 273](#)
- [“Escenarios de anomalías de estructura” en la página 274](#)
- [“Resiliencia a anomalías de conectividad del recurso de acoplamiento” en la página 275](#)
- [“Gestión de la resiliencia a anomalías de conectividad del recurso de acoplamiento” en la página 276](#)
- [“Comportamiento operativo” en la página 278](#)

### Recuperación transaccional

Cuando una aplicación emite una llamada MQBACK o termina de forma anómala (por ejemplo, debido a un EXEC CICS ROLLBACK o una terminación anómala de IMS), la información a nivel de hebra almacenada en el gestor de colas garantiza que la unidad de trabajo en curso no se retrotraiga. Las operaciones MQPUT y MQGET dentro el punto de sincronización en colas compartidas se retrotraen de la misma forma que las actualizaciones en colas no compartidas.

### Recuperación de iguales

Si un gestor de colas falla, se desconecta de forma anómala de las estructuras de recurso de acoplamiento a las que actualmente está conectado. Si la conexión entre la instancia de z/OS y el recurso de acoplamiento falla (por ejemplo, un error en el enlace físico o el apagado de un recurso de acoplamiento o partición), también se detecta como terminación anómala de la conexión entre el gestor de colas y las estructuras del recurso de acoplamiento implicadas. Otros gestores de colas del mismo grupo de compartición de colas que permanecen conectados con dicha estructura detectan la desconexión anómala y todos intentan iniciar la *recuperación de iguales* para el gestor de colas fallido en la estructura. Sólo uno de estos gestores de colas inicia la recuperación de iguales correctamente, pero todos los demás gestores de colas cooperan en la recuperación de unidades de trabajo que eran propiedad del gestor de colas que ha fallado.

Si un gestor de colas falla cuando no hay iguales conectados a una estructura, la recuperación se realiza cuando otro gestor de colas se conecta a esa estructura o cuando se reinicia el gestor de colas que ha fallado.

La recuperación de iguales, llamada con frecuencia PLR (Peer Level Recovery), se realiza estructura por estructura y un solo gestor de colas puede participar en la recuperación de más de una estructura al

mismo tiempo. Sin embargo, el conjunto de iguales que colaboran en la recuperación de estructuras diferentes puede variar dependiendo de los gestores de colas que estaban conectados a las diferentes estructuras en el momento de la anomalía.

Cuando el gestor de colas fallido se reinicia, se vuelve a conectar con las estructuras con las que estaba conectado en el momento de la anomalía, y recupera toda unidad de trabajo restante no resuelta que no se recuperara por recuperación de iguales.

La recuperación de igual es un proceso compuesto por varias fases. Durante la primera fase, se recuperan las unidades de trabajo que se habían procesado además de la fase en curso; ello puede implicar mensajes de confirmación para unidades de trabajo que estén en confirmación y mensajes de bloqueo para unidades de trabajo que estén pendientes. Durante la segunda fase, se comprueban las colas que tenían hebras activas frente a estos mensajes en el gestor de colas fallido, los mensajes no confirmados relacionados con unidades de trabajo en curso se retrotraen y se restablece la información sobre manejadores activos en colas compartidas en el gestor de colas fallido. Esto significa que IBM MQ restablece los indicadores de que el gestor de colas anómalo tenía una cola compartida abierta de entrada exclusiva, lo que permite a otros gestores de colas activos abrir la cola para la entrada.

## Definiciones de cola compartida

Los objetos de cola que representan los atributos de una cola compartida se guardan en el repositorio compartido de Db2 utilizados por el grupo de compartición de colas. Asegúrese de que existen procedimientos adecuados para la copia de seguridad y la recuperación de las tablas de Db2 utilizadas para contener objetos de IBM MQ. También puede utilizar el programa de utilidad CSQUTIL de IBM MQ para crear mandatos MQSC para la reproducción en un gestor de colas para redefinir objetos de IBM MQ, incluidas las definiciones de grupo y cola compartidas almacenadas en Db2.

## Registro

Los grupos de compartición de colas pueden dar soporte a mensajes permanentes, porque los mensajes en colas compartidas se pueden registrar en los registros de los gestores de colas.

## Anomalías de estructura y del recurso de acoplamiento

Se pueden notificar dos tipos de anomalías para la estructura del recurso de acoplamiento (CF): error de estructura y pérdida de conectividad. Los servicios sysplex para compartición de datos (XES) informan a IBM MQ de una anomalía del recurso de acoplamiento (CF) o de una anomalía de estructura de acoplamiento (CF) con un suceso de anomalía de estructura. Si XES crea un suceso de pérdida de conectividad, ello no indica necesariamente que hay un problema con la estructura, es posible que no hay ninguna conexión disponible para comunicarse con la estructura. Es posible que no todos los gestores de colas reciban un suceso de pérdida de conectividad de la estructura; dependerá de la configuración de las conexiones con el recurso de acoplamiento. También se puede recibir un suceso de pérdida de conectividad debido a mandatos del operador, por ejemplo VARY PATH OFFLINE o CONFIG CHP OFFLINE.

Las estructuras CF que IBM MQ utiliza se pueden configurar para utilizar la duplicación gestionada por el sistema. Esto significa que si hay una única anomalía, el proceso de migración tras error gestionado por el sistema oculta la anomalía de una estructura o la pérdida de conectividad y no se informa al gestor de colas sobre anomalía. Si se produce una anomalía en las dos instancias de una estructura o conexión duplicada, el gestor de colas recibe el suceso apropiado y lo maneja de la misma forma que un suceso de anomalía para una estructura de una sola cara. Obtendrá detalles sobre cómo el gestor de colas maneja estos sucesos en [Escenarios](#).

En poco probable caso de una anomalía en un recurso de acoplamiento o estructura, se pierden todos los mensajes no permanentes almacenados en las estructuras de aplicación afectadas. Puede recuperar mensajes permanentes utilizando el mandato RECOVER CFSTRUCT. Si ha fallado una estructura de aplicación recuperable, se impedirá cualquier actividad de aplicación adicional para esta estructura hasta que se haya recuperado la estructura.

Para asegurarse de poder recuperar una estructura CF en un período de tiempo razonable, haga copias de seguridad frecuentes, utilizando el mandato BACKUP CFSTRUCT. Puede elegir realizar copias de seguridad en todos los gestores de colas del grupo de compartición de colas o dedicar un gestor de colas para realizar todas las copias de seguridad. Automatice el proceso de realizar copias de seguridad para garantizar que se haga de forma regular.

Cada copia de seguridad se graba en el conjunto de datos de registro activo del gestor de colas que realiza la copia de seguridad. El repositorio de Db2 de la cola compartida registra el nombre de la estructura de CF de la que se está realizando la copia de seguridad, el nombre del gestor de colas que realiza la copia de seguridad, el rango RBA de esta copia de seguridad en el registro de dicho gestor de colas y la hora de la copia de seguridad.

La estructura de administración contiene información sobre las unidades de trabajo incompletas en las colas compartidas en el momento en que se produce cualquier anomalía en la estructura de la aplicación de modo que la estructura de administración debe estar disponible durante el proceso de RECOVER CFSTRUCT. Si la estructura de administración ha fallado, todos los gestores de colas del grupo de compartición de colas deben haber reconstruido sus entradas de estructura de administración antes de emitir el mandato RECOVER CFSTRUCT.

Los gestores de colas vuelven a crear sus entradas de la estructura de administración de forma automática sin finalizar. Si en el momento en que se produce el error un gestor de colas no está en ejecución, otro gestor de colas puede volver a crear las entradas de la estructura de administración en el grupo de compartición de colas que se ejecuta en el mismo nivel o en un nivel superior.

Para recuperar una estructura de aplicación, emita un mandato RECOVER CFSTRUCT para el gestor de colas al que desea realizar la recuperación. Puede recuperar una estructura de recurso de acoplamiento única o puede recuperar varias estructuras de recurso de acoplamiento simultáneamente. La recuperación puede realizarla mediante cualquier gestor de colas del grupo de compartición de colas, no es necesario que sea el gestor de colas que ha realizado la copia de seguridad, u otro que haya estado conectado previamente a la estructura que ha fallado.

El mandato RECOVER CFSTRUCT utiliza la copia de seguridad, ubicada mediante la información de repositorio de Db2 (por ello, Db2 debe estar disponible en el gestor de colas en la que se está efectuando la recuperación) y la recupera hasta el punto de la anomalía.

El mandato RECOVER CFSTRUCT lo hace aplicando registros de anotaciones de cada gestor de colas del grupo de compartición de colas que ha realizado una llamada MQPUT o MQGET entre el inicio de la copia de seguridad y el momento de la anomalía a cualquier cola compartida que se correlacione con la estructura de recurso de acoplamiento. Es posible que la fusión resultante de los registros requiera leer una considerable cantidad de datos de registro porque se leerán todos los datos de registro escritos por gestores de colas participantes desde la copia de seguridad. Se recomienda encarecidamente que realice copias de seguridad frecuentes (por ejemplo, a cada hora), especialmente si hay mensajes muy grandes dentro de la copia de seguridad.

## Escenarios de anomalías de estructura

### Escenarios

Si se notifica una anomalía para una estructura CF, la acción emprendida por los gestores de colas conectados dependerá de lo siguiente:

- El tipo de anomalía notificado por el componente XES de z/OS a IBM MQ.
- El tipo de estructura (de aplicación o administración)
- El nivel del gestor de colas
- El CFLEVEL del objeto CFSTRUCT de IBM MQ (2, 3, 4 ó 5. Este no es el CFLEVEL del microcódigo de CFCC)
- El atributo RECAUTO de un objeto CFSTRUCT de IBM MQ en CFLEVEL (5)

Los siguientes escenarios describen lo que sucede cuando se informa sobre una anomalía de la estructura de administración:

- Si se recibe un suceso de anomalía de estructura para la estructura de administración y la estructura se vuelve a asignar y crear automáticamente sin que el gestor de colas termine. XES asigna una instancia de la estructura cuando un gestor de colas intenta conectarse a la misma.

Cuando el gestor de colas se ha conectado a la nueva instancia de la estructura, el gestor de colas sólo graba en la estructura las entradas dirigidas a él. Este procesamiento es efectuado por el gestor de colas y no forma parte del proceso de reconstrucción de XES.

Si no se estaba ejecutando un gestor de colas en el momento de la anomalía o si termina antes de que se haya completado la recuperación de parte de la estructura de administración, las entradas de la estructura de administración las puede volver a crear otro gestor de colas del grupo de compartición de colas.

Las entradas de la estructura de administración de un gestor de colas solo las puede volver a crear otro gestor de colas que se ejecute en el mismo nivel o superior. Si otro gestor de colas del grupo de compartición de colas no puede volver a crear las entradas de la estructura de administración de un gestor de colas, reinicie el gestor de colas, de modo que pueda volver a completar la creación de su parte de la estructura.

Se suspenden determinadas acciones hasta que se vuelven a crear las entradas de la estructura de la administración para todos los gestores de colas. Las acciones suspendidas incluyen lo siguiente:

- Apertura y cierre de colas compartidas.
- Confirmación de unidades de recuperación de restitución.
- Aplicaciones serializadas que se conectan o se desconectan del gestor de colas.
- Copia de seguridad o recuperación de una estructura de aplicación.

Todas las aplicaciones serializadas que ya se hayan conectado al gestor de colas pueden continuar el proceso. Cualquier aplicación serializada que intente conectarse con los parámetros MQCNO\_SERIALIZE\_CONN\_TAG\_QSG o MQCNO\_RESTRICT\_CONN\_TAG\_QSG recibe el código de retorno MQRC\_CONN\_TAG\_NOT\_USABLE.

Cuando las entradas de la estructura de administración para el gestor de colas se hayan reconstruido, se reanudarán las acciones suspendidas.

Los siguientes escenarios describen lo que sucede cuando se notifica una anomalía para una estructura de aplicación:

- Si se recibe un suceso de anomalía de estructura para una estructura de aplicación y CFLEVEL es 1 o 2, el gestor de colas termina. Reinicie el gestor de colas. El primer gestor de colas que intente conectarse con la estructura de nuevo hará que XES se asigne a una instancia nueva de la estructura.
- Si se recibe el suceso de una anomalía de estructura para una estructura de aplicación y CFLEVEL es 3, 4, o 5, los gestores de colas conectados con la estructura siguen ejecutándose. Las aplicaciones que no utilizan las colas de la estructura que ha fallado pueden continuar el proceso normal.

Sin embargo, las aplicaciones que intenten realizar operaciones en las colas de la estructura anómala recibirán un error MQRC\_CF\_STRUC\_FAILED hasta que se vuelva a crear correctamente la estructura, y la aplicación pueda volver a abrir las colas.

La reconstrucción de la estructura se inicia automáticamente para las estructuras de aplicaciones CFLEVEL(5) definidas con RECAUTO(YES). De lo contrario, la estructura se vuelve a crear cuando se emite el mandato RECOVER CFSTRUCT.

## **Resiliencia a anomalías de conectividad del recurso de acoplamiento**

### **¿Qué quiere decir resiliencia a anomalías de conectividad del recurso de acoplamiento?**

La resiliencia a anomalías de conectividad del recurso de acoplamiento se refiere a la capacidad que tienen los gestores de colas en un grupo de compartición de colas de tolerar una pérdida de conectividad con una estructura de recurso de acoplamiento sin finalizar. Esta función también

intentará reconstruir la estructura en otro recurso de acoplamiento con una mejor conectividad con el fin de recuperar el acceso a las colas compartidas tan pronto como sea posible.

### ¿Qué es una pérdida parcial de conectividad?

IBM MQ define la pérdida parcial de conectividad como una situación en la que uno o más sistemas del sysplex pierden la conectividad con el recurso de acoplamiento en el que se asigna la estructura a la que accede el sistema, pero al menos un sistema del sysplex mantiene la conectividad con el mismo recurso de acoplamiento.

### ¿Qué es pérdida total de conectividad?

IBM MQ define una pérdida total de conectividad como una situación en la que ningún sistema del sysplex tiene conectividad con el recurso de acoplamiento y la estructura asignada en él.

### ¿Por qué conviene habilitar esta función?

La resiliencia a las anomalías de conectividad del recurso de acoplamiento mejora la disponibilidad de IBM MQ, lo que permite que las colas no compartidas permanezcan disponibles después de que un gestor de colas haya perdido la conectividad con una o más estructuras de recurso de acoplamiento. Además, los gestores de colas que pierden la conectividad con una estructura de recurso de acoplamiento intentarán automáticamente reconstruir la estructura en otro recurso de acoplamiento disponible, mejorar la disponibilidad de las colas compartidas dentro del grupo de compartición de colas.

### Consideraciones al habilitar esta función

Un gestor de colas que tolera la pérdida de conectividad con estructuras de recurso de acoplamiento sin terminar puede que no sea capaz de volver a conectarse a una estructura de recurso de acoplamiento durante algún tiempo si no hay ningún recurso de acoplamiento alternativo disponible. Las colas compartidas definidas en una estructura que ha sufrido la pérdida de conectividad permanecerán no disponibles hasta que se restaure la conectividad con la estructura. En esta situación, las aplicaciones que se conectan a miembros del grupo de compartición de colas para realizar trabajo de colas compartidas podrían encontrar que las colas compartidas a las que necesitan acceder no están disponibles. Para evitar esta situación, es aconsejable configurar los gestores de colas de modo que terminen cuando se pierda la conectividad con una estructura de recurso de acoplamiento. Esta terminación obligará a las aplicaciones a conectarse a otro miembro del grupo de compartición de colas que tenga conectividad con las estructuras de recurso de acoplamiento donde se han definido las colas compartidas que necesita la aplicación.

## Gestión de la resiliencia a anomalías de conectividad del recurso de acoplamiento

### ¿Cómo habilito esta función?

Para habilitar la resiliencia a anomalías de conectividad del recurso de acoplamiento, siga estos pasos:

1. Asegúrese de que el conjunto de datos de acoplamiento de CFRM se ha formateado para admitir la reconstrucción gestionada por el sistema. Esto permite que los gestores de colas inicien una reconstrucción gestionada por el sistema para volver a crear una estructura en un recurso de acoplamiento disponible. Utilice el mandato **DISPLAY XCF, COUPLE, TYPE=CFRM** para determinar el formato del conjunto de datos de acoplamiento CFRM. Como soporte a la reconstrucción gestionada por el sistema, el conjunto de datos de acoplamiento CFRM debe estar formateado especificando lo siguiente:

```
"ITEM NAME(SMREBLD) NUMBER(1) "
```

Consulte la documentación de [z/OS MVS Configuración de un sysplex](#) para obtener más información sobre el formato de un conjunto de datos de acoplamiento CFRM.

2. Asegúrese de que un recurso de acoplamiento alternativo esté disponible y esté en la lista de preferencias de CFRM para todas las estructuras de recurso de acoplamiento de IBM MQ. Esto permitirá que los gestores de colas intenten reconstruir las estructuras en un recurso de

acoplamiento alternativo disponible para restaurar el acceso a las estructuras tan pronto como sea posible.

Las estructuras de IBM MQ deben definirse con ENFORCE (NO) en la política CFRM, de modo que XCF pueda elegir el CF óptimo en la configuración si IBM MQ necesita reasignar la estructura.

Consulte la documentación de [z/OS MVS Configuración de un sysplex](#) para obtener más información sobre las listas de preferencias de estructura.

3. Modifique todas las estructuras del recurso de acoplamiento de aplicación que deben tolerar una pérdida de conectividad con CFLEVEL(5). Éste es el nivel mínimo que puede tolerar una pérdida de conectividad.
4. Determine los valores necesarios para los atributos **QMGR CFCONLOS** y **CFSTRUCT CFCONLOS** y alterarlos en consecuencia. El atributo **QMGR CFCONLOS** controla si se tolera la pérdida de conectividad con la estructura de administración, y el atributo **CFSTRUCT CFCONLOS** controla si cada estructura de recurso de acoplamiento de aplicación tolera la pérdida de conectividad. Si se retienen los valores predeterminados de estos atributos, el gestor de colas terminará después de la pérdida de conectividad con una estructura de recurso de acoplamiento.
5. Determine los valores necesarios para el atributo **CFSTRUCT RECAUTO** para cada estructura de recurso de acoplamiento de aplicación, y modifíquelos en consecuencia. Este atributo controla si se deben recuperar automáticamente las estructuras de recurso de acoplamiento mediante los datos registrados después de una pérdida total de conectividad. Si el valor predeterminado de este atributo se ha retenido, no se realizará ninguna recuperación automática de las estructuras de aplicación después de una pérdida total de conectividad.

### **Escenario 1: Pérdida de conectividad con la estructura de administración**

Los gestores de colas pueden tolerar la pérdida de conectividad con la estructura de administración sin terminar.

Cuando un gestor de colas pierde la conectividad con la estructura de administración y aquel se había configurado para tolerar la pérdida de conectividad con la estructura de administración, todos los miembros del grupo de compartición de colas se desconectarán de la estructura de administración. Todos los gestores de colas activos del grupo de compartición de colas intentarán volver a conectarse a la estructura de administración, haciendo que se reasigne en el recurso de acoplamiento con la mejor conectividad con todos los sistemas del sysplex, y se vuelvan a crear los datos de la estructura de administración.

**Nota:** Puede que el recurso de acoplamiento no tenga necesariamente la mejor conectividad con todos los sistemas que tienen gestores de colas activos.

Si un gestor de colas no puede volver a conectarse a la estructura de administración, por ejemplo, porque ninguno de los recursos de acoplamiento de la lista de preferencias de CFRM están disponibles para la estructura de administración, algunas operaciones de cola compartida permanecerán no disponibles hasta que el gestor de colas pueda volver a conectarse satisfactoriamente con la estructura de administración y reconstruir los datos de la estructura de administración. Esto se producirá automáticamente cuando un recurso de acoplamiento adecuado pase a estar disponible en el sistema.

No se tolera un fallo de conexión con la estructura de administración durante el arranque del gestor de colas como resultado de una falta de conectividad con el recurso de acoplamiento, ni un recurso de acoplamiento disponible no adecuado para asignar la estructura. Todos los gestores de colas activos del grupo de compartición de colas intentarán volver a conectarse con la estructura de administración, lo que hace que se reasigne en otro recurso de acoplamiento si hay alguno disponible, y se vuelvan a crear los datos de la estructura de administración.

### **Escenario 2: Pérdida de conectividad con la estructura de administración**

Se tolera la pérdida de conectividad con las estructuras de aplicación en un nivel **CFLEVEL (5)** o superior sin que el gestor de colas tenga que terminar. Los gestores de colas conectados a las estructuras de aplicación en un nivel **CFLEVEL (4)** o inferior, o las estructuras con nivel **CFLEVEL (5)** que no se hayan configurado para tolerar una pérdida de conectividad, terminarán de forma anómala con el código de razón 00C510AB cuando se pierda la conectividad con la estructura.

Cuando se pierde la conexión con una estructura de aplicación que se ha configurado para tolerar una pérdida de conectividad, se desconectarán todos los gestores de colas que han perdido la conectividad con la estructura. El comportamiento posterior del gestor de colas dependerá de si la pérdida de conectividad es parcial o total.

### **Pérdida parcial de conectividad con una estructura de aplicación**

Si la pérdida de conectividad se determina como parcial, los gestores de colas que han perdido la conectividad con la estructura intentarán iniciar una reconstrucción gestionada por el sistema para pasar la estructura a otro recurso de acoplamiento con conectividad mejorada. Si esta reconstrucción es satisfactoria, los mensajes permanentes y no permanentes de la estructura se copiarán en el otro recurso de acoplamiento, y se restaurará el acceso a las colas de la estructura. Los gestores de colas que no perdieron la conectividad permanecen conectados a la estructura; no obstante, las operaciones que accedan a la estructura se retrasarán durante el proceso de reconstrucción gestionada por el sistema.

Si una estructura de aplicación no se puede reconstruir en otro recurso de acoplamiento con conectividad mejorada, o algunos gestores de colas todavía no tienen conectividad con la estructura después de que se haya reconstruido en otro recurso de acoplamiento, las colas definidas en la estructura permanecerán no disponibles en los gestores de colas que no tienen conectividad con la estructura hasta que se restaure la conectividad con el recurso de acoplamiento. Los gestores de colas volverán a conectarse automáticamente a la estructura cuando esté disponible y se restaurará el acceso a las colas compartidas definidas en la estructura.

### **Pérdida total de conectividad con una estructura de aplicación**

Si todos los sistemas MVS del sysplex han perdido la conectividad con el recurso de acoplamiento en el que está asignada la estructura de aplicación, z/OS desasigna la estructura del recurso de acoplamiento siempre que se realiza un intento de volver a conectar con la estructura. El gestor de colas puede intentar volver a conectarse con la estructura por varias razones, por ejemplo si una aplicación intentara abrir una cola compartida, o una notificación del sistema informando de que nuevos recursos de acoplamiento han quedado disponibles. Por lo tanto, es probable que todos los mensajes no permanentes de la estructura afectada se pierdan después de la pérdida total de conectividad con una estructura de aplicación.

Las estructuras de aplicación recuperables se recuperarán automáticamente después de una pérdida total de conectividad, si se han definido con **RECAUTO (YES)**. La recuperación se iniciará de forma casi inmediata si está disponible un recurso de acoplamiento alternativo en el que asignar la estructura, o cuando dicho recurso de acoplamiento pase a estar disponible. Si una estructura no se ha definido con **RECAUTO (YES)**, la recuperación se puede iniciar emitiendo el mandato **RECOVER CFSTRUCT**. Esto recuperará todos los mensajes permanentes en la estructura, pero se perderán todos los mensajes no permanentes. Dado que este proceso implica leer el registro del gestor de colas, puede tardar algún tiempo en completarse; por lo tanto, es recomendable realizar copias de seguridad de la estructura con regularidad para reducir el tiempo que debe transcurrir hasta que se restaure el acceso a las colas compartidas en la estructura.

Los gestores de colas intentarán volver a conectarse con estructuras de aplicación no recuperables tan pronto como una aplicación intente abrir una cola compartida que se haya definido en la estructura o se reciba una notificación del sistema informando de que han pasado a estar disponibles nuevos recursos de acoplamiento. Si hubiera un recurso de acoplamiento adecuado en el que asignar la estructura, se asignaría una estructura nueva y se restauraría el acceso a las colas compartidas definidas en la estructura. Como los mensajes permanentes no se pueden colocar en colas definidas de estructuras no recuperables, se perderán todos los mensajes de las colas compartidas.

## **Comportamiento operativo**

Si un gestor de colas de IBM WebSphere MQ 7.1 o posterior, configurado para tolerar la pérdida de conectividad con una estructura de recurso de acoplamiento determinada pierde la conectividad, los miembros del grupo de compartición de colas intentan recuperarse automáticamente de la anomalía y volverse a conectar a la estructura. Esta actividad puede implicar la reasignación de la estructura en otro

recurso de acoplamiento con una mejor conectividad, si hubiera uno disponible. Sin embargo, puede que sea necesaria la intervención del operador para recuperarse de la pérdida de conectividad.

Las acciones del operador son normalmente las siguientes:

1. Resolver la causa del error que ha provocado la pérdida de conectividad.
2. Asegúrese de que un recurso de acoplamiento en el que se pueden asignar las estructuras de IBM MQ esté disponible en todos los sistemas del sysplex

Las estructuras que se han reasignado automáticamente en otro recurso de acoplamiento tras la pérdida de conectividad pueden trasladarse al recurso de acoplamiento con la conectividad óptima para todos los gestores de colas del grupo de compartición de colas. Si es necesario, esto se puede conseguir iniciando el mandato de recopilación gestionado por el sistema **SETXCF START, REBUILD**, tal como se describe en [z/OS MVS Referencia de comandos del sistema](#).

En caso de una pérdida parcial de conectividad con una estructura de aplicación, los gestores de colas que hayan perdido conectividad con la estructura intentarán iniciar una reconstrucción gestionada por el sistema. Este proceso sólo asignará la estructura en otro recurso de acoplamiento si dicho recurso de acoplamiento tiene conectividad con todos los gestores de colas activos actualmente conectados a la estructura. Por lo tanto, es posible que en el caso en que la mayoría de los gestores de colas de un grupo de compartición de colas haya perdido la conectividad con una estructura de aplicación, no podrán volver a crear la estructura en otro recurso de acoplamiento debido a que los gestores de colas están aún conectados a la estructura original. En esta situación, los gestores de colas que todavía están conectados a la estructura original se pueden concluir para permitir que se reconstruya la estructura, o se puede emitir el mandato **RESET CFSTRUCT ACTION(FAIL)** para que falle la estructura. La recuperación se puede iniciar en estructuras de aplicación emitiendo el mandato **RECOVER CFSTRUCT**.

**Nota:** Si se produce un error en la estructura y se recupera, se perderán todos los mensajes no permanentes de la estructura.

z/OS

## Conceptos de seguridad en IBM MQ for z/OS

Utilice este tema para comprender la importancia de la seguridad para IBM MQ y las implicaciones de no tener los valores de seguridad adecuados en el sistema.

### Por qué debe proteger los recursos de IBM MQ

IBM MQ maneja la transferencia de información que es potencialmente valiosa. La aplicación de seguridad garantiza que los recursos que IBM MQ posee y gestiona estén protegidos frente a accesos no autorizados. Este acceso puede conducir a la pérdida o divulgación de la información.

Asegúrese de que ningún usuario o proceso no autorizado pueda acceder o cambiar ninguno de los recursos siguientes:

- Conexiones con IBM MQ
- Objetos de IBM MQ como, por ejemplo, colas, procesos y listas de nombres
- IBM MQ Enlaces de transmisión de , es decir, canales de IBM MQ
- Mandatos de control del sistema IBM MQ
- Mensajes de IBM MQ
- Información de contexto asociada a mensajes

Para proporcionar la seguridad necesaria, IBM MQ utiliza el recurso de autorización del sistema (SAF) de z/OS para encaminar las solicitudes de autorización hacia un gestor de seguridad externa (ESM), por ejemplo, Security Server (anteriormente conocido como RACF). IBM MQ no realiza ninguna verificación de seguridad por sí mismo. Cuando se utilizan colas distribuidas o clientes, es posible que necesite medidas de seguridad adicionales, para las que IBM MQ proporciona registros de autenticación de canal, salidas de canal, el atributo de canal MCAUSER y TLS.

La decisión de permitir el acceso a un objeto la toma el ESM y IBM MQ respeta esa decisión. Si el ESM no puede tomar una decisión, IBM MQ impide el acceso al objeto.

## ¿Qué sucede si no protege los recursos de IBM MQ?

Si no hace nada respecto a la seguridad, la consecuencia más probable es que *todos* los usuarios pueden acceder y cambiar *todos* los recursos. Esto incluye no sólo a los usuarios locales, sino también a los situados en sistemas remotos que utilizan colas distribuidas o clientes distribuidos, donde los controles de seguridad del inicio de sesión pueden ser menos estrictos de lo que es habitual para z/OS.

Para habilitar la comprobación de seguridad, debe realizar lo siguiente:

- Instale y active un gestor de seguridad externa (por ejemplo, Security Server).
- Defina la clase MQADMIN si está utilizando un gestor de seguridad externa que no sea Security Server.
- Active la clase MQADMIN.

Debe considerar si la utilización de nombres de recurso que contengan una mezcla de mayúsculas y minúsculas podría ser beneficioso para su empresa. Si utiliza nombres de recurso de ese tipo en los perfiles del gestor de seguridad externa, debe definir y activar la clase MXADMIN.

## Cifrado de conjunto de datos de z/OS

El cifrado de conjunto de datos (DSE) ofrece la posibilidad de cifrar conjuntos de datos de z/OS, para que los datos que contienen solo puedan verlos o modificarlos los ID de usuario a los que se otorga el permiso específico. Esto permite el cifrado de datos en reposo en el sistema de archivos y evita la divulgación accidental de información confidencial a usuarios que tengan una necesidad empresarial legítima y permisos para gestionar los propios conjuntos de datos.

Antes de IBM MQ for z/OS 9.1.4, IBM MQ for z/OS no daba soporte al uso de DSE con los registros activos, los conjuntos de páginas y los conjuntos de datos de mensajes compartidos (SMDS) que proporcionan los mecanismos de persistencia primaria para los mensajes de IBM MQ.

En su lugar, [Advanced Message Security](#) proporciona una solución de cifrado de un extremo a otro para la mensajería de IBM MQ, que abarca toda la red IBM MQ, el cifrado de datos en curso, en reposo e incluso dentro de los procesos de tiempo de ejecución IBM MQ.

Otros conjuntos de datos secuenciales y VSAM utilizados en un subsistema IBM MQ se pueden cifrar utilizando DSE. Por ejemplo:

- Conjunto de datos de rutina de carga (BSDS)
- Los archivos secuenciales que contienen mandatos de configuración del sistema (MQSC) que se leen en el arranque utilizando CSQINPx DDNAME
- Los registros de archivado de IBM MQ, a menudo utilizados para el archivado a largo plazo de datos de registro de IBM MQ para la auditoría.

Puede realizar el cifrado mediante DSE asignando una clase de datos que se define con una etiqueta de clave de conjunto de datos. Para obtener más información, consulte [Planificación del almacenamiento de archivado de registro](#).

Desde IBM MQ for z/OS 9.1.4, IBM MQ for z/OS da soporte al uso de DSE con los registros activos y los conjuntos de páginas además del soporte proporcionado en releases anteriores.

IBM MQ for z/OS no da soporte al uso de DSE para conjuntos de datos de mensajes compartidos (SMDS).

Consulte la sección [Confidencialidad para los datos en reposo en IBM MQ for z/OS con cifrado de conjunto de datos](#), para obtener más información.

### Conceptos relacionados

[Conceptos de seguridad](#)

[Registros de autenticación de canal](#)

[Autorización para trabajar con objetos IBM MQ en z/OS](#)

[Protocolos de seguridad de cifrado: TLS](#)

### Tareas relacionadas

[Configuración de la seguridad en z/OS](#)

## Referencia relacionada

[Mensajes para IBM MQ for z/OS](#)

## Controles y opciones de seguridad en IBM MQ for z/OS

Puede especificar si la seguridad está activada en todo el subsistema IBM MQ y si desea realizar comprobaciones de seguridad en el gestor de colas o en el nivel de grupo de compartición de colas. También puede controlar el número de ID de usuario en los que se ha comprobado la seguridad de recursos de la API.

### Seguridad de subsistema

La seguridad de subsistema es un control que especifica si debe realizarse alguna comprobación de seguridad para todo el gestor de colas. Si no necesita comprobación de seguridad (por ejemplo, en un sistema de prueba) o si está satisfecho con el nivel de seguridad en todos los recursos que pueden conectarse a IBM MQ (incluidos clientes y canales), puede desactivar la comprobación de seguridad para el gestor de colas o el grupo de compartición de colas para que no se realice ninguna comprobación de seguridad adicional.

Se trata de la única comprobación que puede desactivar completamente la seguridad y determinar si deben realizarse otras comprobaciones de seguridad o no. Es decir, si desactiva la comprobación del gestor de colas o del grupo de compartición de colas, no se realiza ninguna otra comprobación de IBM MQ; si lo deja activado, IBM MQ comprueba los requisitos de seguridad para otros recursos de IBM MQ.

También puede activar o desactivar la seguridad para conjuntos de recursos concretos, como por ejemplo, mandatos.

### Comprobación del nivel de gestor de colas o de grupo de compartición de colas

Puede implementar la seguridad a nivel de gestor de colas o a nivel de grupo de compartición de colas. Si implementa la seguridad a nivel de grupo de compartición de colas, todos los gestores de colas del grupo compartirán los mismos perfiles. Esto significa que hay menos perfiles para definir y mantener, haciendo que la gestión de seguridad más fácil. También es más fácil añadir un nuevo gestor de colas al grupo de compartición de colas porque hereda los perfiles de seguridad existentes.

También es posible implementar una combinación de ambos si la instalación así lo requiere, por ejemplo, durante la migración o si tiene un gestor de colas en el grupo de compartición de colas que necesite distintos niveles de seguridad a los otros gestores de colas del grupo.

### Seguridad a nivel de grupo de compartición de colas

La comprobación de la seguridad de nivel de grupo de compartición de colas se realiza para todo el grupo de compartición de colas. Permite simplificar la administración de la seguridad porque requiere que se definan menos perfiles de seguridad. La autorización de un ID de usuario para utilizar un recurso determinado se maneja a nivel de grupo de compartición de colas, y es independiente de cuál sea el gestor de colas que utiliza dicho ID de usuario para acceder al recurso.

Por ejemplo, supongamos que una aplicación de servidor se ejecuta con el usuario ID SERVER, desea acceder a una cola denominada SERVER.REQUEST, y se desea ejecutar una instancia de SERVER en cada imagen de z/OS del sysplex. En lugar de permitir que SERVER abra SERVER.REQUEST en cada gestor de colas de forma individual (seguridad a nivel de gestor de colas), puede permitir únicamente el acceso a nivel de grupo de compartición de colas.

Puede utilizar perfiles de seguridad de nivel de grupo de compartición de colas para proteger todos los tipos de recursos, ya sean locales o compartidos.

### **Seguridad a nivel de gestor de colas**

Puede utilizar perfiles de seguridad de nivel de gestor de colas para proteger todos los tipos de recursos, ya sean locales o compartidos.

### **Combinación de ambos niveles**

Puede utilizar una combinación de la seguridad de nivel de grupo de compartición de colas y de seguridad de nivel de gestor de colas.

Puede alterar temporalmente valores de seguridad a nivel de grupo de compartición de colas para un gestor de colas específico que sea miembro de dicho grupo. Esto significa que puede realizar un nivel distinto de comprobaciones de seguridad en un gestor de colas individual que los realizados en otros gestores de colas del grupo.

Para más información, consulte [Perfiles para controlar la seguridad a nivel de gestor de compartición de colas o de gestor de colas](#).

## **Control del número de ID de usuario comprobados**

RESLEVEL es un perfil de servidor de seguridad que controla el número de ID de usuario seleccionado para la seguridad de recursos de IBM MQ. Normalmente, cuando un usuario intenta acceder a un recurso de IBM MQ, el servidor de seguridad comprueba el ID de usuario o los ID pertinentes para ver si se permite el acceso a dicho recurso. La definición de un perfil RESLEVEL permite controlar si se pueden comprobar cero, uno o, siempre que sea posible, dos identificadores de usuario.

Estos controles se realizan en una conexión por base de conexión, y el último de la vida de la conexión.

Sólo hay un perfil RESLEVEL para cada gestor de colas. El control se implementa mediante el acceso que el ID de usuario tiene para este perfil.

## **Clases de IBM MQ RACF en una combinación de mayúsculas y minúsculas o bien mayúsculas**

Ahora puede utilizar el soporte de perfiles de RACF en una combinación de mayúsculas y minúsculas, que permite utilizar nombres de recursos en mayúsculas y minúsculas y definir perfiles IBM MQ RACF para protegerlos.

Puede elegir entre:

- Seguir utilizando Clases de IBM MQ RACF únicamente en mayúsculas, como en los releases anteriores o bien
- Utilizar las nuevas clases de IBM MQ RACF en una combinación de mayúsculas y minúsculas.

Sin el uso de perfiles RACF en mayúsculas y minúsculas, todavía puede seguir utilizando nombres de recursos en mayúsculas y minúsculas en IBM MQ for z/OS; sin embargo, estos nombres de recursos sólo pueden estar protegidos por perfiles RACF genéricos en las clases IBM MQ en mayúsculas. Cuando se utiliza el soporte de perfiles IBM MQ RACF en mayúsculas y minúsculas, puede proporcionar un nivel de protección más detallado definiendo los perfiles IBM MQ RACF en las clases IBM MQ en mayúsculas y minúsculas.

## **Recursos que puede proteger en IBM MQ for z/OS**

Cuando se inicia un gestor de colas, o cuando se lo indica un mandato de operador, IBM MQ for z/OS determina los recursos que desea proteger.

Puede controlar qué comprobaciones de seguridad se realizan para cada gestor de colas. Por ejemplo, puede aplicar varias comprobaciones de seguridad para un gestor de colas de producción, pero ninguno para un gestor de colas de prueba.

## Seguridad de conexión

La comprobación de la seguridad de la conexión se lleva a cabo cuando un programa de aplicación intenta conectarse a un gestor de colas. Se realiza emitiendo una solicitud MQCONN o MQCONNX, o cuando el iniciador de canal, o CICS o el adaptador IMS emiten una solicitud de conexión.

Si está utilizando la seguridad a nivel de gestor de colas, puede desactivar el control de seguridad de conexión para un gestor de colas determinado. Sin embargo, si hace esto, cualquier usuario se podrá conectar a ese gestor de colas.

Para el adaptador CICS, sólo se utiliza el ID de usuario del espacio de direcciones de CICS para la comprobación de seguridad de conexión, no el ID de usuario de terminal CICS. Para el adaptador IMS, cuando las regiones de control o dependientes de IMS se conectan a IBM MQ, se comprueba el ID de usuario de espacio de direcciones de IMS. Para el iniciador de canal, se comprueba el ID de usuario utilizado por el espacio de direcciones del iniciador de canal.

Puede activar o desactivar la comprobación de seguridad de conexión a nivel de gestor de colas o a nivel de grupo de compartición de colas.

## Seguridad de mandatos

La comprobación de seguridad de mandatos se lleva a cabo cuando un usuario emite un mandato MQSC desde cualquiera de los orígenes descritos en [Orígenes desde los que puede emitir mandatos MQSC y PCF en IBM MQ for z/OS](#). Puede realizar una comprobación separada para el recurso especificado por el mandato, tal como se describe en [“Seguridad de recursos de mandatos”](#) en la página 283.

Si desactiva la comprobación de mandatos, no se comprueban los emisores de los mandatos para determinar si tienen autorización para emitir el mandato.

Si se emiten mandatos MQSC desde una consola, la consola debe tener el atributo de autorización de consola SYS para z/OS. Los mandatos que se emiten desde los conjuntos de datos CSQINP1 o CSQINP2, o emitidos internamente por el gestor de colas, están exentos de cualquier comprobación de seguridad, mientras que los emitidos para CSQINPX utilizan el ID de usuario del espacio de direcciones del iniciador de canal. Debe controlar quién está autorizado para actualizar estos conjuntos de datos mediante la protección normal de conjuntos de datos.

Puede activar o desactivar la comprobación de seguridad de mandatos a nivel del gestor de colas o en el grupo de compartición de colas.

## Seguridad de recursos de mandatos

Algunos mandatos MQSC, por ejemplo la definición de una cola local incluyen la manipulación de recursos de IBM MQ. Cuando la seguridad de recursos de mandatos está activa, cada vez que se emite un mandato que implica un recurso, IBM MQ comprueba si se permite al usuario cambiar la definición de dicho recurso.

Puede utilizar la seguridad de recursos de mandatos para facilitar la aplicación de convenios de denominación. Por ejemplo, un administrador de nóminas puede estar autorizado para suprimir y definir solamente las colas cuyo nombre comience por "PAYROLL". Si la seguridad de recursos de mandatos está inactiva, no se realizan comprobaciones de seguridad para el recurso que es manejado por el mandato. No se debe confundir la seguridad de recursos de mandatos con la seguridad de mandatos; las dos son independientes.

Desactivar la comprobación de seguridad de recursos de mandatos no afecta a la comprobación de recursos que se realiza específicamente para otros tipos de proceso en los que no intervienen mandatos.

Puede activar o desactivar la comprobación de seguridad de recursos de mandatos a nivel del gestor de colas o del grupo de compartición de colas.

## Consideraciones sobre la seguridad de canal

### Seguridad de canal

Cuando utiliza canales, las funciones de seguridad disponibles dependen de protocolo de comunicaciones que vaya a utilizar. Si utiliza TCP, el protocolo de comunicaciones no proporciona funciones de seguridad, aunque puede utilizar TLS. Si utiliza APPC, puede enviar, a través de la red, la información de ID de usuario desde el MCA emisor al MCA de destino con fines de verificación.

Para ambos protocolos, puede especificar los ID de usuario que desea comprobar con fines de seguridad, y cuántos. De nuevo, las opciones disponibles dependen de protocolo que utilice, de lo que especifique al definir el canal y de los valores RESLEVEL del iniciador de canal.

Para obtener más información sobre los tipos de seguridad de canal disponibles, consulte [Registros de autenticación de canal](#) y [Visión general de la salida de seguridad](#).

### Referencia relacionada

[“Seguridad de recursos de la API en IBM MQ for z/OS” en la página 284](#)

Los recursos se comprueban cuando una aplicación abre un objeto con una llamada MQOPEN o mqput1. El acceso necesario para abrir un objeto depende de las opciones de apertura que se hayan especificado al abrir la cola.

### Seguridad de recursos de la API en IBM MQ for z/OS

Los recursos se comprueban cuando una aplicación abre un objeto con una llamada MQOPEN o mqput1. El acceso necesario para abrir un objeto depende de las opciones de apertura que se hayan especificado al abrir la cola.

La seguridad de recursos de API está subdividida en las comprobaciones siguientes:

- [Cola](#)
- [Proceso](#)
- [Lista de nombres](#)
- [Usuario alternativo](#)
- [Contexto](#)

No se realiza ninguna comprobación al abrir el objeto de gestor de colas ni al acceder a objetos de clase de almacenamiento.

### Cola

La comprobación de seguridad de cola controla a quién se le permite abrir qué cola y con qué opciones pueden abrirla. Por ejemplo, un usuario puede tener permiso para abrir una cola denominada PAYROLL.INCREASE.SALARY para examinar los mensajes en la cola (utilizando la opción MQOO\_BROWSE), pero no para eliminar mensajes de la cola (utilizando una de las opciones MQOO\_INPUT\_\*). Si desactiva la comprobación de colas, cualquier usuario puede abrir cualquier cola con cualquier opción de apertura válida (es decir, cualquier opción MQOO\_\* válida en una llamada MQOPEN o MQPUT1).

Puede activar o desactivar la comprobación de seguridad de cola en el gestor de colas o en el grupo de compartición de colas.

### Proceso

La comprobación de seguridad del proceso se lleva a cabo cuando un usuario abre un objeto de definición de proceso. Si desactiva la comprobación de procesos, cualquier usuario puede abrir cualquier proceso.

Puede activar o desactivar la comprobación de seguridad de proceso a nivel del gestor de colas o del grupo de compartición de colas.

## Lista de nombres

La comprobación de seguridad de nombres se lleva a cabo cuando un usuario abre una lista de nombres. Si desactiva la comprobación de listas de nombre, cualquier usuario puede abrir cualquier lista de nombres.

Puede activar o desactivar la comprobación de seguridad de lista de nombres a nivel del gestor de colas o del grupo de compartición de colas.

## Usuario alternativo

La seguridad de usuario alternativo controla si un ID de usuario puede utilizar la autorización de otro ID de usuario para abrir un objeto IBM MQ.

Por ejemplo:

- Un programa de servidor que se está ejecutando bajo el ID de usuario PAYSERV recupera un mensaje de solicitud de una cola que el ID de usuario USER1 puso en la cola.
- Cuando el programa servidor obtiene el mensaje de solicitud, procesa la solicitud y vuelve a transferir la respuesta a la cola de respuestas especificada con el mensaje de solicitud.
- En lugar de utilizar su propio ID de usuario (PAYSERV) para autorizar la apertura de una cola de respuestas, el servidor puede especificar otro ID de usuario, en este caso, USER1. En este ejemplo, la seguridad de usuario alternativo controlaría si el ID de usuario PAYSERV dispone de permiso para especificar al ID de usuario USER1 como un ID de usuario alternativo al abrir la cola de respuesta.

El ID de usuario alternativo se especifica en el campo *AlternateUserId* del descriptor de objeto (MQOD).

Puede utilizar ID de usuario alternativos en cualquier objeto de IBM MQ, por ejemplo, procesos o listas de nombres. No afecta al ID de usuario utilizado por cualquier otro gestor de recursos; por ejemplo, para la seguridad de CICS o para la seguridad del conjunto de datos de z/OS.

Si la seguridad de usuario alternativo no está activa, cualquier usuario puede utilizar cualquier otro ID de usuario como un ID de usuario alternativo.

Puede activar o desactivar la comprobación de seguridad de usuario alternativo a nivel de gestor de colas o a nivel de grupo de compartición de colas.

## Contexto

El contexto es información que se puede aplicar a un mensaje particular y está incluido en el descriptor de mensaje (MQMD) que forma parte del mensaje. En la información de contexto hay dos secciones:

### Sección de identidad

El usuario de la aplicación que coloca primero el mensaje en una cola. Consiste en los siguientes campos:

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

### Sección de origen

La aplicación que pone el mensaje en la cola dónde actualmente está almacenado. Consiste en los siguientes campos:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Las aplicaciones pueden especificar los datos de contexto cuando se realiza un MQPUT o una MQPUT1 llamada. La aplicación puede generar los datos, los datos pueden pasarse desde otro mensaje, o el gestor de colas puede generar los datos de forma predeterminada. Por ejemplo, los programas servidor pueden utilizar los datos de contexto para comprobar la identidad del solicitante, es decir, ¿ese mensaje procedía de la aplicación correcta? Normalmente, el campo *UserIdentifier* se utiliza para determinar el ID de usuario de un usuario alternativo.

La seguridad de contexto se utiliza para controlar si el usuario puede especificar alguna de las opciones de contexto en cualquier llamada MQOPEN o MQPUT. Para obtener información sobre las opciones de contexto, consulte [Opciones MQOPEN relacionadas con el contexto de mensaje](#). Para obtener descripciones de los campos del descriptor de mensaje relacionados con el contexto, consulte [MQMD - Descriptor de mensaje MQMD - Descriptor de mensaje](#).

Si desactiva la comprobación de la seguridad de contexto, cualquier usuario podrá utilizar cualquiera de las opciones de contexto que permita la seguridad de cola.

Puede activar o desactivar la comprobación de seguridad de contexto a nivel de cola, gestor de colas o a nivel de grupo de compartición de colas.

## Disponibilidad en z/OS

IBM MQ for z/OS tiene muchas características de alta disponibilidad. En este tema se describen algunas de las consideraciones de disponibilidad.

Varias características de IBM MQ pueden aumentar la disponibilidad del sistema si el gestor de colas o el iniciador de canal falla. Para obtener más información acerca de estas características, consulte los apartados siguientes:

- [Consideraciones sobre sysplex](#)
- [Colas compartidas](#)
- [Canales compartidos](#)
- [Disponibilidad de red de IBM MQ](#)
- [Utilización del gestor de reinicio automático \(ARM\) de z/OS](#)
- [Utilización del recurso de recuperación ampliado \(XRF\) de z/OS](#)
- [Utilización del atributo GROUPUR de z/OS para la recuperación en un grupo de compartición de colas](#)
- [Dónde encontrar más información sobre la disponibilidad](#)

### Consideraciones sobre Sysplex

En un *sysplex*, un número de imágenes de sistemas operativos de z/OS colaboran en una única imagen del sistema y se comunican utilizando un recurso de acoplamiento. IBM MQ puede utilizar los recursos del entorno *sysplex* para mejorar la disponibilidad.

Eliminar las afinidades entre un gestor de colas y una imagen específica de z/OS permite que un gestor de colas se reinicie en una imagen de z/OS diferente en el caso de una anomalía de la imagen. El mecanismo de reinicio puede ser manual, utilizar ARM, o utilizar automatización del sistema, si se asegura de lo siguiente:

- Todos los conjuntos de páginas, registros, conjuntos de datos de rutina de carga, bibliotecas de código, y conjuntos de datos de configuración del gestor de colas se definen en volúmenes compartidos.
- La definición de subsistema tiene ámbito *sysplex* y un nombre exclusivo dentro del *sysplex*.
- El nivel de *código anticipado* instalado en cada imagen de z/OS durante la IPL está en el mismo nivel.
- Las direcciones IP virtuales TCP (VIPA) están disponibles en cada pila TCP del *sysplex*, y ha configurado escuchas TCP y conexiones de entrada de IBM MQ para utilizar VIPA en lugar de nombres de host predeterminados.

Para obtener más información sobre el uso de TCP en un sysplex, consulte *TCP/IP in a sysplex*, SG24-5235, una publicación de IBM Redbooks.

También puede configurar varios gestores de colas que se ejecutan en distintas imágenes de sistema operativo en un sysplex para que funcionen como un grupo de compartición de colas, y aprovechar las colas compartidas y los canales compartidos para una mayor disponibilidad y equilibrio de carga de trabajo.

## Colas compartidas

En el entorno de grupo de compartición de colas, una aplicación se puede conectar a cualquiera de los gestores de colas dentro del grupo de compartición de colas. Debido a que todos los gestores de colas del grupo de compartición de colas pueden acceder al mismo conjunto de colas compartidas, la aplicación no depende de la disponibilidad de un gestor de colas determinado; cualquier gestor de colas del grupo de compartición de colas puede prestar servicio a cualquier cola. Esto ofrece una mayor disponibilidad si un gestor de colas se detiene, ya que todos los demás gestores de colas del grupo de compartición de colas pueden seguir procesando la cola. Para obtener información sobre alta disponibilidad de los gestores de cola, consulte [“Ventajas de la utilización de colas compartidas”](#) en la página 192.

Para mejorar todavía más la disponibilidad de mensajes en un grupo de compartición de colas, IBM MQ detecta si otro gestor de colas del grupo se desconecta del recurso de acoplamiento de forma anómala y completa las unidades de trabajo de ese gestor de colas que todavía están pendientes, cuando ello sea posible. Esto se conoce como *recuperación de iguales* y se describe en [“Recuperación de iguales”](#) en la página 272.

La recuperación de iguales no puede recuperar unidades de trabajo que estaban pendientes en el momento de la anomalía. Puede utilizar el gestor de reinicio automático (ARM) para reiniciar todos los sistemas en la anomalía (por ejemplo, CICS, Db2 y IBM MQ), así como para asegurarse de que todos se reinician en el mismo nuevo procesador. Esto significa que pueden resincronizarse, y proporciona una recuperación rápida de las unidades de trabajo pendientes. Esto se describe en [“Utilización del Gestor de reinicio automático \(ARM\) de z/OS”](#) en la página 288.

## Canales compartidos

En el entorno de grupo de compartición de colas, IBM MQ proporciona funciones que proporcionan alta disponibilidad a la red. El iniciador de canal permite utilizar productos de red que equilibran las solicitudes de red entre un conjunto de servidores aptos y ocultan las anomalías del servidor de la red (por ejemplo, recursos genéricos VTAM). IBM MQ utiliza un puerto genérico para las solicitudes entrantes para que las solicitudes se puedan direccionar a cualquier iniciador de canal disponible en el grupo de compartición de colas. Esto se describe en [“Canales compartidos”](#) en la página 214.

Los canales de salida compartidos toman los mensajes que envían de una cola de transmisión compartida. La información sobre el estado de un canal compartido se almacena en un lugar para todo el nivel de grupo de compartición de colas. Esto significa que un canal se puede reiniciar automáticamente en un iniciador de canal diferente del grupo de compartición de colas si el iniciador de canal, el gestor de colas o el subsistema de comunicaciones fallan. Esto se denomina *recuperación de canales iguales* y se describe en [Canales de salida compartidos](#).

## Disponibilidad de red de IBM MQ

Los mensajes de IBM MQ se transportan desde el gestor de colas hasta el gestor de colas de una red IBM MQ utilizando canales. Puede cambiar la configuración en varios niveles para mejorar la disponibilidad de red de un gestor de colas y la capacidad de un canal IBM MQ para detectar un problema de red y volverse a conectar.

*Mantener activo* de TCP está disponible para canales TCP/IP. Hace que TCP envíe paquetes periódicamente entre sesiones para detectar anomalías en la red. El atributo de canal KAINTE determina la frecuencia de estos paquetes para un canal.

*AdoptMCA* permite que un canal, bloqueado en el proceso de recepción como resultado de una interrupción de la red, sea terminado y sustituido por una nueva solicitud de conexión. Puede controlar *AdoptMCA* utilizando la propiedad de gestor de colas *ADOPTMCA* con el programa de utilidad *MQSC* o la propiedad *AdoptNewMCAType* con la interfaz de Formatos de mandatos programables.

*ReceiveTimeout* impide que un canal sea bloqueado permanentemente en una llamada de recepción de red. Los parámetros de iniciador de canal *RCVTIME* y *RCVTMIN* determinan las características de tiempo de espera de recepción para los canales, como una función de su intervalo de pulsaciones. Consulte [Parámetro del gestor de cola](#) para obtener más detalles.

## Utilización del Gestor de reinicio automático (ARM) de z/OS

Puede utilizar IBM MQ for z/OS junto con el gestor de reinicio automático (ARM) de z/OS. Si un gestor de colas o un iniciador de canal ha fallado, el ARM lo reinicia en la misma imagen de z/OS. Si z/OS falla, también falla todo un grupo de subsistemas y aplicaciones relacionados. ARM puede reiniciar automáticamente todo los sistemas anómalos en un orden predefinido o bien, en otra imagen de z/OS dentro del sysplex. Esto se denomina reinicio de sistemas cruzados.

ARM permite una recuperación rápida de las transacciones pendientes en el entorno de colas compartidas. También proporciona una mayor disponibilidad si no se utilizan grupos de compartición de colas.

Puede utilizar el ARM para reiniciar un gestor de colas en una imagen de z/OS distinta dentro del sysplex en el caso de una anomalía de z/OS.

Para habilitar el reinicio automático, debe hacer lo siguiente:

1. Configurar un conjunto de datos de acoplamiento ARM.
2. Defina las acciones de reinicio automático que desea que realice z/OS en una *política de ARM*.
3. Inicie la política del ARM.

Si desea reiniciar automáticamente gestores de colas en diferentes imágenes de z/OS, cada gestor de colas de cada imagen de z/OS en el que se puede reiniciar dicho gestor de colas debe definirse en un nombre de subsistema de 4 caracteres exclusivo para todo el sysplex.

El uso de ARM con IBM MQ se describe en [Utilización de ARM en una red IBM MQ](#).

## Utilización del recurso de recuperación ampliado (XRF) de z/OS

Puede utilizar IBM MQ en un entorno de recurso de recuperación ampliado (XRF). Todos los conjuntos de datos propiedad de IBM MQ (código ejecutable, BSDS, registros y conjuntos de páginas) deben estar en DASD compartidos entre los procesadores XRF activos y alternativos.

Si utiliza XRF para la recuperación, debe detener el gestor de colas en el procesador activo e iniciarlo en el procesador alternativo. Para CICS, puede hacerlo utilizando la tabla de lista de mandatos (CLT) proporcionada por CICS o bien el operador del sistema puede hacerlo manualmente. Para IMS, es una operación manual y debe realizarla después de que el sistema IMS de coordinación haya completado el cambio de procesador.

Los programas de utilidad de IBM MQ se deben completar o terminar antes de que el gestor de colas pueda conmutarse al procesador alternativo. Analice detenidamente el efecto de esta posible interrupción cuando planifique los planes de recuperación de XRF.

Procure evitar que el gestor de colas se inicie en el procesador alternativo antes de que el gestor de colas en el procesador activo termine. Un inicio prematuro puede causar graves problemas de integridad en los datos, el catálogo y el registro. La utilización de la serialización de recursos globales (GRS) ayuda a evitar

los problemas de integridad al impedir el uso simultáneo de IBM MQ en los dos sistemas. Debe incluir el conjunto de datos de rutina de carga como un recurso protegido, y debe incluir los procesadores XRF activo y alternativo en el anillo GRS.

## Utilización del atributo GROUPUR de z/OS para la recuperación en un grupo de compartición de colas.

Los grupos de compartición de colas (QSG) permiten recursos transaccionales adicionales que se describen en este tema. El atributo GROUPUR permite a las aplicaciones cliente XA realizar cualquier recuperación de transacciones pendientes que pueda ser necesaria en cualquier miembro del QSG.

Si una aplicación de cliente XA se conecta a un grupo de compartición de colas (QSG) a través de un sysplex, no puede garantizar a qué gestor de colas específico se conecta. El uso del atributo GROUPUR por parte de los gestores de colas del grupo de compartición de colas puede permitir que cualquier recuperación de transacciones pendientes que pueda ser necesaria se produzca en cualquier miembro del QSG. Incluso si el gestor de colas al que la aplicación estaba conectada inicialmente no está disponible, la recuperación de transacciones puede llevarse a cabo.

Esta característica libera a la aplicación de cliente XA de cualquier dependencia de miembros específicos del QSG y, por lo tanto, amplía la disponibilidad del gestor de colas. El grupo de compartición de colas aparece en la aplicación transaccional como una única entidad que proporciona todas las características de IBM MQ y sin un único punto de error del gestor de colas.

Esta función no resulta aparente para la aplicación transaccional.

## Dónde encontrar más información sobre la disponibilidad

Puede encontrar más información sobre estos temas en las siguientes fuentes:

<i>Tabla 24. Dónde encontrar más información sobre la disponibilidad</i>	
<b>Tema</b>	<b>Dónde buscar</b>
Grupos de compartición de colas	<a href="#">“Colas compartidas y grupos de compartición de colas” en la página 171</a>
Parámetros del sistema	<a href="#">Configuración de parámetros del sistema</a>
Utilización del gestor de reinicio automático Programas de utilidad	<a href="#">Utilización de ARM en una red de IBM MQ</a>
Mandatos MQSC	<a href="#">Mandatos MQSC</a>

z/OS

## Supervisión y estadísticas en IBM MQ for z/OS

IBM MQ for z/OS tiene un conjunto de recursos para supervisar el gestor de colas y recopilar estadísticas.

IBM MQ proporciona recursos para supervisar el sistema y recopilar estadísticas. Si desea más información sobre estos recursos, consulte las secciones siguientes:

- [“Supervisión en línea” en la página 289](#)
- [“Rastreo de IBM MQ” en la página 290](#)
- [“Sucesos” en la página 290](#)

### Supervisión en línea

IBM MQ incluye los siguientes mandatos para supervisar el estado de los objetos IBM MQ:

- DISPLAY CHSTATUS muestra el estado de un canal especificado.
- DISPLAY QSTATUS muestra el estado de una cola especificada.
- DISPLAY CONN muestra el estado de una conexión especificada.

Para obtener más información sobre estos mandatos, consulte [Los mandatos MQSC](#).

## Rastreo de IBM MQ

IBM MQ suministra un recurso de rastreo que puede utilizar para recopilar la siguiente información mientras el gestor de colas está en ejecución:

### Estadísticas de rendimiento

El rastreo estadístico reúne la información siguiente para ayudarle a supervisar el rendimiento y ajustar el sistema:

- Recuentos de diferentes solicitudes MQI (estadísticas del gestor de mensajes)
- Recuentos de distintas solicitudes de objeto (estadísticas del gestor de datos)
- Información sobre el uso de Db2 (estadísticas sobre el gestor de Db2)
- Información sobre el uso del recurso de acoplamiento (estadísticas del gestor de recursos de acoplamiento)
- Información sobre el uso de SMDS (estadísticas del conjunto de datos de mensaje compartido)
- Información sobre el uso de la agrupación de almacenamiento intermedio (estadísticas del gestor de almacenamiento intermedio)
- Información sobre anotaciones cronológicas (estadísticas del gestor de registros)
- Información sobre el uso del almacenamiento (estadísticas del gestor de almacenamiento)
- Información sobre solicitudes de bloqueo (estadísticas del gestor de bloqueo)

### Datos de contabilidad

- El rastreo de contabilidad recopila información sobre el tiempo que el procesador emplea en procesar llamadas de la MQI y sobre el número de solicitudes MQPUT y MQGET realizadas por un usuario particular.
- IBM MQ también puede recopilar información sobre cada tarea utilizando IBM MQ. Estos datos se recopilan como un registro de contabilidad a nivel de hebra. Para cada hebra, IBM MQ también recopila información sobre cada cola utilizada por esa hebra.

Los datos generados por el rastreo se envían al Recurso de gestión del sistema (SMF) o al recurso de rastreo generalizado (GTF).

## Sucesos

Los sucesos de IBM MQ proporcionan información sobre errores, avisos y otras apariciones significativas en un gestor de colas. Al incorporar estos sucesos a su propia aplicación de gestión del sistema, puede supervisar las actividades en muchos gestores de colas, para varias aplicaciones de IBM MQ. En concreto, puede supervisar todos los gestores de colas en el sistema de un único gestor de colas.

Los sucesos se pueden notificar a través de un mecanismo de creación de informes escrito por el usuario a una aplicación de administración que soporte la presentación de los sucesos a un operador. Los sucesos también permiten que las aplicaciones actúen como agentes para otras redes de administración; por ejemplo, NetView, para supervisar informes y crear las alertas apropiadas.

### Tareas relacionadas

[Utilización del rastreo de IBM MQ](#)

[Utilización de sucesos de IBM MQ](#)

## Disposición de unidad de recuperación en z/OS

Determinadas aplicaciones transaccionales pueden utilizar una disposición de unidad de recuperación GROUP, en lugar de QMGR, al conectarse con un gestor de colas en un grupo de compartición de colas (QSG), especificando el nombre de QSG al conectarse en lugar del nombre del gestor de colas. Ello permite que la recuperación de transacciones sea más flexible y robusta eliminando el requisito de reconexión con el mismo gestor de colas en QSG.

Las transacciones iniciadas por las aplicaciones que se han conectado utilizando el nombre del grupo de compartición de colas también tienen una disposición de unidad de recuperación GROUP.

Cuando una aplicación transaccional se conecta con una disposición de unidad de recuperación GROUP, se conecta de forma lógica al grupo de compartición de colas y no mantiene ninguna afinidad con ningún gestor de colas específico. Las transacciones de confirmación en dos fases que ha iniciado y han completado la primera fase del proceso de confirmación, es decir, que están pendientes, pueden consultarse y resolverse, cuando están conectadas a cualquier gestor de colas en QSG. En un escenario de recuperación esto significa que el coordinador de transacciones no tiene que volver a conectarse con el mismo gestor de colas, que puede no estar disponible en ese momento.

Las aplicaciones que se conectan con una disposición de unidad de recuperación QMGR tienen una afinidad directa con el gestor de colas al que están conectadas. En un escenario de recuperación, el coordinador de transacciones debe volver a conectarse al mismo gestor de colas para resolver las transacciones pendientes, independientemente de si el gestor de colas pertenece a un grupo de compartición de colas.

Cuando las aplicaciones especifican un nombre de grupo de compartición de colas y se conectan por lo tanto a un gestor de colas de un QSG con una disposición de unidad de recuperación GROUP, el grupo de compartición de colas es lógicamente un gestor de recursos separado. Esto significa que las transacciones pendientes solo están visibles para una aplicación si se vuelve a conectar con la misma disposición de unidad de recuperación. Las transacciones pendientes con una disposición de unidad de recuperación QMGR no están visibles para las aplicaciones que se han conectado con una disposición de unidad de recuperación GROUP y viceversa.

### Conceptos relacionados

[“Habilitación de unidades de recuperación GROUP” en la página 291](#)

Un grupo de compartición de colas puede configurar y habilitar el soporte de unidades de recuperación GROUP.

[“Soporte de aplicación” en la página 292](#)

Utilice esta página para determinar qué aplicaciones se pueden conectar con una disposición de unidad de recuperación GROUP.

## Habilitación de unidades de recuperación GROUP

Un grupo de compartición de colas puede configurar y habilitar el soporte de unidades de recuperación GROUP.

Para utilizar unidades de recuperación GROUP en un gestor de colas en un QSG, habilite el atributo del gestor de colas GROUPUR. Para obtener más información sobre este concepto, consulte [“Disposición de unidad de recuperación en z/OS” en la página 291](#) antes de seguir leyendo este tema.

Cuando se habilita el atributo del gestor de colas GROUPUR, el gestor de colas acepta conexiones nuevas con una disposición de unidad de recuperación GROUP. Si inhabilita este atributo, no se aceptan las conexiones nuevas con esta disposición, aunque las aplicaciones ya conectadas no se ven afectadas hasta que se desconectan.

Cuando una aplicación se conecta con una disposición de unidad de recuperación GROUP y consulta qué transacciones están pendientes o intenta resolver una transacción iniciada en alguna otra ubicación del grupo de compartición de colas, el gestor de colas al que se ha conectado ahora debe poder comunicarse con los otros miembros del grupo de compartición de colas para poder procesar la solicitud. Para hacerlo, utiliza una cola compartida denominada SYSTEM.QSG.UR.RESOLUTION.QUEUE. Esta cola debe encontrarse en una estructura de aplicaciones recuperable denominada CSQSYSAPPL. La estructura debe

ser recuperable porque los mensajes permanentes se almacenan en esta cola cal procesar solicitudes de resolución.

Antes de habilitar las unidades de recuperación GROUP, debe asegurarse de que se hayan definido la estructura de recurso de acoplamiento y la cola compartida. Puede utilizar las definiciones del ejemplo CSQ4INSS. Cuando se define la cola o se detecta durante el inicio, cada gestor de colas del grupo de compartición de colas abre la cola de forma para poder recibir solicitudes de entrada. Si desea suprimir o mover la cola porque se ha definido incorrectamente, puede solicitar que los gestores de colas cierren los manejadores que estén abiertos en la cola, actualizando el objeto de cola para inhibir las solicitudes de MQGET. Una vez realizadas las correcciones necesarias, si se permite que las aplicaciones obtengan mensajes de la cola una vez más, se indica a cada gestor de colas que vuelva a abrirla. Utilice el mandato DISPLAY QSTATUS para identificar qué manejadores están abiertos en una cola.

Cuando haya completado esta configuración podrá habilitar unidades de recuperación GROUP en cada gestor de colas en el que desee que las aplicaciones transaccionales puedan conectarse con una disposición de unidad de recuperación GROUP. No es necesario que todos los gestores de colas estén en el grupo de compartición de colas, pero si decide habilitar esta función solo en un subconjunto del grupo de compartición de colas, debe asegurarse de que sus aplicaciones solo intenten conectarse con los gestores de colas en los que está habilitada dicha función. Para obtener más información, consulte [“Soporte de aplicación”](#) en la página 292.

Cuando intente habilitar el atributo del gestor de colas GROUPUR, se realizan numerosas comprobaciones de configuración. El gestor de colas comprueba que:

- Pertenzca a un grupo de compartición de colas.
- Se haya definido la cola compartida denominada SYSTEM.QSG.UR.RESOLUTION.QUEUE de acuerdo con la definición en CSQ4INSS.
- La cola compartida SYSTEM.QSG.UR.RESOLUTION.QUEUE esté en una estructura de recurso de acoplamiento recuperable denominada CSQSYSAPPL.

Si falla alguna de las comprobaciones anteriores, el atributo GROUPUR permanece inhabilitado y devuelve el código de mensaje.

Estas comprobaciones de configuración también se realizan durante el inicio del gestor de colas si se ha habilitado el atributo del gestor de colas. Si alguna de las comprobaciones falla durante el inicio, se inhabilitan las unidades de recuperación GROUP y el gestor de colas emite un mensaje que indica la comprobación que ha fallado. Cuando haya realizado la acción correctiva necesaria, deberá volver a habilitar el atributo del gestor de colas.

## Soporte de aplicación

Utilice esta página para determinar qué aplicaciones se pueden conectar con una disposición de unidad de recuperación GROUP.

El soporte para la disposición de la unidad de recuperación GROUP está limitado a determinados tipos de aplicaciones transaccionales para las que IBM MQ for z/OS es un gestor de recursos, pero no el coordinador de transacciones. Las aplicaciones transaccionales soportadas actualmente son:

- Aplicaciones cliente transaccional ampliadas de IBM MQ
- Aplicaciones IBM MQ classes for JMS que se ejecutan en un servidor de aplicaciones, como por ejemplo WebSphere Application Server.
- CICS Aplicaciones que se ejecutan en CICS Transaction Server 4.2 o posterior, cuando la definición de recurso CICS MQCONN se configura con RESYNCMEMBER (GROUPRESYNC).

### Conceptos relacionados

[“Aplicaciones cliente transaccional ampliadas de IBM MQ”](#) en la página 293

Utilice esta página para determinar cómo las aplicaciones cliente transaccional ampliadas de IBM MQ pueden utilizar la disposición de la unidad GROUP de recuperación.

[“Aplicaciones de CICS”](#) en la página 293

Utilice esta página para determinar cómo CICS puede utilizar la disposición GROUP de la unidad de recuperación.

### **Aplicaciones cliente transaccional ampliadas de IBM MQ**

Utilice esta página para determinar cómo las aplicaciones cliente transaccional ampliadas de IBM MQ pueden utilizar la disposición de la unidad GROUP de recuperación.

Un ejemplo de aplicación cliente transaccional ampliada de IBM MQ es aquella que utiliza JMS y se ejecuta en WebSphere Application Server, conectándose a IBM MQ a través de TCP/IP, en lugar de enlaces locales. Estas aplicaciones cliente se conectan a IBM MQ for z/OS a través de conexiones de red, como por ejemplo a través de TCP/IP. Para estas aplicaciones, es el valor especificado para el parámetro QMNAME de la serie xa\_info que se pasa en la llamada xa\_open que especifica si se utiliza una disposición de unidad de recuperación QMGR o GROUP. Para obtener más información sobre xa\_open, consulte [Formato de una serie xa\\_open](#) y [Proceso de errores adicional para xa\\_open](#). Para las aplicaciones JMS esto se consigue especificando el nombre del grupo de compartición de colas (QSG) en ConnectionFactory en lugar del nombre de un gestor de colas específico.

Para que las aplicaciones cliente XA puedan beneficiarse de la disposición de la unidad de recuperación GROUP, debe establecer la configuración del TCP/IP de forma que las aplicaciones cliente puedan direccionarse a los gestores de colas del grupo de compartición de colas que tiene habilitado el atributo GROUPUR, y no a un gestor de colas específico. Una de las tecnologías de dirección IP virtual que puede utilizar para ello es el distribuidor Sysplex de z/OS. Para obtener más detalles, consulte [z/OS Servidor de comunicaciones](#) y [z/OS Habilidades básicas: direccionamiento virtual dinámico](#). Si desea habilitar unidades de recuperación GROUP en un subconjunto de los gestores de colas de un grupo de compartición de colas, asegúrese de que las aplicaciones cliente no se puedan direccionar a aquellos en los que no está habilitado.

Las aplicaciones cliente no deben conectarse al grupo de compartición de colas utilizando canales compartidos.

### **Aplicaciones de CICS**

Utilice esta página para determinar cómo CICS puede utilizar la disposición GROUP de la unidad de recuperación.

CICS 4.2 y posteriores proporciona la opción de resincronización de grupo, RESYNCMEMBER (GROUPRESYNC) en una definición de recurso MQCONN. CICS configurado con esta opción se puede conectar a cualquier gestor de colas adecuado en un grupo de compartición de colas que se ejecuta en el mismo LPAR que la región CICS. Para dar soporte a la opción GROUPRESYNC de CICS, un gestor de colas debe estar en ejecución en MQ V7.1 o una versión posterior y debe estar habilitado para el soporte de GROUPUR.

Las transacciones que se ejecutan dentro de una región de CICS conectada a MQ utilizando GROUPRESYNC crean unidades de trabajo con una disposición de unidad de recuperación GROUP.

Puede utilizar RESYNCMEMBER (GROUPRESYNC) para habilitar una recuperación más rápida después de una anomalía del gestor de colas, ya que permite a la región CICS conectarse inmediatamente a un gestor de colas elegible alternativo que se ejecuta en la misma LPAR, resolviendo las transacciones dudosas según sea necesario, sin esperar a que se reinicie el gestor de colas.

RESYNCMEMBER (GROUPRESYNC) también habilita opciones de reinicio más flexibles para CICS. Una región de CICS con la conexión MQ configurada para utilizar GROUPRESYNC y colas compartidas MQ se puede reiniciar en cualquier LPAR en el que haya un gestor de colas ejecutándose como un miembro del mismo grupo de compartición de colas.

## **IBM MQ y otros productos de z/OS**

Utilice este tema para comprender cómo IBM MQ puede funcionar con otros productos de z/OS.

### **Conceptos relacionados**

[“IBM MQ y CICS” en la página 294](#)

Todas las versiones de CICS soportadas por IBM MQ 9.0.0 y posteriores, utilizan la versión suministrada por CICS del adaptador y el puente.

[“IBM MQ for z/OS y WebSphere Application Server” en la página 300](#)

Utilice este tema para comprender el uso de IBM MQ for z/OS por parte de WebSphere Application Server.

#### **Referencia relacionada**

[“IBM MQ y IMS” en la página 295](#)

Utilice este tema para comprender cómo funciona IBM MQ con IMS. El adaptador de IMS permite conectar el gestor de colas con IMS y permite que las aplicaciones IMS utilicen la MQI.

[“IBM MQ y los adaptadores por lotes, TSO y RRS de z/OS” en la página 299](#)

Utilice este tema para comprender cómo funciona IBM MQ con los adaptadores por lotes, TSO y RRS de z/OS.

## **z/OS IBM MQ y CICS**

Todas las versiones de CICS soportadas por IBM MQ 9.0.0 y posteriores, utilizan la versión suministrada por CICS del adaptador y el puente.

Para obtener más información sobre cómo configurar el adaptador IBM MQ CICS y los componentes de IBM MQ CICS bridge , consulte la sección [Configuración de conexiones con IBM MQ](#) de la documentación de CICS .

#### **Tareas relacionadas**

[Utilización de IBM MQ con CICS](#)

## **z/OS Conexión del grupo CICS**

La conexión de grupo CICS ofrece la posibilidad de que una región de CICS se conecte a cualquier miembro activo de un grupo de compartición de colas de IBM MQ en el mismo LPAR en lugar de especificar un gestor de colas individual. CICS sigue conectándose a un único gestor de colas individual a la vez.

Necesita al menos dos gestores de colas en LPAR para ofrecer soporte a la conexión de grupo CICS. Utilizar una conexión de grupo ofrece mayor disponibilidad ya que no es necesario que un gestor de colas determinado esté activo. CICS se conecta a cualquier gestor de colas en el grupo de compartición de colas de LPAR.

Para obtener más información, consulte la documentación CICS en el recurso MQCONN.

CICS intenta conectarse a MQNAME transferido como si fuese un gestor de colas:

- Si el gestor de colas existe y está activo, la conexión seguirá funcionando.
- Si la conexión no se realiza correctamente, CICS consulta el estado de los gestores de colas del grupo para determinar que están activos en el mismo LPAR.
- Si hay varios gestores de colas activos, CICS comprueba RESYNCMEMBER (YES) y el estado de UOW para determinar si CICS necesita conectarse, o debe conectarse, a un miembro determinado, o esperar si no está activo.
- Si no es necesario conectarse a un miembro determinado, CICS selecciona un gestor de colas (utilizando un algoritmo de generación aleatoria).
- CICS intenta conectarse al gestor de colas seleccionado.
- Si el intento no se realiza correctamente, a continuación, dependiendo del código de retorno, CICS elige el siguiente miembro y, a continuación, recorre de nuevo por el bucle de selección.
- Si los gestores de colas no están activos, CICS emite varias conexiones a la lista de gestores de colas y espera en ECBLIST hasta que el primer gestor de colas esté disponible.

#### **Conceptos relacionados**

[“Recuperación de unidades de grupo \(GROUPUR\) para CICS” en la página 295](#)

IBM MQ GROUPUR for CICS proporciona recuperación de igual para unidades de trabajo dudosas en un grupo de compartición de colas (QSG). Un gestor de colas de IBM MQ puede resolver unidades de trabajo dudosas en nombre de otro gestor de colas en el grupo de compartición de colas. Esto significa que si CICS se vuelve a conectar a través de la conexión de grupo a un gestor de colas diferente en el QSG, puede resolver las transacciones dudosas desde una conexión IBM MQ anterior.

### **Información relacionada**

[Soporte para grupos de compartición de colas de IBM MQ](#)

## **z/OS Recuperación de unidades de grupo (GROUPUR) para CICS**

IBM MQ GROUPUR for CICS proporciona recuperación de igual para unidades de trabajo dudosas en un grupo de compartición de colas (QSG). Un gestor de colas de IBM MQ puede resolver unidades de trabajo dudosas en nombre de otro gestor de colas en el grupo de compartición de colas. Esto significa que si CICS se vuelve a conectar a través de la conexión de grupo a un gestor de colas diferente en el QSG, puede resolver las transacciones dudosas desde una conexión IBM MQ anterior.

Si una región de CICS trabaja con un gestor de colas y el gestor de colas finaliza de forma anómala, las transacciones pendientes se recuperan. Esto elimina la necesidad de que la región CICS espere a que se reinicie el gestor de colas con el que estaba trabajando y a continuación, resuelva las unidades de trabajo pendientes. Esto significa que necesita al menos dos gestores de colas en LPAR, de modo que CICS se puede conectar a otro gestor de colas en el caso de una terminación anómala del primer gestor de colas.

El nuevo valor RESYNCMEMBER (GROUPRESYNC) en la definición MQCONN de CICS :

- Utiliza la función de conexión de grupo de IBM MQ y la recuperación de igual.
- Requiere un gestor de colas con el atributo GROUPUR habilitado.
- Todavía da soporte a los valores de CICS MQCONN RESYNCMEMBER existentes (YES y NO):
  - Utiliza la función existente de conexión de grupo de CICS y ninguna recuperación de iguales.
  - El cambio de los valores RESYNCMEMBER entra en vigor la próxima vez que CICS se conecte a IBM MQ

### **Conceptos relacionados**

[“Habilitación de unidades de recuperación GROUP” en la página 291](#)

Un grupo de compartición de colas puede configurar y habilitar el soporte de unidades de recuperación GROUP.

## **z/OS IBM MQ y IMS**

Utilice este tema para comprender cómo funciona IBM MQ con IMS. El adaptador de IMS permite conectar el gestor de colas con IMS y permite que las aplicaciones IMS utilicen la MQI.

El puente IBM MQ - IMS opcional adicional permite que las aplicaciones ejecuten una aplicación IMS que no utilice la MQI. Esto significa que puede utilizar las aplicaciones existentes con IBM MQ, sin necesidad de reescribirlas.

Para obtener más información sobre estos componentes, consulte los subtemas siguientes:

### **Conceptos relacionados**

[IMS y las aplicaciones puente IMS en IBM MQ for z/OS](#)

### **Tareas relacionadas**

[Configuración del adaptador IMS](#)

[Configuración del puente IMS](#)

[Funcionamiento del adaptador IMS](#)

### **Referencia relacionada**

[MQIIH – Cabecera información de IMS](#)

El adaptador IMS es una interfaz entre los programas de aplicación de IMS y un subsistema IBM MQ.

Los adaptadores IBM MQ permiten que diferentes entornos de aplicaciones envíen y reciban mensajes a través de una red de colas de mensajes. El adaptador IMS es la interfaz entre los programas de aplicación de IMS y un subsistema IBM MQ. Hace posible que los programas de aplicación IMS utilicen la MQI.

El adaptador IMS recibe e interpreta las solicitudes de acceso a IBM MQ utilizando el Recurso de conexión de subsistema externo (ESAF) proporcionado por IMS. Normalmente, IMS se conecta a IBM MQ automáticamente sin intervención del operador.

El adaptador IMS proporciona acceso a recursos de IBM MQ para programas que se ejecutan en las siguientes modalidades o estados:

- Modalidad (TCB) de tarea
- Estado del problema
- Modalidad de memoria no cruzada
- Modalidad de registro sin acceso

El adaptador proporciona una hebra de conexión desde un bloque de control de tareas de aplicación (TCB) a IBM MQ.

El adaptador da soporte a un protocolo de confirmación en dos fases para los cambios realizados en los recursos que son de propiedad de IBM MQ con IMS actuando como coordinador del punto de sincronización. Las conversaciones en las que IMS no es el coordinador del punto de sincronización; por ejemplo, las conversaciones protegidas por APPC (SYNCLVL=SYNCPT) no están soportadas por el adaptador IMS.

El adaptador también proporciona una transacción de supervisor desencadenante (CSQQTRMN). Esto se describe en “El supervisor desencadenante de IMS” en la página 297.

Puede utilizar IBM MQ con IMS Extended Recovery Facility (XRF) para ayudar a la recuperación de un error de IMS.

**Nota:** A partir de IMS 15.2 Extended Recovery Facility (XRF) ya no está soportado. Consulte la documentación de IMS para obtener más información.

## Utilización del adaptador

Los programas de aplicación y el adaptador IMS se ejecutan en el mismo espacio de direcciones. El gestor de colas está separado, en su propio espacio de direcciones.

Debe editar el enlace de cada programa que emite una o más llamadas MQI a un módulo de interfaz de lenguaje IMS adecuado y, a menos que utilice llamadas MQI dinámicas, el programa de apéndice de API suministrado por IBM MQ, CSQQSTUB. Cuando la aplicación emite una llamada MQI, el apéndice transfiere el control al adaptador a través de la interfaz de subsistema externo de IMS, que gestiona el proceso de la solicitud por el gestor de colas de mensajes.

## Administración del sistema y funcionamiento con IMS

Un operador del terminal IMS autorizado puede emitir mandatos de IMS para controlar y supervisar la conexión con IBM MQ. Sin embargo, el operador del terminal IMS no tiene control sobre el espacio de direcciones de IBM MQ. Por ejemplo, el operador no puede concluir IBM MQ desde un espacio de direcciones de IMS.

## Restricciones

Las siguientes llamadas de API de IBM MQ no están soportadas en una aplicación que utiliza el adaptador IMS:

- MQCB
- MQCB\_FUNCTION
- MQCTL

## El supervisor desencadenante de IMS

El supervisor desencadenante de IMS ( **CSQQTRMN** ) es una aplicación IBM MQ proporcionada IMS que inicia una transacción IMS cuando se produce un suceso IBM MQ , por ejemplo, cuando se coloca un mensaje en una cola específica.

### Cómo funciona

Cuando se coloca un mensaje en una cola de mensajes de aplicación, se genera un desencadenante si se cumplen las condiciones desencadenantes. A continuación, el gestor de colas escribe un mensaje (que contiene datos definidos por el usuario), conocido como *mensaje de desencadenante*, en la cola de inicio que se ha especificado para dicha cola de mensajes. En un entorno IMS, puede iniciar una instancia de CSQQTRMN para supervisar una cola de iniciación y recuperar de ella los mensajes de desencadenante a medida que van llegando. Normalmente, CSQQTRMN planifica otra transacción IMS con un INSERT (ISRT) en la cola de mensajes de IMS. La aplicación de IMS iniciada lee el mensaje de la cola de mensajes de la aplicación y, a continuación, lo procesa. CSQQTRMN debe ejecutarse como una persistencia gestionada por bean sin mensaje.

Cada copia de CSQQTRMN sirve a una única cola de inicio. Cuando se ha iniciado, el supervisor desencadenante se ejecuta hasta que finaliza IBM MQ o IMS.

La macro APPLCTN para CSQQTRMN debe especificar SCHDTYP=PARALLEL.

Puesto que el supervisor desencadenante es una BMP orientada a lotes, las transacciones IMS que el supervisor desencadenante ha iniciado contienen lo siguiente:

- Espacios en blanco en el campo LTERM de IOPCB
- El nombre del PSB de la BMP del supervisor desencadenante en el campo ID de usuario del bloque de comunicaciones de programa de entrada/salida (IOPCB)

Si la transacción IMS de destino está protegida por el servidor de seguridad (conocido previamente como RACF), puede que sea necesario definir CSQQTRMN como un ID de usuario en el servidor de seguridad.

## El puente IBM MQ - IMS

El puente IBM MQ - IMS es el componente de IBM MQ for z/OS que permite el acceso directo desde aplicaciones IBM MQ a aplicaciones en el sistema IMS.

El puente IBM MQ - IMS habilita el *soporte MQI implícito*. Esto significa que puede volver a diseñar aplicaciones heredadas controladas por terminales conectados a 3270 para que sean controladas por mensajes de IBM MQ, sin tener que reescribirlas, volverlas a compilar o volverlas a enlazar. El puente es un cliente IMS *OTMA* (Gestor de transacción abierto).

En las aplicaciones puente no hay llamadas de IBM MQ dentro de la aplicación de IMS. La aplicación obtiene su entrada utilizando una llamada GET UNIQUE (GU) a la IOPCB y envía su salida mediante una llamada ISRT a la IOPCB. Las aplicaciones IBM MQ utilizan la cabecera de IMS (la estructura MQIIH) en los datos del mensaje para garantizar que las aplicaciones se pueden ejecutar tal como lo hacían cuando estaban controladas por terminales no programables. Si está utilizando una aplicación IMS que procesa mensajes de varios segmentos, tenga en cuenta que todos los segmentos deben estar contenidos en un mensaje IBM MQ.

El puente IMS se muestra en la [Figura 78](#) en la [página 298](#).

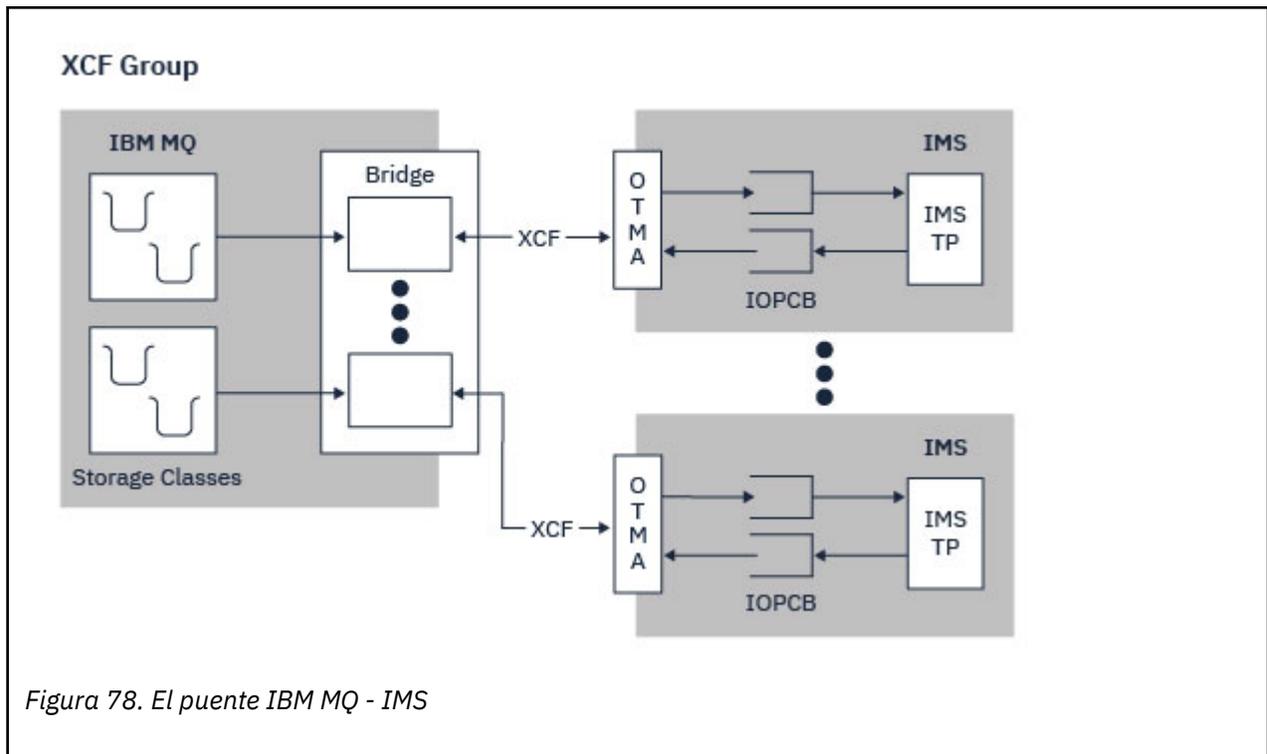


Figura 78. El puente IBM MQ - IMS

Un gestor de colas se puede conectar a uno o varios sistemas IMS y más de un gestor de colas se puede conectar a un sistema IMS. La única restricción es que todos ellos deben pertenecer al mismo grupo XCF y todos deben estar en el mismo sysplex.

Consulte [Configuración del puente IMS](#) para obtener información sobre cómo configurar un puente IMS y añadir una conexión IMS adicional al mismo gestor de colas.

## ¿Qué es OTMA?

El recurso OTMA de IMS es un protocolo de cliente/servidor sin conexión basado en transacciones que se ejecuta en IMS. Funciona como una interfaz para servidores de comunicaciones basados en host que acceden a aplicaciones IMS TM a través del [recurso de acoplamiento de sistemas cruzados \(XCF\) dez/OS](#).

OTMA permite que los clientes se conecten a IMS para ofrecer un alto rendimiento para las interacciones entre clientes e IMS para una red grande o un gran número de sesiones. OTMA se implementa en un entorno sysplex de z/OS. Por lo tanto, el dominio de OTMA está restringido al dominio de XCF.

## Supervisión del rendimiento de OTMA

El soporte para los mensajes de protocolo OTMA x '3C', disponibles en IMS v10 o superior, se incluye en el puente IBM MQ - IMS en IBM MQ for z/OS. IMS envía estos mensajes a los clientes OTM para informar sobre su estado de salud.

Si un socio de IMS no puede procesar el volumen de solicitudes de transacción que se envían, notificará a IBM MQ que se ha producido un aviso de desbordamiento. En respuesta, IBM MQ ralentizará la velocidad a la que se envían las solicitudes a través del puente.

Si IMS sigue sin poder procesar las solicitudes de transacciones y se produce una condición de desbordamiento completo, se suspenderán todas las TPIPE en el socio IMS. Tras la notificación del socio de IMS de que la condición de desbordamiento o aviso de desbordamiento se ha aliviado, IBM MQ reanudará todos los TPIPE suspendidos, si procede, y aumentará gradualmente la velocidad a la que se envían las solicitudes de transacción hasta alcanzar la velocidad máxima. IBM MQ emite mensajes de consola en respuesta a un cambio en el estado de los socios de IMS.

Si utiliza socios de IMS v10, deberá asegurarse de que se haya aplicado el PTF UK45082.

## Envío de transacciones de IMS desde IBM MQ

Para enviar una transacción IMS que utiliza el puente, las aplicaciones colocan mensajes en una cola de IBM MQ, como de costumbre. Los mensajes contienen datos de transacción de IMS; pueden tener una cabecera IMS (la estructura MQIIH) o permitir que el puente IBM MQ - IMS realice suposiciones sobre los datos del mensaje.

A continuación, IBM MQ transfiere el mensaje a una cola de IMS (primero se pone en cola en IBM MQ para habilitar el uso de puntos de sincronización para garantizar la integridad de los datos). La clase de almacenamiento de la cola de IBM MQ determina si la cola es una *cola OTMA* (es decir, una cola utilizada para transmitir mensajes al puente IBM MQ - IMS) y el socio IMS específico al que se envían los datos del mensaje.

Los gestores de colas remotos también pueden iniciar transacciones de IMS escribiendo en estas colas OTMA en IBM MQ for z/OS.

Los datos devueltos por el sistema IMS se graban directamente en la cola de respuesta de IBM MQ especificada en la estructura del descriptor de mensaje (MQMD). (Esta puede ser una cola de transmisión para el gestor de colas especificado en el campo **ReplyToQMGR** del MQMD.)

### Conceptos relacionados

[IMS y las aplicaciones puente IMS en IBM MQ for z/OS](#)

### Tareas relacionadas

[Personalizar el puente IMS](#)

### Referencia relacionada

“IBM MQ y IMS” en la página 295

Utilice este tema para comprender cómo funciona IBM MQ con IMS. El adaptador de IMS permite conectar el gestor de colas con IMS y permite que las aplicaciones IMS utilicen la MQI.

z/OS

## IBM MQ y los adaptadores por lotes, TSO y RRS de z/OS

Utilice este tema para comprender cómo funciona IBM MQ con los adaptadores por lotes, TSO y RRS de z/OS.

### Introducción a los adaptadores por lotes

Los adaptadores por lotes/TSO son la interfaz entre IBM MQ y los programas de aplicación z/OS que se ejecutan bajo JES, TSO o z/OS UNIX System Services. Estos adaptadores permiten a los programas de aplicación de z/OS utilizar la MQI.

Los adaptadores proporcionan acceso a los recursos de IBM MQ para los programas que se ejecutan en las siguientes modalidades o estados:

- Modalidad (TCB) de tarea
- Problema o estado de supervisor
- Modalidad de memoria no cruzada
- Modalidad de registro sin acceso

Las conexiones entre programas de aplicación y IBM MQ están en el nivel de tarea. Los adaptadores proporcionan una hebra de conexión desde un bloque de control de tareas de aplicación (TCB) hasta IBM MQ.

El adaptador por lotes/TSO da soporte a un protocolo de confirmación en una sola fase para los cambios realizados en los recursos que son propiedad de IBM MQ. No admite protocolos de confirmación de varias fases. El adaptador RRS permite a las aplicaciones IBM MQ participar en protocolos de confirmación de dos fases con otros productos habilitados para RRS, coordinados por los z/OS Servicios de recuperación de recursos (RRS).

Los adaptadores utilizan el servicio STIMERM de z/OS para planificar un suceso asíncrono cada segundo. Este suceso ejecuta un bloque de solicitud de interrupción (IRB) que no implica ninguna espera para la

tarea de la aplicación por lotes. Este IRB comprueba para ver si se ha publicado el BCE de terminación de IBM MQ. Si se ha publicado el BCE de terminación, el IRB publica todos los ECB de aplicación que están esperando un suceso en IBM MQ (por ejemplo, una señal o una espera).

## El adaptador por lotes/TSO

El adaptador IBM MQ por lotes/TSO proporciona soporte de IBM MQ para aplicaciones de proceso por lotes y TSO de z/OS. Todos los programas de aplicación que se ejecutan por lotes o TSO de z/OS deben tener el programa apéndice de la interfaz de programación de aplicaciones (API), CSQBSTUB editado con enlaces con ellos. El apéndice proporciona la aplicación con acceso a todas las llamadas de la MQI. Utilice una confirmación en una sola fase y la restitución para aplicaciones emitiendo las llamadas de la MQI **MQCMIT** y **MQBACK**.

## El adaptador RRS

Los [Servicios de recuperación de recursos \(RRS\)](#) es un subcomponente de z/OS que proporciona un servicio de todo el sistema para coordinar la confirmación en dos fases a través de productos de z/OS. El adaptador RRS de IBM MQ por lotes/TSO (el adaptador RRS) proporciona soporte de IBM MQ para aplicaciones de proceso por lotes y TSO de z/OS que desean utilizar estos servicios. El adaptador RRS permite a IBM MQ convertirse en un participante completo en la coordinación de RRS. Las aplicaciones pueden participar en el proceso de confirmación de dos fases con otros productos que dan soporte a RRS (por ejemplo, Db2).

El adaptador RRS proporciona dos apéndices; debe enlazar/editar aquellos programas de aplicación que quieran utilizar RRS con uno de estos apéndices.

### CSQBRSTB

Este apéndice permite utilizar la confirmación en dos fases y la restitución de aplicaciones mediante servicios de recuperación de recursos invocables RRS en lugar de las llamadas de la MQI **MQCMIT** y **MQBACK**.

También debe enlazar/editar el módulo ATRSCSS desde la biblioteca SYS1.CSSLIB de la aplicación. Si utiliza las llamadas de la MQI **MQCMIT** y **MQBACK**, recibirá el código de retorno MQRC\_ENVIRONMENT\_ERROR.

### CSQBRSI

Este apéndice permite utilizar llamadas MQI **MQCMIT** y **MQBACK**; IBM MQ en realidad implementa estas llamadas como llamadas RRS de **SRRCMIT** y **SRRBACK**.

Para obtener información sobre la creación de programas de aplicación que utilicen el adaptador RRS, consulte [El adaptador RSS por lotes](#).

## Dónde encontrar información sobre los adaptadores por lotes, TSO y RRS de z/OS

Puede obtener más información sobre los temas de esta sección en las fuentes siguientes:

<i>Tabla 25. Dónde encontrar más información sobre la utilización de z/OS por lotes con IBM MQ</i>	
Tema	Dónde buscar
Configuración de adaptadores por lotes	<a href="#">Tarea 19: Configurar adaptadores de lotes, TSO y RRS</a>
Servicios de recuperación de recursos invocables RRS	<a href="#">MVS Programming: Callable Services for High Level Languages</a>

## IBM MQ for z/OS y WebSphere Application Server

Utilice este tema para comprender el uso de IBM MQ for z/OS por parte de WebSphere Application Server.

Las aplicaciones escritas en Java que se ejecutan bajo WebSphere Application Server pueden utilizar la especificación Java Message Service (JMS) para realizar la mensajería. Un gestor de colas de IBM MQ for z/OS puede proporcionar mensajería punto a punto en este entorno.

Una ventaja de utilizar un gestor de colas de IBM MQ for z/OS para proporcionar la mensajería es que la conexión de aplicaciones JMS puede participar plenamente en la funcionalidad de una red IBM MQ. Por ejemplo, pueden utilizar el puente ISM o intercambiar mensajes con los gestores de colas que se ejecutan en otras plataformas.

## **Conexión entre WebSphere Application Server y un gestor de colas**

Consulte [Utilización de IBM MQ y WebSphere Application Server juntos](#) para obtener más información.

## **Utilización de funciones de IBM MQ de aplicaciones JMS**

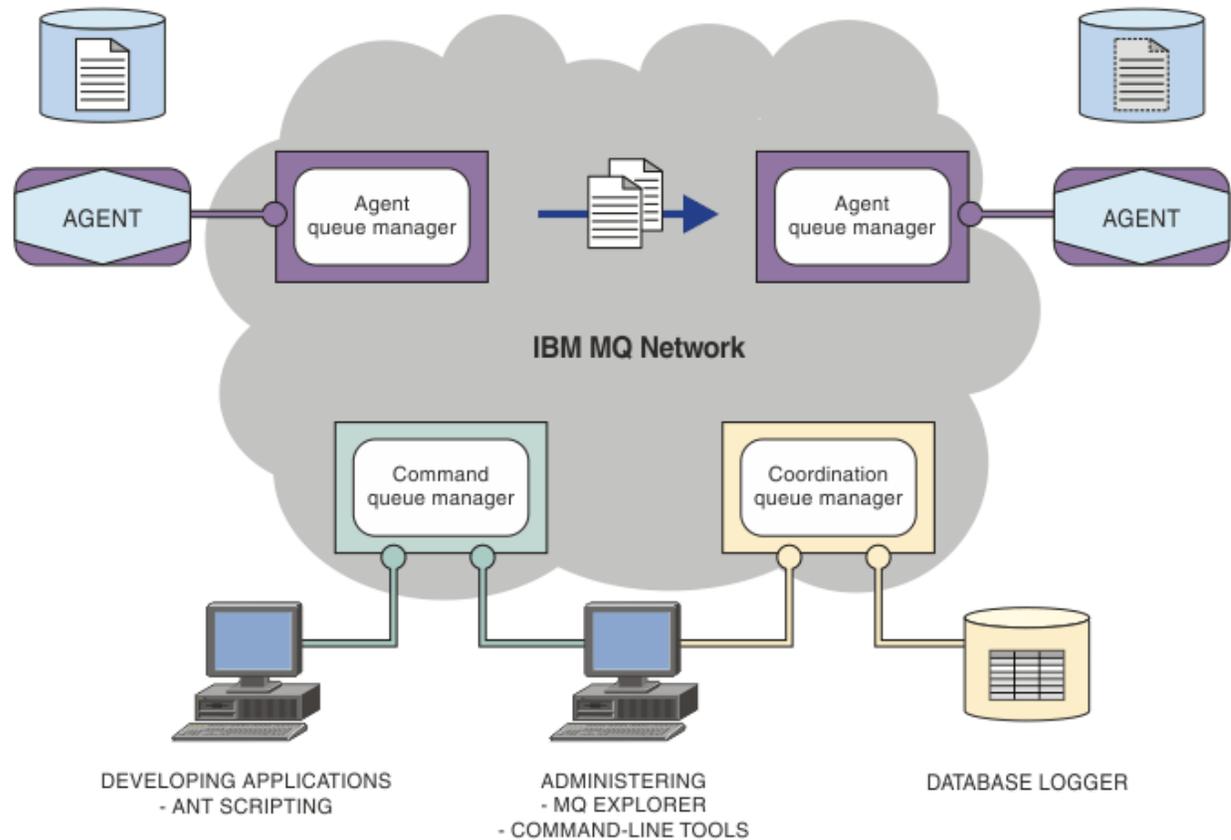
De forma predeterminada, los mensajes de JMS guardados en las colas de IBM MQ utilizan una cabecera MQRFH2 para contener parte de la información de la cabecera del mensaje de JMS. Muchas aplicaciones de IBM MQ heredadas no pueden procesar mensajes con estas cabeceras y requieren sus propias cabeceras características, por ejemplo, las aplicaciones MQCIH para CICS Bridge o MQWIH para IBM MQ Workflow. Para obtener más detalles sobre estas consideraciones especiales, consulte [Correlación de mensajes de JMS con mensajes de IBM MQ](#).

## **Managed File Transfer**

---

Managed File Transfer transfiere archivos entre sistemas de forma gestionada y auditable, independientemente del tamaño de archivo o de los sistemas operativos utilizados.

Puede utilizar Managed File Transfer para construir una solución personalizada, escalable y automatizada que le permita gestionar, confiar en, y proteger las transferencias de archivos. Managed File Transfer elimina costosas redundancias, reduce los costes de mantenimiento y maximiza las inversiones existentes en TI.



El diagrama muestra una topología simple de Managed File Transfer. Hay dos agentes, cada uno de ellos se conecta a su propio gestor de colas de agente en una red IBM MQ. Se transfiere un archivo desde el agente en un lado del diagrama, a través de la red IBM MQ, hasta el agente en el otro lado del diagrama. También en la red IBM MQ están el gestor de colas de coordinación y un gestor de colas de mandatos. Las aplicaciones y herramientas se conectan a estos gestores de colas para configurar, administrar, realizar operaciones y registrar la actividad de Managed File Transfer en la red IBM MQ.

Managed File Transfer se puede instalar como cuatro opciones diferentes, en función del sistema operativo y de la configuración general. Estas opciones son Managed File Transfer Agent, Managed File Transfer Logger, Managed File Transfer Service y Managed File Transfer Tools. Para obtener más información, consulte [Opciones del producto Managed File Transfer](#).

Puede utilizar Managed File Transfer para realizar las siguientes tareas:

- Crear transferencias de archivos gestionadas
  - Windows Linux Crear nuevas transferencias de archivos desde IBM MQ Explorer en plataformas Linux o Windows.
  - Crear nuevas transferencias de archivos desde la línea de mandatos en todas las plataformas soportadas.
  - Integrar la función de transferencia de archivos a la herramienta Apache Ant.
  - Escribir aplicaciones que controlen Managed File Transfer colocando mensajes en colas de mandatos de agente.
  - Planificar transferencias de archivos para que se realicen con posterioridad. Además, puede desencadenar transferencias de archivos planificadas basándose en una serie de sucesos de sistema de archivos, por ejemplo un nuevo archivo que se crea.

- Supervisar continuamente un recurso, por ejemplo un directorio, y cuando el contenido de dicho recurso cumpla una determinada condición predefinida, iniciar una tarea. Esta tarea puede ser una transferencia de archivos, un script Ant o un trabajo JCL.
- Transferir archivos a y de las colas de IBM MQ
- Transferir archivos a y desde servidores FTP, FTPS o SFTP.
- Transferir archivos a y desde nodos Connect:Direct.
- Transferir archivos de texto y binarios. Los archivos de texto se convierten automáticamente entre las páginas de códigos y las convenciones de fin de línea de los sistemas de origen y destino.
- Las transferencias se pueden proteger, mediante los estándares del sector para conexiones basadas en SSL (Socket Layer Layer).
- Ver las transferencias en curso y registrar información sobre todas las transferencias de la red
  -  Ver el estado de la transferencias en cursos desde IBM MQ Explorer en plataformas Linux o Windows.
  -  Comprobar el estado de las transferencias que se han completado utilizando IBM MQ Explorer en plataformas Linux o Windows.
  - Utilizar la característica de registrador de base de datos de Managed File Transfer para guardar los mensajes de registro en una base de datos Db2 u Oracle.

Managed File Transfer se basa en IBM MQ, que proporciona una entrega segura y única de mensajes entre aplicaciones. Puede aprovechar las diversas características de IBM MQ. Por ejemplo, puede utilizar la compresión de canales para comprimir los datos que envíe entre agentes a través de canales de IBM MQ y utilizar canales SSL para proteger los datos que envíe entre agentes. Los archivos se transfieren de forma fiable y pueden tolerar que falle la infraestructura a través de la cual se lleva a cabo la transferencia de archivos. Si experimenta una interrupción de la red, la transferencia de archivos se reinicia desde el punto en que se quedó cuando se restaura la conectividad.

Gracias a la consolidación de la transferencia de archivos con la red IBM MQ existente, puede evitar el gasto de los recursos necesarios de mantenimiento de dos infraestructuras distintas. Si todavía no es un cliente de IBM MQ, al crear una red IBM MQ de soporte de Managed File Transfer, está creando la red troncal para una futura implementación SOA. Si ya es cliente de IBM MQ, Managed File Transfer puede aprovechar la infraestructura existente de IBM MQ, incluidos IBM MQ Internet Pass-Thru e IBM Integration Bus.

Puede aprovechar las soluciones de alta disponibilidad de IBM MQ para mejorar la resiliencia de la configuración de Managed File Transfer. Si los agentes utilizan gestores de colas de datos replicados (RDQM), debe configurarlos para utilizar la característica de dirección IP flotante. Esto significa que los agentes utilizan la misma dirección IP para comunicarse con cualquiera de las tres instancias RDQM que se están ejecutando actualmente y se reconectan automáticamente en la migración tras error (consulte [Alta disponibilidad RDQM](#) y [Creación y supresión de una dirección IP flotante](#)). Si utiliza la solución de gestor de colas de varias instancias, las aplicaciones utilizan una dirección IP diferente para comunicarse con cada instancia, que gestiona la reconexión de cliente en la migración tras error (consulte [Gestores de colas de varias instancias](#) y [Reconexión de canal y cliente](#)).

Managed File Transfer se integra con una serie de otros productos IBM:

### **IBM Integration Bus**

Procese archivos que han sido transferidos por Managed File Transfer como parte de un flujo de IBM Integration Bus. Para obtener más información, consulte [Trabajar con MFT desde IBM Integration Bus](#).

### **IBM Sterling Connect:Direct**

Transfiera archivos a y desde una red Connect:Direct existente utilizando el puente Managed File TransferConnect:Direct. Para obtener más información, consulte [El puente Connect:Direct bridge](#).

## IBM Tivoli Composite Application Manager

IBM Tivoli Composite Application Manager proporciona un agente que puede utilizar para supervisar la información que se publica en el gestor de colas de coordinación.

### Conceptos relacionados

Opciones de producto de Managed File Transfer

[“Visión general de la topología de MFT” en la página 304](#)

Una visión general de cómo se conectan los agentes de Managed File Transfer con el gestor de colas de coordinación en una red IBM MQ.

[“¿Cómo funciona MFT con IBM MQ?” en la página 304](#)

Managed File Transfer interactúa de varias maneras con IBM MQ.

## ¿Cómo funciona MFT con IBM MQ?

Managed File Transfer interactúa de varias maneras con IBM MQ.

- Managed File Transfer transfiere archivos entre procesos de agente dividiendo cada archivo en uno o varios mensajes y transmitiendo los mensajes a través de la red IBM MQ.
- El agente procesa los datos de archivo utilizando mensajes no persistentes para minimizar el impacto en los registros de IBM MQ. La intercomunicación entre agentes hace que los procesos de agente regulen el flujo de los mensajes que contienen datos de archivo. Esto impide que los mensajes que contienen datos de archivo se acumulen en las colas de transmisión de IBM MQ y garantiza que si alguno de los mensajes no persistentes no se entrega, los datos de archivo se envíen de nuevo.
- Los agentes de Managed File Transfer utilizan un número de colas de IBM MQ.. Para obtener más información, consulte [Colas del sistema MFT y el tema del sistema](#).
- Si bien algunas de estas colas son estrictamente para uso interno, un agente puede aceptar solicitudes en forma de mensajes de mandato con formato especial que se envían a una cola específica que el agente lee. Tanto los mandatos de línea de mandatos como el conector de IBM MQ Explorer envían mensajes de IBM MQ al agente para indicarle que realice la acción deseada. Puede escribir aplicaciones de IBM MQ que interactúen con el agente de este modo. Para obtener más información, consulte [Control de MFT colocando mensajes en la cola de mandatos del agente](#).
- Los agentes de Managed File Transfer envían información sobre su estado, progreso y resultado de las transferencias a un gestor de colas de MQ designado como gestor de colas de coordinación. El gestor de colas de coordinación publica esta información a la que se pueden suscribir las aplicaciones que desean supervisar el progreso de las transferencias o mantener un registro de las transferencias que se han producido. Tanto los mandatos de línea de mandatos como el conector de IBM MQ Explorer pueden utilizar la información que se ha publicado. Puede escribir aplicaciones de IBM MQ que utilicen esta información. Para obtener más información sobre el tema en el que se publica la información, consulte [SYSTEM.FTE TemaFTE](#).
- Los componentes clave de Managed File Transfer aprovechan la capacidad de los gestores de colas de IBM MQ para almacenar y reenviar mensajes. Esto significa que si sufre una interrupción, las partes de la infraestructura que no se ven afectadas pueden continuar transfiriendo archivos. Esto se aplica por extensión al gestor de colas de coordinación, en el que una combinación de suscripciones duraderas de almacenamiento y envío permite que el gestor de colas de coordinación acepte dejar de estar disponible sin perder información clave sobre las transferencias de archivos que se han realizado.

## Visión general de la topología de MFT

Una visión general de cómo se conectan los agentes de Managed File Transfer con el gestor de colas de coordinación en una red IBM MQ.

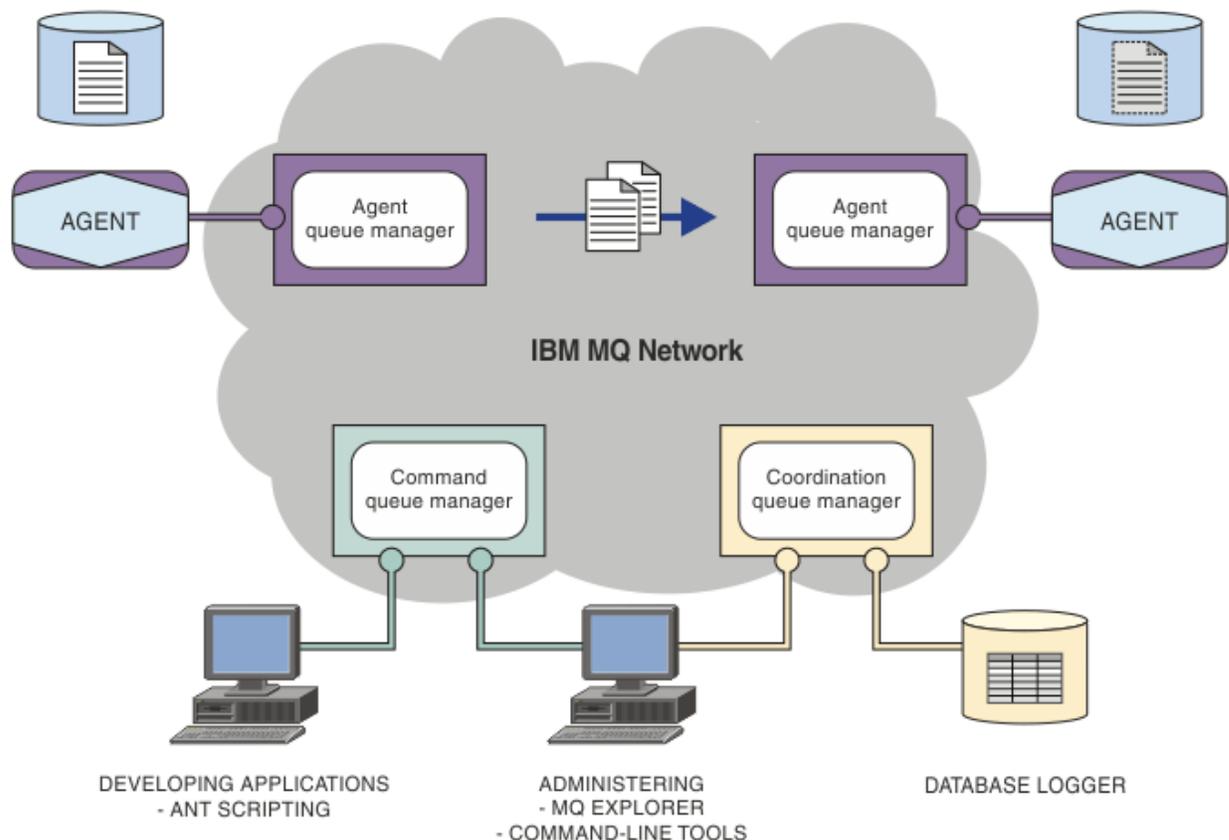
Los agentes de Managed File Transfer envían y reciben los archivos que se transfieren. Cada agente tiene su propio conjunto de colas en el gestor de colas asociado y el agente está vinculado al gestor de colas en modalidad de enlaces o de cliente. Un agente también puede utilizar el gestor de colas de coordinación como gestor de colas.

El gestor de colas de coordinación difunde información de auditoría y de archivos. El gestor de colas de coordinación representa un único punto para la recopilación de información de agente, de estado de transferencias y de auditoría de transferencias. No es necesario que el gestor de colas de coordinación esté disponible para que se produzcan transferencias. Si el gestor de colas de coordinación queda no disponible temporalmente, las transferencias continúan con toda normalidad. Los mensajes de auditoría y de estado se almacenan en los gestores de colas de agente hasta que el gestor de colas de coordinación queda disponible, y luego se pueden procesar con normalidad.

Los agentes se registran con el gestor de colas de coordinación y publican los detalles en dicho gestor de colas. Esta información de agente la utiliza el conector de Managed File Transfer para habilitar el inicio de transferencias desde IBM MQ Explorer. La información de agente recopilada en el gestor de colas de coordinación también es utilizada por los mandatos para mostrar información sobre agentes y estado de agentes.

La información de estado de transferencias y de auditoría de transferencias se publica en el gestor de cola de coordinación. El estado de la transferencia y la información de auditoría de transferencia de Managed File Transfer utiliza el estado de transferencia y la información de auditoría de transferencia para supervisar el progreso de las transferencias desde IBM MQ Explorer. La información de auditoría de transferencias almacenada en el gestor de colas de coordinación se puede conservar para facilitar la auditabilidad.

El gestor de colas de mandatos se utiliza para conectarse a la red IBM MQ y es el gestor de colas al que se ha conectado al emitir mandatos de Managed File Transfer.



### Conceptos relacionados

[“Managed File Transfer” en la página 301](#)

Managed File Transfer transfiere archivos entre sistemas de forma gestionada y auditable, independientemente del tamaño de archivo o de los sistemas operativos utilizados.

[“¿Cómo funciona MFT con IBM MQ?” en la página 304](#)

Managed File Transfer interactúa de varias maneras con IBM MQ.

## Descripción general de MFT REST API

La REST API da soporte a determinados mandatos de Managed File Transfer, incluidos el listado de transferencias y detalles sobre los agentes de transferencia de archivos.

A partir de IBM MQ 9.1.0, la REST API incluye opciones para listar todas las transferencias Managed File Transfer actuales y para consultar el estado de los agentes de Managed File Transfer. Para obtener más información, consulte [Iniciación a la REST API MFT](#).

## IBM MQ Internet Pass-Thru

---

IBM MQ Internet Pass-Thru (MQIPT) es un componente opcional de IBM MQ que se puede utilizar para implementar soluciones de mensajería entre sitios remotos a través de Internet.

A partir de IBM MQ 9.2.0, MQIPT es un componente opcional de IBM MQ. Para obtener los archivos de instalación de MQIPT para IBM MQ 9.3.x, vaya a [IBM Fix Central para IBM MQ](#). Antes de IBM MQ 9.2.0, MQIPT estaba disponible como paquete de soporte.

No es necesario que ejecute IBM MQ 9.3 para utilizar MQIPT en IBM MQ 9.3. Puede utilizar MQIPT para conectar cualquier versión soportada de IBM MQ y no tiene que instalar ningún otro componente IBM MQ en la misma versión que MQIPT.

Si ha adquirido la titularidad para IBM MQ, puede instalar tantas copias de MQIPT como sea necesario. Las instancias de MQIPT no se contarán en relación con la titularidad de IBM MQ que ha adquirido. Para obtener más información sobre la licencia de IBM MQ, consulte [Información de licencia de IBM MQ](#).

**Nota:** Esta documentación está relacionada con MQIPT en IBM MQ 9.3. Para la documentación del paquete de soporte de MQIPT (versión 2.1) en IBM Documentation, consulte [MQIPT \(SupportPac MS81\)](#) en la documentación de IBM MQ 9.0.

**Nota:** Si utiliza MQIPT 2.1 o anterior, se recomienda actualizar a MQIPT para IBM MQ 9.3, ya que la fecha de fin de soporte para el paquete de soporte de MQIPT era el 30th de septiembre de 2020.

IBM MQ Internet Pass-Thru se ejecuta como un servicio autónomo que puede recibir y reenviar flujos de mensajes de IBM MQ, ya sea entre dos gestores de colas de IBM MQ o entre un cliente de IBM MQ y un gestor de colas de IBM MQ.

MQIPT permite esta conexión cuando el cliente y el servidor no están en la misma red física.

Una o más instancias de MQIPT se pueden colocar en la vía de comunicación entre dos gestores de colas de IBM MQ o entre un cliente de IBM MQ y un gestor de colas de IBM MQ. Las instancias de MQIPT permiten que los dos sistemas IBM MQ intercambien mensajes sin necesidad de una conexión TCP/IP directa entre los dos sistemas. Esto resulta útil si la configuración del cortafuegos prohíbe una conexión TCP/IP directa entre dos sistemas.

MQIPT escucha en uno o varios puertos TCP/IP las conexiones de entrada, que pueden transportar mensajes de IBM MQ, mensajes de IBM MQ normales mediante el túnel HTTP, o mensajes cifrados utilizando TLS (Transport Layer Security) o SSL (Secure Sockets Layer). MQIPT puede manejar varias conexiones simultáneas.

El canal IBM MQ que realiza la solicitud de conexión TCP/IP inicial recibe el nombre de *linterlocutor*, el canal al que se intenta conectarse es el *respondedor* y el gestor de colas al que finalmente intenta contactar es el *gestor de colas de destino*.

MQIPT conserva los datos en la memoria a medida que los envía desde su origen a su destino. No se guardan datos en el disco (excepto la memoria de paginación a disco por parte del sistema operativo). La única vez que MQIPT accede al disco de forma explícita es para leer su archivo de configuración y grabar el registro de conexión y los registros de rastreo.

El rango completo de tipos de canal de IBM MQ se puede realizar a través de una o más instancias de MQIPT. La presencia de MQIPT en una ruta de comunicación no tiene ningún efecto sobre las

características funcionales de los componentes de IBM MQ conectados, pero puede que tenga algún efecto sobre el rendimiento de la transferencia de mensajes.

MQIPT se puede utilizar junto con IBM MQ y IBM Integration Bus, tal como se describe en [“Configuraciones posibles de MQIPT”](#) en la página 310.

Para instalar MQIPT, consulte [Instalación de MQIPT](#).

#### **Tareas relacionadas**

[Configuración de IBM MQ Internet Pass-Thru](#)

[Administración y configuración de IBM MQ Internet Pass-Thru](#)

#### **Referencia relacionada**

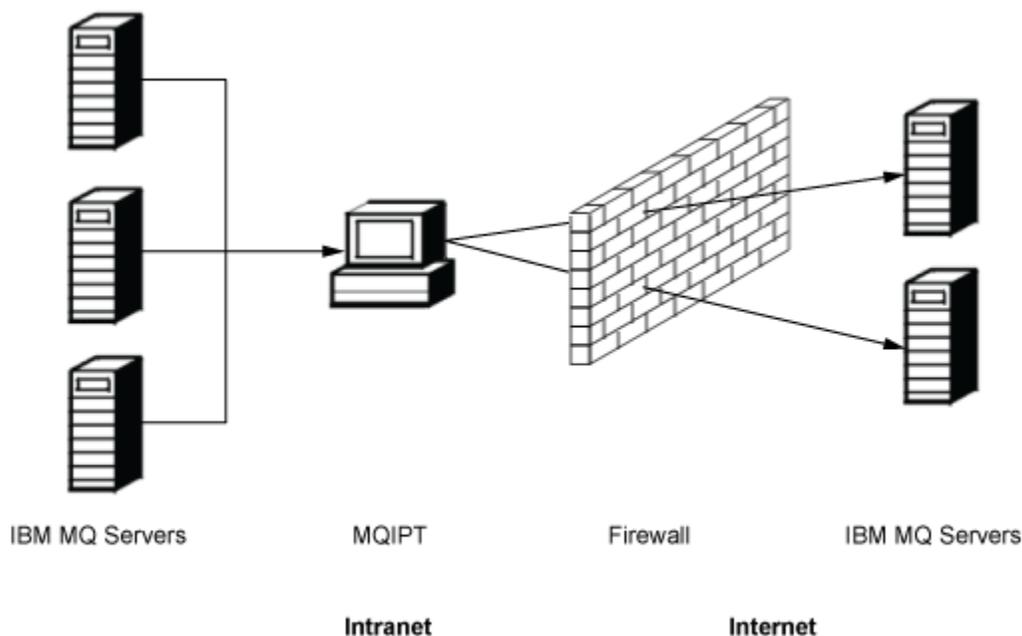
[Referencia de configuración de IBM MQ Internet Pass-Thru](#)

## **usos de MQIPT**

Hay una serie de usos potenciales para IBM MQ Internet Pass-Thru (MQIPT).

### **MQIPT se puede utilizar como un concentrador de canales**

Si se utiliza MQIPT de este modo, los canales hacia o desde varios hosts separados pueden mostrarse para un cortafuegos como si todos fueran hacia o desde el host MQIPT. Esto hace que resulte más fácil definir y gestionar las reglas de filtro del cortafuegos.



*Figura 79. Ejemplo de MQIPT como un concentrador de canales.*

### **MQIPT se puede colocar en un DMZ para proporcionar un único punto de acceso**

Un cortafuegos DMZ (a veces denominado cortafuegos de «zona desmilitarizada») es una configuración de cortafuegos para proteger redes de área local. Actúa como el punto expuesto a una red no fiable, normalmente Internet. Si MQIPT se coloca en un cortafuegos DMZ, en un ordenador con una dirección de protocolo de Internet (IP) conocida y de confianza, se puede utilizar MQIPT para escuchar las conexiones entrantes del canal de IBM MQ, que luego puede reenviar a la intranet de confianza; el cortafuegos interno debe permitir que este ordenador de confianza realice conexiones entrantes. En esta configuración, MQIPT impide que las solicitudes de acceso externas reciban las direcciones IP verdaderas de los sistemas en la intranet de confianza. De este modo, MQIPT proporciona un único punto de acceso.

Si es necesario, MQIPT se puede configurar para aceptar conexiones TLS y reenviar datos al destino utilizando una conexión TLS separada, por lo tanto, terminando la sesión TLS en la DMZ.

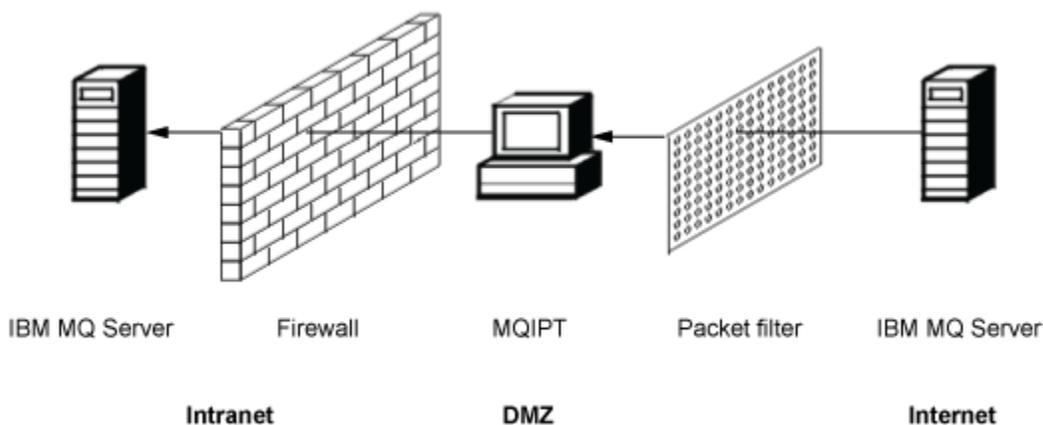


Figura 80. Ejemplo de MQIPT en un cortafuegos DMZ

### MQIPT se puede comunicar mediante el túnel HTTP

Si se despliegan dos instancias de MQIPT en línea, se pueden comunicar utilizando HTTP. La función de túnel HTTP permite transmitir las solicitudes a través de cortafuegos, utilizando los proxys HTTP existentes. El primer MQIPT inserta el protocolo IBM MQ en HTTP y el segundo extrae el protocolo IBM MQ de su derivador HTTP y lo reenvía al gestor de colas de destino.

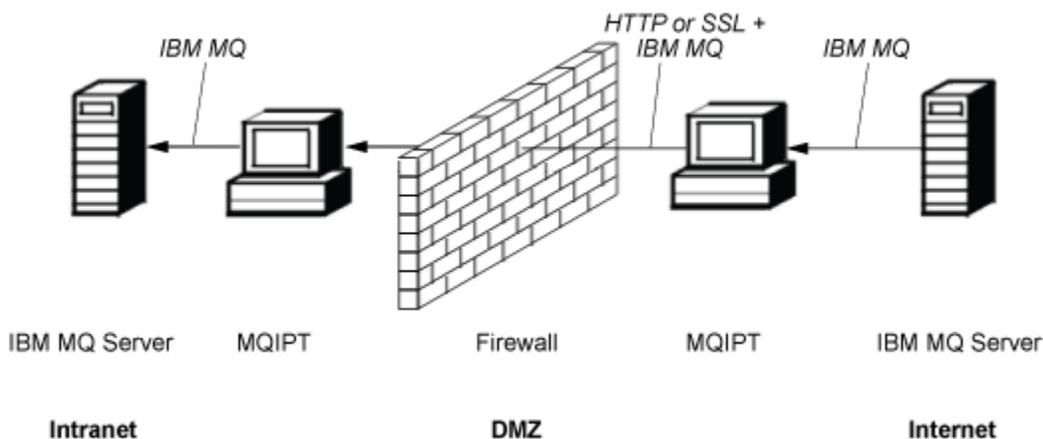


Figura 81. Ejemplo de MQIPT y túnel HTTP

### MQIPT puede cifrar mensajes

Si MQIPT se configura como en el ejemplo anterior, se pueden cifrar las solicitudes antes de la transmisión a través de cortafuegos. El primer MQIPT cifra los datos y el segundo los descifra utilizando SSL/TLS antes de enviarlos al gestor de colas de destino.

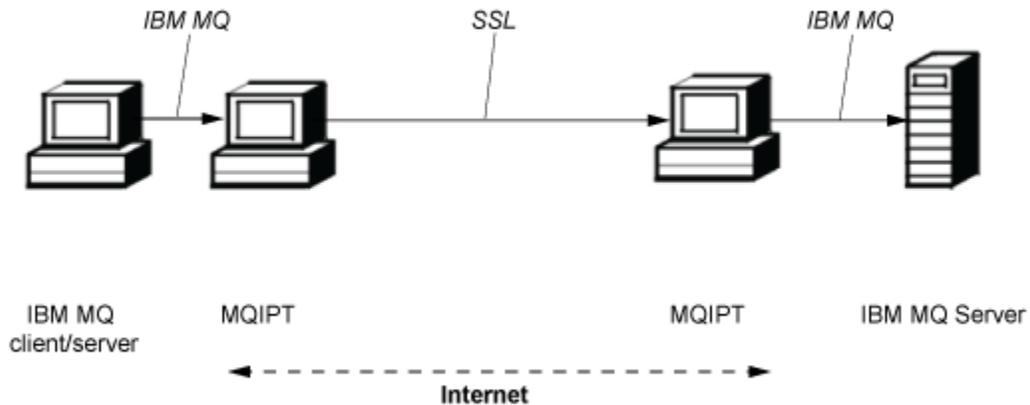


Figura 82. Ejemplo de MQIPT y SSL/TLS

## Cómo funciona MQIPT

En su configuración más sencilla, MQIPT actúa como un reenviador de protocolo de IBM MQ. Permanece a la escucha en un puerto TCP/IP y acepta solicitudes de conexión de los canales de IBM MQ.

Si se recibe una solicitud con el formato correcto, MQIPT establece una conexión TCP/IP adicional entre él y el gestor de colas de IBM MQ de destino. A continuación, pasa todos los paquetes de protocolo que recibe desde su conexión de entrada al gestor de colas de destino, y devuelve los paquetes de protocolos desde el gestor de colas de destino a la conexión de entrada original.

No hay ningún cambio en el protocolo de IBM MQ (cliente/servidor o gestor de colas para el gestor de colas) porque ninguno de los dos extremos conoce directamente la presencia del intermediario. No se necesitan nuevas versiones del cliente o código de servidor de IBM MQ.

Para utilizar MQIPT, el canal emisor debe configurarse para que utilice el nombre de host y puerto de MQIPT y no el nombre de host y puerto del gestor de colas de destino. Se define con la propiedad **CONNNAME** del canal IBM MQ. MQIPT lee los datos de entrada y simplemente los pasa al gestor de colas de destino. Otros campos de configuración, tales como el ID de usuario y la contraseña de un canal de cliente/servidor, se pasan del mismo modo al gestor de colas de destino.

## Varios gestores de colas

Se puede utilizar MQIPT para permitir el acceso a más de un gestor de colas de destino. Para que esto funcione, debe haber un mecanismo que indique a MQIPT con qué gestor de colas se ha de conectar, por lo que MQIPT utiliza el número de puerto TCP/IP para determinar a qué gestor de colas se ha de conectar.

Por lo tanto, puede configurar MQIPT para que escuche en varios puertos TCP/IP. Cada puerto de escucha se correlaciona con un gestor de colas de destino a través de una *rutade* MQIPT. Puede definir un máximo de 100 rutas de este tipo, que asocian un puerto TCP/IP de escucha con el nombre de host y el puerto del gestor de colas de destino. Esto significa que el nombre de host (dirección IP) del gestor de colas de destino nunca está visible para el canal de origen. Cada ruta puede manejar varias conexiones entre su puerto de escucha y el destino, y cada conexión actúa de forma independiente.

## Archivo de configuración de MQIPT

MQIPT utiliza un archivo de configuración denominado `mqipt.conf`. Este archivo contiene las definiciones de todas las rutas y sus propiedades asociadas. Consulte [Administración y configuración de IBM MQ Internet Pass-Thru](#) para obtener más información sobre `mqipt.conf`.

Cuando se inicia MQIPT, inicia cada una de las rutas que se listan en el archivo de configuración. Los mensajes se graban en la consola del sistema y muestran el estado de cada ruta. Cuando se muestra

el mensaje MQCPI078 para una ruta, significa que esta ruta está preparada para aceptar solicitudes de conexión.

## Configuraciones posibles de MQIPT

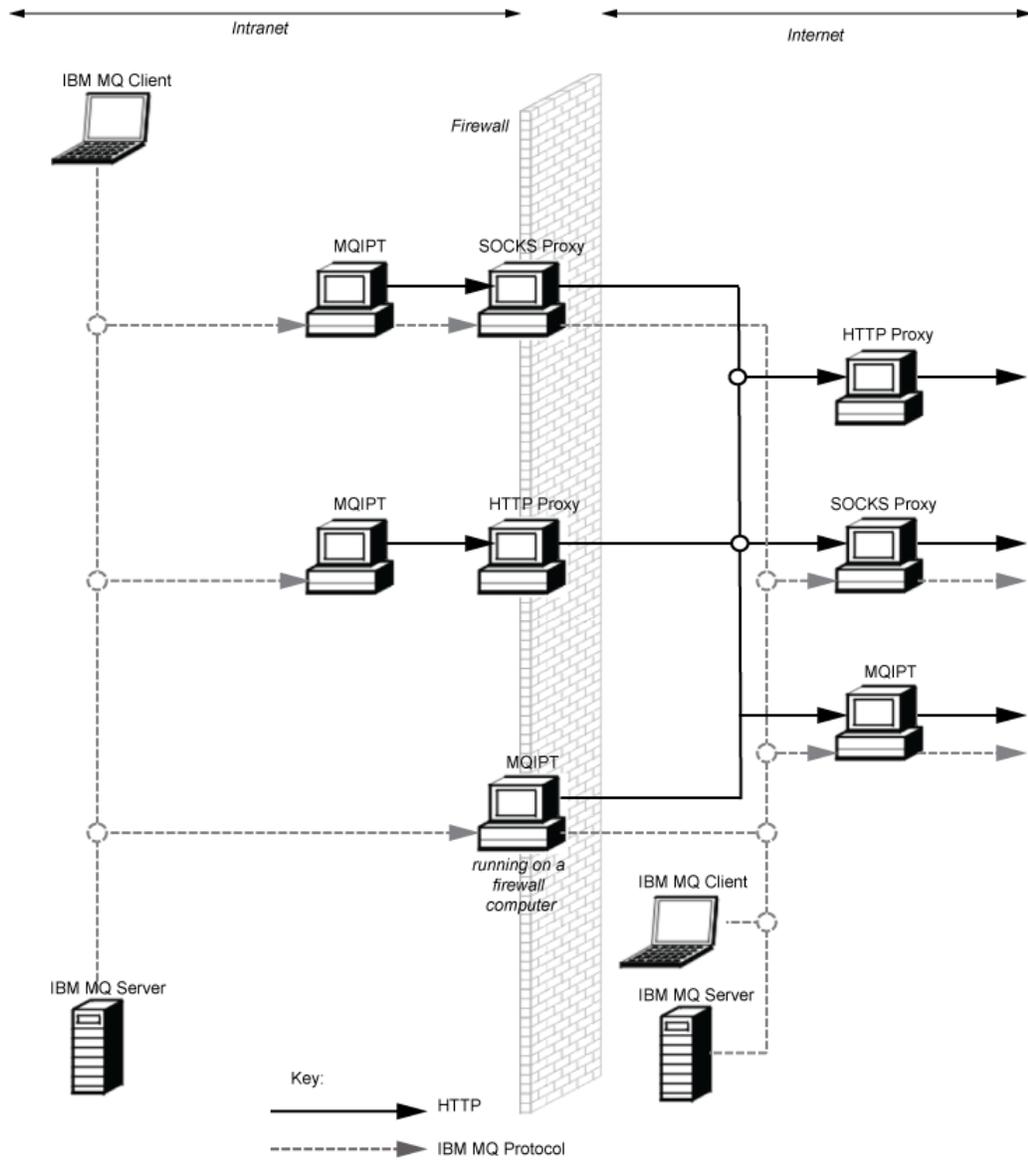
MQIPT se puede utilizar junto con IBM MQ y IBM Integration Bus.

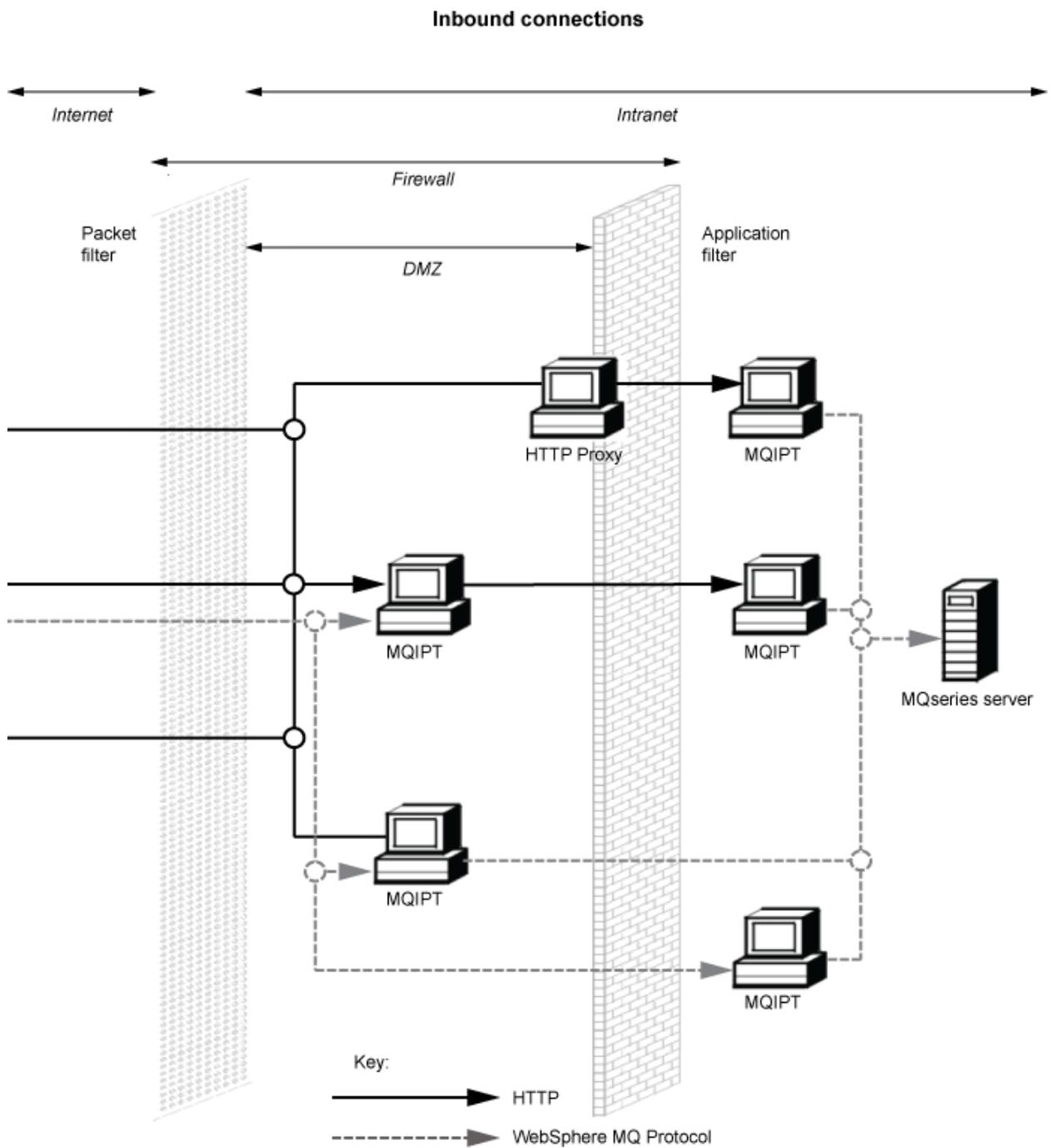
En la siguiente figura de varias partes se muestran muchas de las posibles configuraciones para MQIPT en una topología de IBM MQ. Ilustra los diferentes modos en que MQIPT puede enviar mensajes. Muestra clientes y servidores en una intranet, en un cortafuegos y en internet fuera del cortafuegos, pasando mensajes a MQIPT, el proxy HTTP o el proxy SOCKS, que los reenvía.

Los mensajes los recibe un proxy MQIPT o un proxy HTTP en un DMZ antes de pasar el mensaje a través del cortafuegos de entrada a un servidor.

Tenga en cuenta que el proxy HTTP, el proxy SOCKS y los sistemas MQIPT del extremo de la intranet del cortafuegos representan la posibilidad de varios sistemas encadenados conjuntamente en internet. Por ejemplo, un sistema MQIPT se puede comunicar a través de uno o varios sistemas proxy SOCKS o HTTP, o sistemas MQIPT adicionales, antes de llegar a su destino.

### Outbound connections





## Configuraciones compatibles

Escenarios de conexión compatibles en los que un cliente o gestor de colas de IBM MQ se comunica con MQIPT. Se utiliza la misma ruta de MQIPT, o una segunda ruta, para la comunicación con el gestor de colas de destino.

## Configuraciones compatibles con una única ruta de MQIPT

Puede utilizar una única ruta de MQIPT para comunicarse con IBM MQ.

Las columnas de la [Tabla 26](#) en la [página 313](#) contienen la información siguiente:

1. El protocolo utilizado entre IBM MQ y la ruta MQIPT. La conexión puede ser creada por un cliente de IBM MQ o un gestor de colas, y puede utilizar IBM MQ Formatos y Protocolos (FAP) o un protocolo SSL/TLS.
2. La modalidad en que opera la ruta de MQIPT. El formato de la comunicación a través de Internet entre MQIPT y IBM MQ, viene determinado por la configuración de la ruta MQIPT. Tenga en cuenta que donde la tabla menciona SSL, también puede utilizar TLS.
3. El protocolo utilizado entre la ruta de MQIPT y el gestor de colas de destino.

<b>1. IBM MQ Protocolo de origen</b>	<b>2. Modalidad de la ruta de MQIPT</b>	<b>3. IBM MQ Protocolo de destino</b>
FAP	FAP-proxy (valor predeterminado)	FAP
	FAP-server y SSL-client	SSL/TLS
SSL/TLS	SSL-proxy	SSL/TLS
	SSL-server y FAP-client	FAP
	SSL-server and SSL-client	SSL/TLS

## Configuraciones compatibles con más de una ruta de MQIPT

Puede optar por utilizar más de una ruta, en una o varias instancias de MQIPT, para comunicarse con IBM MQ.

Las columnas de la [Tabla 27](#) en la [página 314](#) contienen la información siguiente:

1. El protocolo utilizado entre IBM MQ y la primera ruta de MQIPT. La conexión puede ser creada por un cliente de IBM MQ o un gestor de colas, y puede utilizar IBM MQ Formatos y Protocolos (FAP) o un protocolo SSL/TLS.
2. La modalidad en que opera la primera ruta de MQIPT. El formato de la comunicación a través de Internet entre MQIPT y IBM MQ, viene determinado por la configuración de la ruta MQIPT. Tenga en cuenta que donde la tabla menciona SSL, también puede utilizar TLS.
3. La modalidad en que opera la segunda ruta de MQIPT.
4. El protocolo utilizado entre la segunda ruta de MQIPT y el gestor de colas de destino.

Tabla 27. Configuraciones válidas con varias instancias de MQIPT

1. IBM MQ Protocolo de origen	2. Modalidad de la primera ruta de MQIPT	3. Modalidad de la segunda ruta de MQIPT	4. IBM MQ Protocolo de destino
FAP (valor predeterminado)	FAP-proxy (valor predeterminado)	FAP-proxy (valor predeterminado)	FAP
	FAP-server y SSL-client	SSL-proxy	SSL/TLS
		SSL-server y FAP-client	FAP
		SSL-server and SSL-client	SSL/TLS
	HTTP-client	HTTP-server y SSL-client	SSL/TLS
	HTTPS-client	HTTPS-server y SSL-client	SSL/TLS
	HTTP-client	HTTP-server	FAP
HTTPS-client	HTTPS-server	FAP	
SSL/TLS	SSL-proxy	SSL-proxy	SSL/TLS
		SSL-server y FAP-client	FAP
		SSL-server and SSL-client	SSL/TLS
	HTTP-client	HTTP-server	FAP
	HTTPS-client	HTTPS-server	SSL/TLS
	HTTP-client	HTTP-server y SSL-client	FAP
	HTTPS-client	HTTPS-server y SSL-client	SSL/TLS

## Configuraciones de canales soportadas

Se da soporte a todos los tipos de canal IBM MQ, pero la configuración está restringida a las conexiones TCP/IP. En un cliente o gestor de colas de IBM MQ, MQIPT aparece como si fuera el gestor de colas de destino. Donde la configuración del canal requiere un host de destino y un número de puerto, se especifican el nombre de host de MQIPT y el número de puerto de escucha.

### Canales de cliente/servidor

MQIPT permanece a la escucha de las solicitudes de conexión de cliente entrantes y, a continuación, las reenvía utilizando el túnel HTTP, SSL/TLS o como paquetes de protocolo estándar de IBM MQ. Si MQIPT utiliza el túnel HTTP o SSL/TLS, las envía en una conexión a un segundo MQIPT. Si no está utilizando el túnel HTTP, las envía en una conexión a lo que ve como el gestor de colas de destino (aunque esto pueda ser a su vez un MQIPT adicional). Cuando el gestor de colas de destino ha aceptado la conexión del cliente, transmite los paquetes entre el cliente y el servidor.

### Canales de emisor/receptor del clúster

Si MQIPT recibe una solicitud de entrada desde un canal emisor del clúster, presupone que el gestor de colas está habilitado para SOCKS y la dirección de destino verdadera se obtiene durante el proceso de reconocimiento de SOCKS. Envía la solicitud al siguiente MQIPT o gestor de colas de destino, exactamente del mismo modo que para los canales de conexión de cliente. Esto también incluye los canales de emisor de clúster definidos automáticamente.

**Emisor/receptor**

Si MQIPT recibe una solicitud de entrada desde un canal emisor, la envía al siguiente MQIPT o gestor de colas de destino, exactamente del mismo modo que para los canales de conexión de cliente. El gestor de colas de destino valida la solicitud de entrada e inicia el canal receptor, si resulta adecuado. Todas las comunicaciones entre el canal emisor y receptor (incluidos los flujos de seguridad) se transmiten en relé.

**Solicitante/servidor**

Esta combinación se maneja de la misma manera que las configuraciones anteriores. La validación de la solicitud de conexión la realiza el canal del servidor en el gestor de colas de destino.

**Solicitante/emisor**

La configuración de "devolución de llamada" puede resultar útil si los dos gestores de colas no pueden establecer conexiones directas entre sí, pero ambos pueden conectar con MQIPT y aceptar conexiones desde el mismo.

**Servidor/solicitante y servidor/receptor**

MQIPT los maneja de la misma forma que maneja la configuración de Sender/Receiver .

## Terminación del canal y condiciones de anomalía

Cuando MQIPT detecta el cierre (normal o anormal) de un canal de IBM MQ, propaga su cierre. Si cierra una ruta utilizando MQIPT, se cierran todos los canales que transitan esta ruta.

MQIPT proporciona un recurso de tiempo de espera desocupado opcional. Si MQIPT detecta que un canal ha estado desocupado durante un periodo de tiempo que supera el tiempo de espera, realiza una conclusión inmediata de las dos conexiones específicas.

Los sistemas IBM MQ en cualquiera de ambos extremos del canal observan estas condiciones anómalas de cierre, ya sea como anomalías de red o como terminación del canal por parte de su socio. A continuación, se puede reiniciar y recuperar el canal (si el error se produce durante un periodo dudoso del protocolo), como si no se estuviera utilizando MQIPT.

## Seguridad de mensajes

La gestión de colas distribuidas de IBM MQ garantiza que los mensajes se entreguen correctamente. Esto continúa siendo el caso cuando está presente MQIPT entre los dos extremos del canal. MQIPT no almacena datos de mensajes ni participa en el procedimiento de punto de sincronización que garantiza la entrega correcta de los mensajes.

Cuando se utilizan mensajes de IBM MQ rápidos y no persistentes, si la ruta de MQIPT falla o se reinicia cuando un mensaje de IBM MQ está en tránsito, es posible que se pierda el mensaje. Antes de reiniciar la ruta, asegúrese de que todos los canales IBM MQ que utilizan la ruta MQIPT estén inactivos.

Para obtener más información sobre la seguridad de los mensajes en IBM MQ, consulte [Seguridad de mensajes](#).

## Gestores de colas de varias instancias y alta disponibilidad

MQIPT se puede utilizar con gestores de colas de varias instancias en entornos de alta disponibilidad.

MQIPT no tiene ningún estado persistente y, por lo tanto, no supone ninguna ventaja realizar la migración tras error de MQIPT a otro sistema. En su lugar, tenga varias instancias de MQIPT con archivos de configuración `mqipt.conf` idénticos que se ejecutan en sistemas diferentes. Supervise la disponibilidad de cada instancia de MQIPT y reiniciela (en el mismo sistema) si es necesario. Esto proporciona un conjunto de instancias de MQIPT idénticas que se pueden utilizar para direccionar las conexiones. A continuación, debe asegurarse de que IBM MQ puede direccionar las conexiones a MQIPT y de que MQIPT puede reenviar esas conexiones al gestor de colas de destino.

Los canales de IBM MQ de salida se pueden dirigir a una instancia de MQIPT disponible de varias maneras, por ejemplo:

- Utilice un equilibrador de carga o un direccionador de alta disponibilidad, tal como IBM Network Dispatcher, desde el producto WebSphere Edge Components.
- Especifique varios nombres de conexión en la definición del canal IBM MQ utilizando una lista separada por comas. A continuación, IBM MQ intenta conectar con cada dirección de MQIPT por orden, hasta que encuentra una instancia de MQIPT disponible.

También debe direccionar las conexiones desde MQIPT al gestor de colas de destino. Si la configuración de alta disponibilidad garantiza que la dirección migra tras error con el gestor de colas de destino, no es necesaria ninguna configuración especial de MQIPT: Especifique la dirección IP de destino en la propiedad de ruta **Destination** y permita la operación de migración tras error para mover la dirección IP con el gestor de colas.

No obstante, si la dirección IP del gestor de colas cambia después de una migración tras error, debe establecer que MQIPT envíe la conexión al destino correcto. Esto se puede hacer de una de las maneras siguientes:

- Escriba una salida de direccionamiento que compruebe qué dirección IP y número de puerto están accesibles y, a continuación, reemplace el destino de ruta de cada conexión. Existen algunas salidas de direccionamiento de ejemplo que se proporcionan con MQIPT; se pueden adaptar para este fin.
- Utilice un equilibrador de carga de alta disponibilidad para redireccionar la conexión.
- Defina varias rutas de MQIPT, una para cada dirección IP y puerto donde es posible que se esté ejecutando el gestor de colas. A continuación, dirija las conexiones de IBM MQ a las distintas rutas de MQIPT, por ejemplo, listando todas las direcciones IP de ruta y los números de puerto en una lista separada por comas en el nombre de conexión del canal de salida.

También es importante ajustar todos los componentes de un extremo a otro en la vía de acceso de red:

1. Los intentos de conexión con sistemas no disponibles deben fallar rápidamente, de modo que los intentos de reconexión puedan pasar al primer destino disponible.

Para las rutas SSL de MQIPT, ajuste la propiedad de ruta **SSLClientConnectTimeout** para asegurar un error de conexión rápido para destinos no disponibles. Consulte la documentación de IBM MQ para obtener más detalles sobre los parámetros de ajuste de IBM MQ. Asimismo, consulte la documentación del sistema operativo para obtener detalles sobre cómo ajustar TCP/IP para el sistema operativo. En todos los casos, los intentos de conexión erróneos deben devolver rápidamente un error de red (por ejemplo, un paquete de restablecimiento TCP) o deben agotar el tiempo de espera sin ningún retardo indebido.

2. Las conexiones activas con un sistema anómalo se deben interrumpir rápidamente para que se puedan establecer nuevas conexiones.

También debe tener en cuenta el impacto de una migración tras error en un periodo en que las conexiones están utilizando MQIPT activamente. Es probable que las conexiones de red se interrumpan durante una migración tras error. En el caso de las aplicaciones cliente, puede utilizar la característica de reconexión automática de cliente de IBM MQ para restablecer las conexiones interrumpidas. Para los canales de mensajes, puede especificar un intervalo de reintento corto, de modo que el canal se vuelva a conectar rápidamente. Consulte la documentación de IBM MQ para obtener más información sobre la reconexión automática del cliente y la configuración del reintento del canal de mensajes.

## V 9.3.5 IBM MQ Console y REST API

Puede utilizar IBM MQ Console y REST API para administrar IBM MQy realizar operaciones de mensajería utilizando HTTP.

- Puede utilizar IBM MQ Console para realizar tareas de administración básicas desde un navegador web. Si desea más información, consulte [Administración utilizando IBM MQ Console](#).
- Puede utilizar la administrative REST API para administrar objetos IBM MQ como, por ejemplo, gestores de colas y colas, y agentes y transferencias Managed File Transfer. Si desea más información, consulte [Administración utilizando REST API](#).

- Puede utilizar messaging REST API para realizar una mensajería de punto a punto simple y de publicación. Para obtener más información, consulte [Mensajería utilizando REST API](#).

## Opciones de instalación

IBM MQ Console y REST API se ejecutan en un servidor WebSphere Liberty , denominado mqweb. A partir de IBM MQ 9.3.5, puede instalar el servidor mqweb como un componente opcional en una instalación de IBM MQ o como una instalación autónoma de IBM MQ Web Server .

### Linux > V 9.3.5 Instalación de IBM MQ Web Server autónomo

A partir de IBM MQ 9.3.5, el servidor mqweb se puede ejecutar en una instalación autónoma de IBM MQ Web Server. Una instalación autónoma de IBM MQ Web Server le permite instalar y ejecutar el servidor mqweb en sistemas independientes de las instalaciones de IBM MQ . La instalación de un IBM MQ Web Server autónomo proporciona una mayor flexibilidad en cuanto a qué sistemas, y el número de sistemas, en los que elige ejecutar los servidores mqweb. Si es necesario, se pueden ejecutar varias instancias del servidor mqweb en distintas máquinas para proporcionar la escalabilidad y la disponibilidad que necesita.

Si ha adquirido la titularidad de IBM MQ , puede instalar tantas copias como sea necesario del IBM MQ Web Server autónomo. Las instancias de IBM MQ Web Server no se contarán en relación con la titularidad de IBM MQ que ha adquirido. Para obtener más información sobre la licencia de IBM MQ, consulte [Información de licencia de IBM MQ](#).

Las restricciones siguientes se aplican en una instalación autónoma de IBM MQ Web Server :

- IBM MQ Console se puede utilizar para administrar sólo gestores de colas remotos.
- El messaging REST API sólo se puede utilizar con gestores de colas remotos.
- El administrative REST API no está disponible.

El IBM MQ Web Server autónomo solo está soportado en plataformas Linux .

Para obtener más información sobre la instalación del IBM MQ Web Server autónomo, consulte [Instalación del IBM MQ Web Server autónomo](#).

### Componente opcional de una instalación de IBM MQ

Puede optar por instalar el componente IBM MQ Console y REST API como parte de una instalación de IBM MQ .

Todas las características IBM MQ Console y REST API están disponibles cuando el servidor mqweb se ejecuta en una instalación de IBM MQ .

- IBM MQ Console se puede utilizar para administrar gestores de colas locales y remotos.
- El messaging REST API se puede utilizar con gestores de colas locales y remotos.
- administrative REST API se puede utilizar para administrar gestores de colas locales y remotos.

Para utilizar el componente IBM MQ Console y REST API , instale el componente siguiente como parte de la instalación de IBM MQ :

- **AIX** En AIX, instale el conjunto de archivos mqm.web.rte.
- **IBM i** En IBM i, instale el componente WEB.
- **Linux** En Linux, instale el componente MQSeriesWeb.
- **Windows** En Windows, instale la característica Web Administration.
- **z/OS** En z/OS, instale la característica IBM MQ for z/OS UNIX System Services Web Components.



## Avisos

---

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos.

Es posible que IBM no ofrezca los productos, servicios o las características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar podrá utilizarse cualquier producto, programa o servicio equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país o envíe las consultas por escrito a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokio 103-8510, Japón

**El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones contradigan la legislación vigente:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INCUMPLIMIENTO, COMERCIALIZACIÓN O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Cualquier referencia en esta información a sitios web que no son de IBM se realiza por razones prácticas y de ninguna manera sirve como un respaldo de dichos sitios web. Los materiales de dichos sitios web no forman parte de este producto de IBM y la utilización de los mismos será por cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione del modo que considere apropiado sin incurrir por ello en ninguna obligación con respecto al usuario.

Los titulares de licencias de este programa que deseen información del mismo con el fin de permitir: (i) el intercambio de información entre los programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N

Rochester, MN 55901  
U.S.A.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

El programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible para el mismo lo proporciona IBM bajo los términos del Acuerdo de cliente de IBM, el Acuerdo de licencia de programas internacional de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar de manera significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en sistemas disponibles generalmente. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se obtuvo de los proveedores de esos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad o alguna reclamación relacionada con productos que no sean de IBM. Todas las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relacionadas con una futura intención o tendencia de IBM están sujetas a cambios o se pueden retirar sin previo aviso y sólo representan metas y objetivos.

Este documento contiene ejemplos de datos e informes que se utilizan diariamente en la actividad de la empresa. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es puramente casual.

#### LICENCIA DE DERECHOS DE AUTOR:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pagar ninguna cuota a IBM para fines de desarrollo, uso, marketing o distribución de programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. IBM, por tanto, no puede garantizar la fiabilidad, servicio o funciones de estos programas.

Puede que si visualiza esta información en copia software, las fotografías e ilustraciones a color no aparezcan.

## Información acerca de las interfaces de programación

---

La información de interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación para su uso con este programa.

Este manual contiene información sobre las interfaces de programación previstas que permiten al cliente escribir programas para obtener los servicios de WebSphere MQ.

Sin embargo, esta información puede contener también información de diagnóstico, modificación y ajustes. La información de diagnóstico, modificación y ajustes se proporciona para ayudarle a depurar el software de aplicación.

**Importante:** No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

## Marcas registradas

---

IBM, el logotipo de IBM , [ibm.com](http://ibm.com), son marcas registradas de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Hay disponible una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information"[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas.

Microsoft y Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o otros países.

UNIX es una marca registrada de Open Group en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Este producto incluye software desarrollado por Eclipse Project (<https://www.eclipse.org/>).

Java y todas las marcas registradas y logotipos son marcas registradas de Oracle o sus afiliados.







Número Pieza:

(1P) P/N: