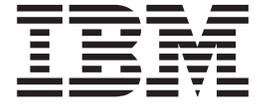


IBM WebSphere Process Server for Multiplatforms



# Visiones generales técnicas

*Versión 7.0.0*

**30 de abril de 2010**

Este manual es la traducción del original en inglés *IBM WebSphere Process Server for Multiplatforms Technical Overviews Version 7.0.0*.

Esta edición se aplica a la versión 7, release 0, modificación 0 de WebSphere Process Server for Multiplatforms (número de producto 5724-L01) y a todos los releases y las modificaciones subsiguientes hasta que se indique lo contrario en nuevas ediciones.

Para enviar comentarios sobre este documento, envíe un mensaje de correo electrónico a [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com). Esperamos sus comentarios.

Cuando se envía información a IBM, se otorga a IBM un derecho no exclusivo de utilizar o distribuir la información del modo que estime apropiado sin incurrir por ello en ninguna obligación con el remitente.

© Copyright IBM Corporation 2005, 2010.

---

# Contenido

## Visiones generales técnicas . . . . . 1

|  |    |
|--|----|
| Service Component Architecture . . . . .                 | 1  |
| SCA e invocación de servicios . . . . .                  | 1  |
| Módulos. . . . .   | 2  |
| Componentes de servicio . . . . .                        | 5  |
| Referencias autónomas . . . . .                          | 16 |
| Objetos de negocio . . . . .                             | 16 |
| Objetos de negocio . . . . .                             | 17 |
| Definición de objetos empresariales . . . . .            | 18 |
| Trabajo con objetos empresariales . . . . .              | 19 |
| Objetos empresariales especiales . . . . .               | 21 |
| Modalidad de análisis de objetos empresariales . . . . . | 22 |
| Relaciones . . . . .                                     | 25 |

|  |    |
|--|----|
| Servicio de relaciones . . . . .   | 28 |
| Gestor de relaciones . . . . .   | 28 |
| Relaciones en entornos de despliegue de red . . . . .                        | 29 |
| APIs de servicio de relaciones . . . . .                                     | 29 |
| El bus de integración de servicio en WebSphere<br>Process Server . . . . .   | 29 |
| Conexión de servicios a través de un bus de<br>servicio de empresa . . . . . | 30 |
| Infraestructura de mensajería de Enterprise<br>Service Bus . . . . .         | 31 |
| Aplicaciones de servicio y módulos de servicio . . . . .                     | 36 |
| Clientes de servicios de mensajes . . . . .                                  | 51 |



---

## Visiones generales técnicas

La información de visión general técnica presenta los estándares y los aspectos técnicos relacionados con la arquitectura del producto.

---

### Service Component Architecture

SCA (Service Component Architecture) habilita una arquitectura orientada a servicios y está disponible a través de muchas empresas, como IBM®. SCA es una plataforma y un modelo de programación independiente de proveedor que proporciona unos métodos sencillos y coherentes para expresar la lógica empresarial y los datos empresariales como servicios SOA, independientemente de los detalles de implementación técnica. En esta sección, se examinan los servicios SCA y los objetos de datos.

#### SCA e invocación de servicios

Si tiene en cuenta los tres aspectos de un modelo de programación, que son los datos, la invocación y la composición, y aplica alguno de los nuevos paradigmas de un enfoque basado en servicios, el nuevo modelo de programación para SOA empieza a tomar importancia. SCA (Service Component Architecture) proporciona un modo de invocar los servicios empresariales dentro de las soluciones SOA.

Las construcciones que componen la arquitectura orientada a servicios incluyen un modo de representar los datos que se intercambian entre los servicios, un mecanismo para invocar los servicios y un modo de componer servicios dentro de aplicaciones empresariales integradas de gran tamaño. Actualmente, existen muchos modelos de programación diferentes para dar soporte a los mismos. Esta situación presenta a los desarrolladores el reto de tener que resolver no sólo un problema empresarial concreto sino también de seleccionar y comprender la tecnología de implementación adecuada. Uno de los objetivos importantes de la solución SOA de WebSphere Process Server es mitigar estas complejidades. Esto se lleva a cabo mediante la convergencia de diferentes modelos de programación utilizados para implementar las aplicaciones empresariales orientadas a servicios en un modelo de programación simplificado.

En esta sección se describe específicamente SCA (Service Component Architecture) en WebSphere Process Server como el modelo de componente orientado a servicios para definir e invocar los servicios empresariales. SCA juega un papel importante a la hora de proporcionar un modelo de invocación para la solución SOA en WebSphere Process Server. SCA también juega un papel en la creación de servicios empresariales dentro de aplicaciones empresariales compuestas.

En primer lugar, veremos que los datos se representan primordialmente mediante XML (Extensible Markup Language) y se programan con objetos empresariales basados en la especificación SDO (Service Data Object) o mediante los recursos XML nativos como, por ejemplo, XPath o XSLT (Extensible Stylesheet Language Transformation). En segundo lugar, la invocación de servicios se correlaciona con SCA (Service Component Architecture). Para terminar, la composición se incorpora en la orquestación de procesos mediante BPEL (Business Process Execution Language). En la figura se muestran los tres aspectos de este nuevo modelo de programación.

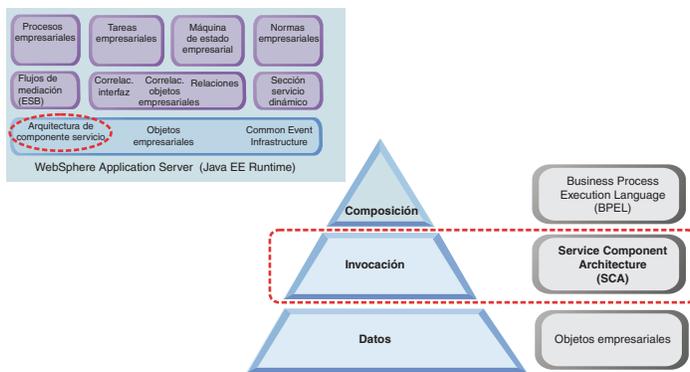


Figura 1. Representación de datos, invocación y componentes dentro de un modelo de programación para SOA

SCA tiene como objetivo proporcionar un modelo de programación simplificado para escribir aplicaciones que se ejecutan en un entorno de tiempo de ejecución Java EE, y está basado en los conceptos y técnicas que son una refinación de la tecnología Java EE existente. Uno de los aspectos importantes de SCA es que permite separar la lógica empresarial de las aplicaciones y los detalles de la implementación. Para poder realizarlo, SCA proporciona una sola abstracción para los tipos de servicios que puede estar ya expresada como beans de sesión, servicios Web, clases Java o BPEL. La posibilidad de separar la lógica empresarial de la lógica de la infraestructura es importante para ayudar a disminuir los recursos de TI necesarios para crear una aplicación empresarial y proporciona a los desarrolladores más tiempo para trabajar en un problema empresarial concreto, sin tener que emplear su tiempo en detalles sobre qué tecnología de implementación se ha de utilizar.

## Módulos

Un *módulo* es una unidad de despliegue que determina qué artefactos se empaquetan juntos en un archivo archivador empresarial (EAR). Los componentes dentro de un módulo se ubican para el rendimiento y pueden pasar sus datos por referencia. Un módulo se puede ver como una mecanismo de ámbito; es decir, establece un límite organizativo para artefactos.

Un módulo es un compuesto de componentes de servicio, importaciones y exportaciones. Los componentes de servicio, importaciones y exportaciones residen en el mismo proyecto y la misma carpeta raíz, que también contienen la conexión que enlaza los componentes y los enlaces necesarios para las importaciones y exportaciones. Asimismo, un módulo contiene las implementaciones e interfaces a las que hacen referencia sus componentes, importaciones y exportaciones, o éstos se pueden colocar en otro proyectos como, por ejemplo, un proyecto de biblioteca.

Hay dos tipos de módulos. En primer lugar, un módulo llamado *módulo* (al que a veces se hace referencia como un módulo de integración empresarial) que contiene una selección de muchos tipos de componentes utilizados, a menudo, para soportar un proceso de negocio. En segundo lugar, un módulo llamado *módulo de mediación*, que contiene hasta un componente, uno o más flujos de mediación, además de cero o más componentes Java que aumentan el componente de flujo de mediación.

Un módulo puede contener uno o más componentes de flujo de mediación.

¿Por qué hay dos tipos de módulo? El primer tipo de módulo se ha diseñado básicamente para los procesos de negocio. Un módulo de mediación es como una pasarela a servicios externos existentes, que es común en arquitecturas de bus de servicio empresarial. Se accede a estos servicios externos o exportaciones de un módulo de mediación mediante importaciones o proveedores de servicios. Al desacoplar los solicitantes de servicio de cliente de los proveedores de servicios mediante un flujo de mediación, las aplicaciones ganan flexibilidad y capacidad de recuperación, un objetivo de la arquitectura orientada a servicios. Por ejemplo, el flujo de mediación puede registrar los mensajes de entrada, direccionar mensajes a un servicio específico determinado durante el tiempo de ejecución o transformar los datos para hacerlos aptos para pasar a otro servicio. Estas funciones se pueden añadir o modificar a lo largo del tiempo sin modificar el solicitante o los servicios de proveedor.

Un módulo genera una aplicación de servicio probada y desplegada en WebSphere Process Server. Un módulo de mediación genera una aplicación de servicio probada y desplegada en WebSphere Process Server o en WebSphere Enterprise Service Bus Server. Ambos tipos de módulos soportan las importaciones y las exportaciones.

A menudo, las implementaciones, interfaces, objetos de negocio, correlaciones de objetos de negocio, roles, relaciones y otros artefactos se deben compartir entre los módulos. Una *biblioteca* es un proyecto utilizado para almacenar estos recursos compartidos.

En Figura 2 en la página 4, el módulo contiene dos componentes de servicio, cada uno contiene una implementación. Asimismo, el módulo contiene las interfaces y referencias apropiadas necesarias para los componentes de servicio. El segundo componente de servicio no contiene una referencia porque no invoca ningún servicio externo.

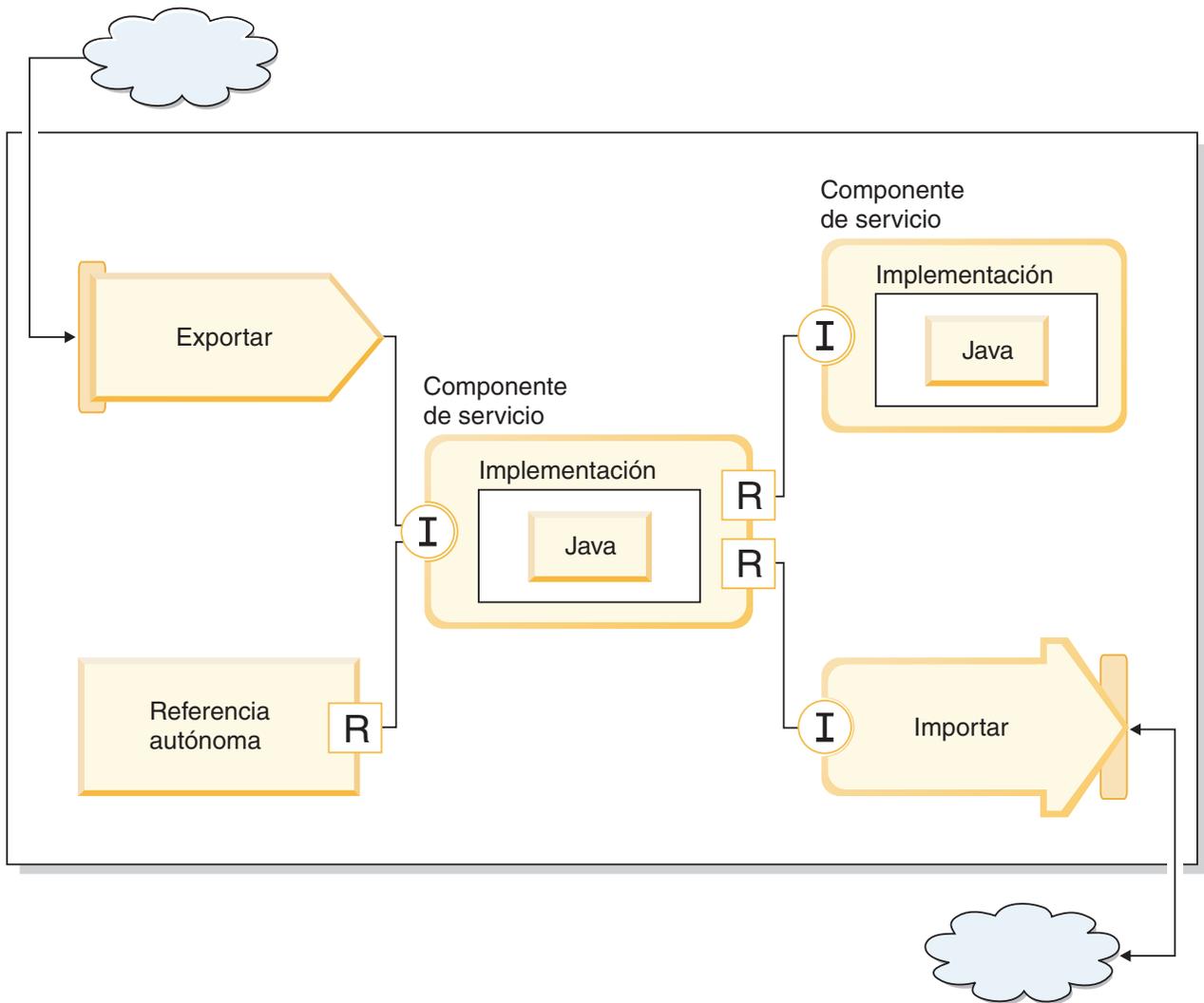
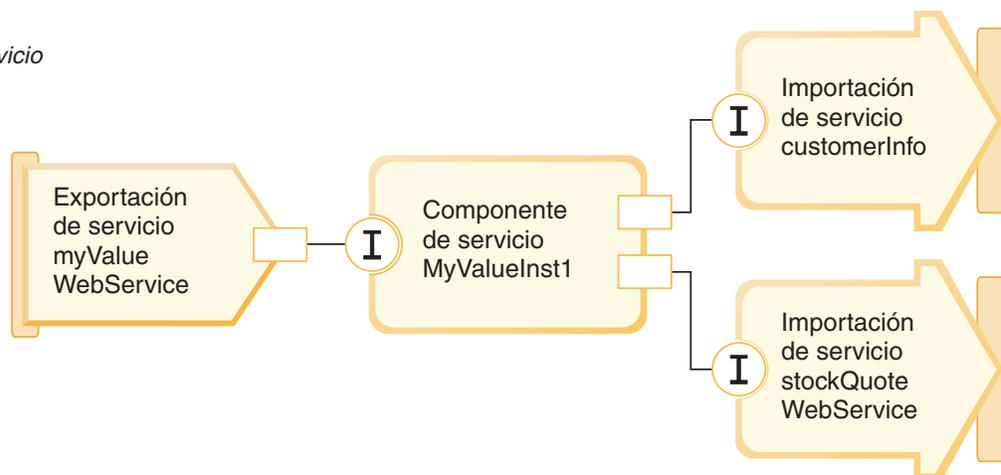


Figura 2. Estructura de un módulo

En Figura 3, el módulo contiene una exportación, dos importaciones y un componente de servicio que los utiliza. La conexión que se muestra enlaza los interfaces y las referencias.

Figura 3. Módulo de servicio



El módulo y los artefactos del módulo de mediación incluyen:

- Definición de módulo: define el módulo.
- Componentes de servicio: las definiciones de los servicios en el módulo. Un nombre de componente de servicio dentro de un módulo es exclusivo. Sin embargo, un componente de servicio puede tener un nombre de visualización arbitrario, que normalmente es un nombre más práctico para un usuario.
- Importaciones: definiciones de importaciones, que son llamadas a servicios externos a este módulo. Las importaciones tienen enlaces, que se describen en la sección Enlaces.
- Exportaciones: definiciones de exportaciones, que se utilizan para exponer componentes a los emisores que son externos a este módulo. Las exportaciones tienen enlaces, que se describen en la sección Enlaces.
- Referencias: referencias de un componente o otro en el módulo.
- Referencias autónomas: hace referencia a aplicaciones que no están definidas como componentes SCA (Service Component Architecture), por ejemplo, JavaServer Pages, que permiten a estas aplicaciones interactuar con componentes SCA (Service Component Architecture). Sólo puede haber un artefacto de referencias autónomas por módulo.
- Otros artefactos: estos artefactos incluyen los archivos WSDL, las clases Java, los archivosXSD, los procesos BPEL, etc.

## Componentes de servicio

Un componente de servicio configura una implementación de servicio. Un componente de servicio se presenta en un diagrama de bloques estándar.

Además de proporcionar una sintaxis y mecanismo coherentes para la invocación de servicios, SCA (Service Component Architecture) es la infraestructura de invocación que proporciona un método para que los desarrolladores encapsulen las implementaciones de servicio en componentes reutilizables. SCA permite a los desarrolladores definir interfaces, implementaciones y referencias de un modo independiente de la tecnología que se utilice. Este enfoque ofrece la oportunidad de enlazar los elementos a cualquier tecnología que elija. SCA separa la lógica empresarial de la infraestructura, de forma que los programadores de aplicaciones pueden centrarse en cómo resolver los problemas empresariales.

Un componente consta de una implementación, que está oculta cuando se utilizan las herramientas de WebSphere Integration Developer, una o más interfaces, que define sus entradas, salidas y errores y cero o más referencias. Una referencia identifica la interfaz de otro servicio o componente que requiere o consume este componente. Una interfaz se puede definir en uno de estos dos lenguajes: el tipo de puerto WSDL o Java. Una implementación soporta los estilos de interacción síncrono y asíncrono. Una implementación de componente puede estar en distintos lenguajes.

El tipo de interfaz recomendado es WSDL y nuestros ejemplos y guías de aprendizaje utilizan de forma coherente el tipo de interfaz WSDL. Sin embargo, se da soporte a una interfaz Java y se utiliza, la mayoría de las veces, cuando se importa un EJB de sesión sin estado. Debe desarrollar un componente Java en sentido descendente, es decir, definir un componente y añadir más adelante la implementación Java, debe seguir utilizando una interfaz WSDL. No puede combinar los componentes basados en interfaz WSDL con los componentes basados en la interfaz Java.

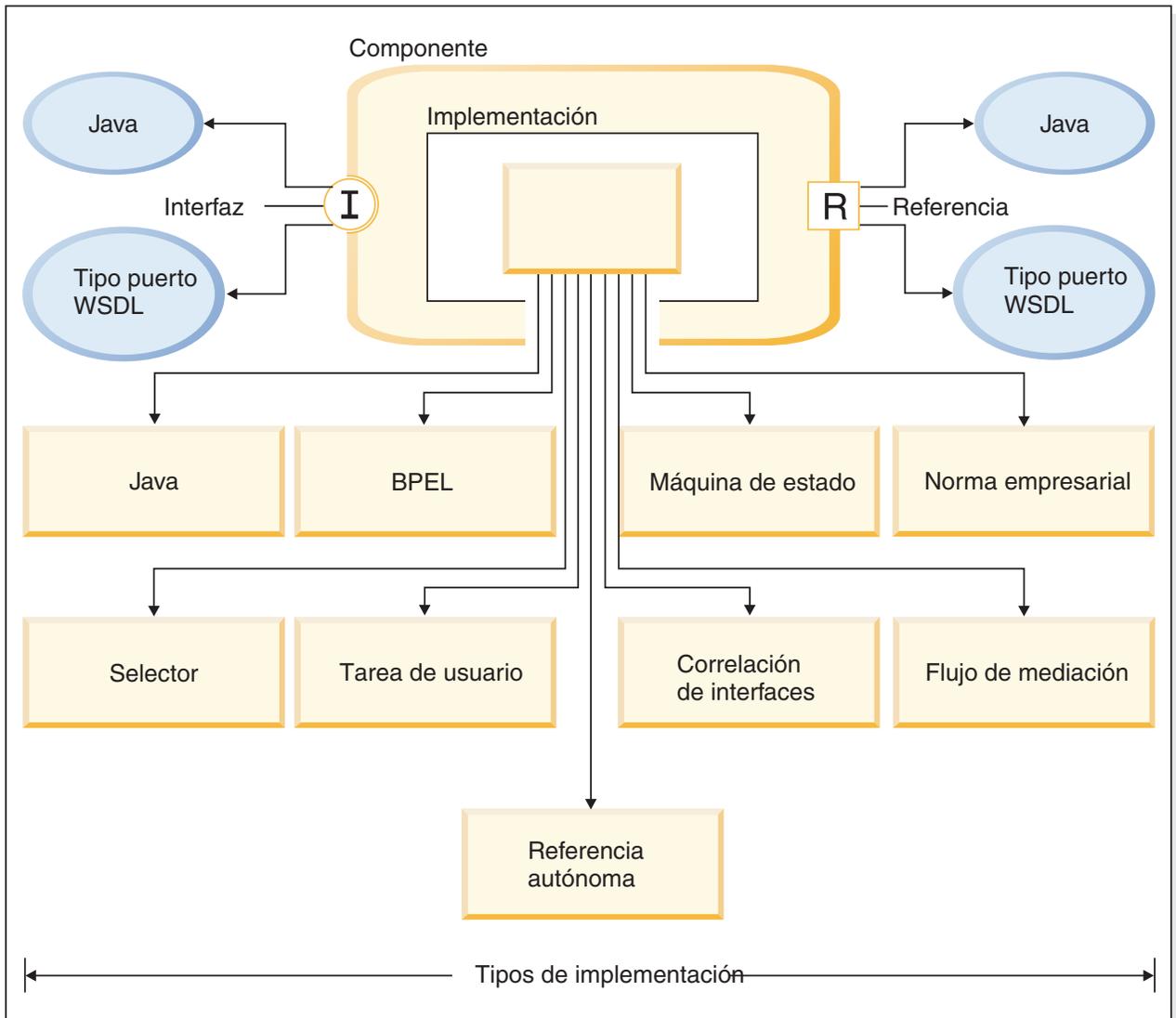


Figura 4. Estructura de un componente

En Figura 5, se tiene un componente en el centro. Su implementación, MyValueImpl, está en Java como su interfaz. Tiene dos referencias: otra interfaz Java y una interfaz WSDL.

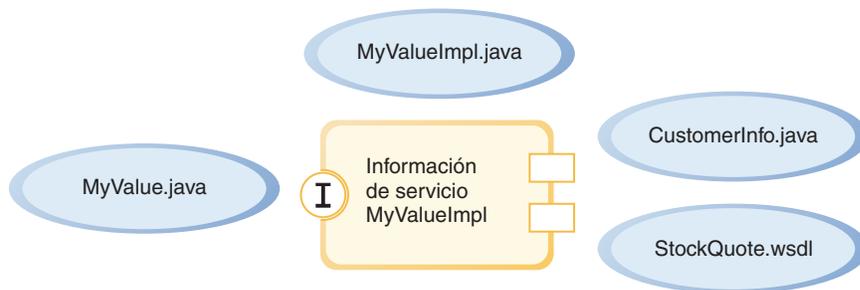


Figura 5. Estructura de componente de servicio

Cuando se trabaja con este componente, tal como se indica a continuación, sólo verá de forma efectiva el propio componente. Una referencia a este componente desde otro componente se revelará de forma visual mediante una línea en su interfaz. Una referencia de este componente se revelaría a través de una línea desde su punto de referencia en la interfaz de otro componente. Una referencia representa un servicio que consume este componente. A través de la denominación de una referencia y la única especificación de su interfaz, se permite al autor de la implementación del componente aplazar el enlace que hace referencia a un servicio real hasta más adelante. En ese momento, el especialista de la integración lo hará conectando la referencia a la interfaz de otro componente o importación. Este acoplamiento abierto, que permite el enlace aplazado y la reutilización de implementaciones, es una de las razones clave para utilizar el Service Component Architecture (SCA) de WebSphere Integration Developer.

Un componente también puede tener propiedades y calificadores. Un calificador es una directiva de calidad de servicio (QoS) en interfaces y referencias para el tiempo de ejecución.

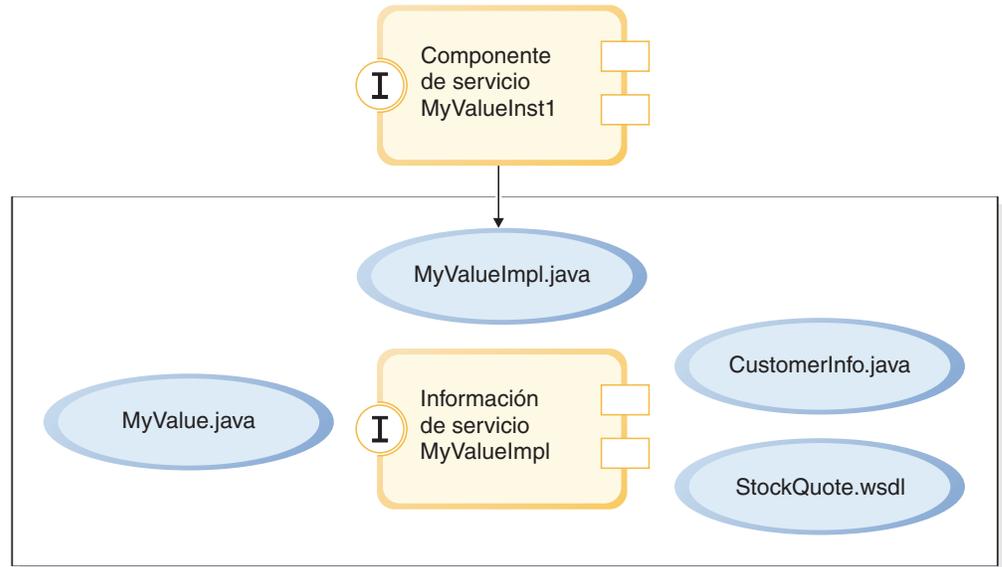


Figura 6. Instancia de un componente de servicio

Los tipos de implementación de componente de servicio son las implementaciones de los componentes de servicios.

WebSphere Integration Developer soporta los siguientes artefactos de implementación para WebSphere Process Server y WebSphere Enterprise Service Bus:

Tabla 1. Artefactos de implementación

| WebSphere Process Server      | WebSphere Enterprise Service Bus |
|-------------------------------|----------------------------------|
| Objetos Java                  | Objetos Java                     |
| Procesos de negocio           | Flujos de mediación              |
| Máquinas de estado de empresa |                                  |
| Normas empresariales          |                                  |
| Selectores                    |                                  |

Tabla 1. Artefactos de implementación (continuación)

| WebSphere Process Server  | WebSphere Enterprise Service Bus |
|---------------------------|----------------------------------|
| Tareas de usuario         |                                  |
| Correlaciones de interfaz |                                  |
| Flujos de mediación       |                                  |

**Nota:** Nota: las correlaciones de interfaces están en desuso como parte de WebSphere Process Server versión 7.0. Puede migrar los componentes de correlación de interfaces existentes a WebSphere Integration Developer para utilizar las funciones en el componente de flujo de mediación.

Las implementación del componente estándar de los servicios se describen en los temas de esta sección. Estas implementaciones aparecen en los servicios del editor de ensamblajes y/o dentro de procesos BPEL.

### Objetos Java

Se hace referencia a una implementación de un componente en Java como un objeto Java.

Una implementación común es un componente escrito en Java. A veces esta implementación recibe el apodo de "plain old Java object" o POJO. Por regla general, esta implementación tiene un tipo de interfaz WSDL, aunque esta implementación también podría tener una interfaz Java. Si hay varias interfaces especificadas, no puede mezclar las interfaces WSDL con las interfaces Java. Sin embargo, se puede "unir" una aplicación creada con un conjunto de interfaces WSDL a una aplicación con un conjunto de interfaces Java. Un ejemplo listado en la galería de ejemplos de la vista de bienvenida le muestra cómo.

Cuando se trabaja con un objeto Java, el código permanece oculto para el usuario dentro del contexto de los editores.

Un objeto Java se puede utilizar en un módulo de mediación. Se puede desplegar en un servidor WebSphere Process Server o un servidor WebSphere Enterprise Service Bus Server.

### Proceso BPEL

Un componente de *proceso BPEL* implementa un proceso de negocio.

Su lenguaje de implementación es el estándar del sector BPEL4WS (Business Process Execution Language for Web Services) y sus extensiones IBM. Un proceso BPEL implementa un servicio potencialmente de larga ejecución a través del uso de más servicios elementarios. Un proceso BPEL creado en el editor de procesos puede realizar lo siguiente:

- Describir la orquestación de otros servicios que utilizan los gráficos de flujo de control
- Utilizar variables para conservar el estado de proceso
- Utilizar un manejo de errores sofisticado mediante el manejo de errores
- Dar soporte a los sucesos asíncronos
- Correlacionar las peticiones de entrada con la instancia correcta de un proceso particular utilizando los conjuntos de correlaciones para marcar los datos de negocio dentro de la petición que identifican la instancia (por ejemplo, un ID de cliente).

- Proporcionar transacciones ampliadas a través del soporte de compensación sofisticado

Además de estos elementos BPEL estándar, WebSphere Integration Developer también amplía BPEL para incluir a las personas en un proceso con el soporte de *tarea de usuario*. Por ejemplo, esta ampliación podría añadir a un proceso el requisito que una persona apruebe un crédito.

El editor de procesos utiliza las representaciones visuales de las construcciones BPEL para generar su proceso de negocio rápida y fácilmente.

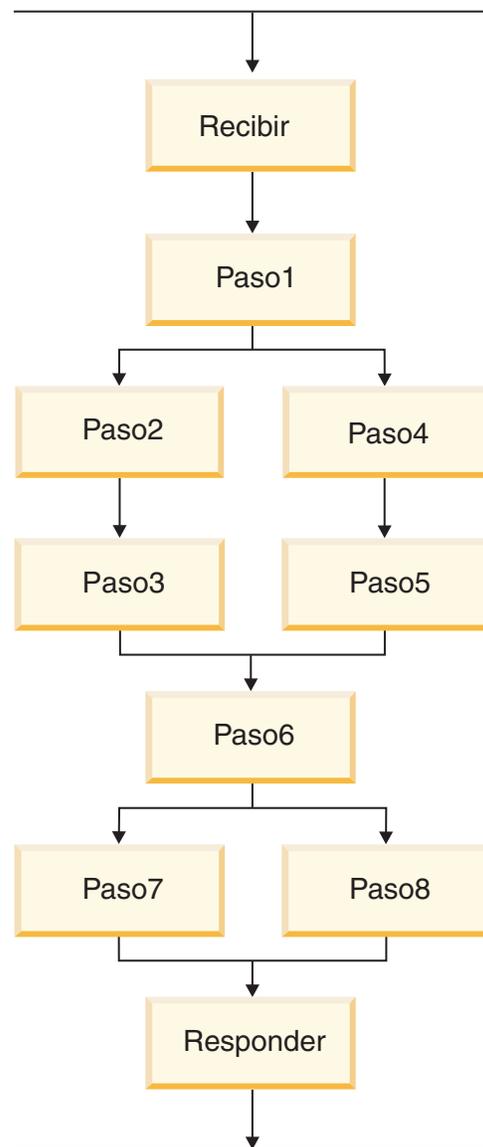


Figura 7. Proceso de negocio sencillo

Un proceso BPEL no se puede utilizar en un módulo de mediación. Sólo se puede desplegar en un servidor WebSphere Process Server.

## Máquinas de estado

Una máquina de estado es un método alternativo de crear un proceso de negocio. Una máquina de estado es apta para los procesos relacionados con la modificación de estados y no con el flujo de control. Un estado define qué puede realizar un artefacto en un momento dado. Una *máquina de estado* es una implementación de este conjunto de estados.

Las máquinas de estado son una forma común de visualizar un conjunto de estados interrelacionados en un proceso. Una máquina de estado familiar es un dispensador de bebidas. Se insertan algunas monedas en la máquina y junto a la bebida, que se dispensa al final, se obtiene el cambio exacto, ya que la máquina de estado divide mecánicamente las monedas que se deben devolver basándose en las monedas que se han insertado. En el diagrama que aparece a continuación, se muestra una máquina de estado típica creada por el editor de máquinas de estado. En la máquina de estado, se adquiere un artículo y se entrega a un cliente.

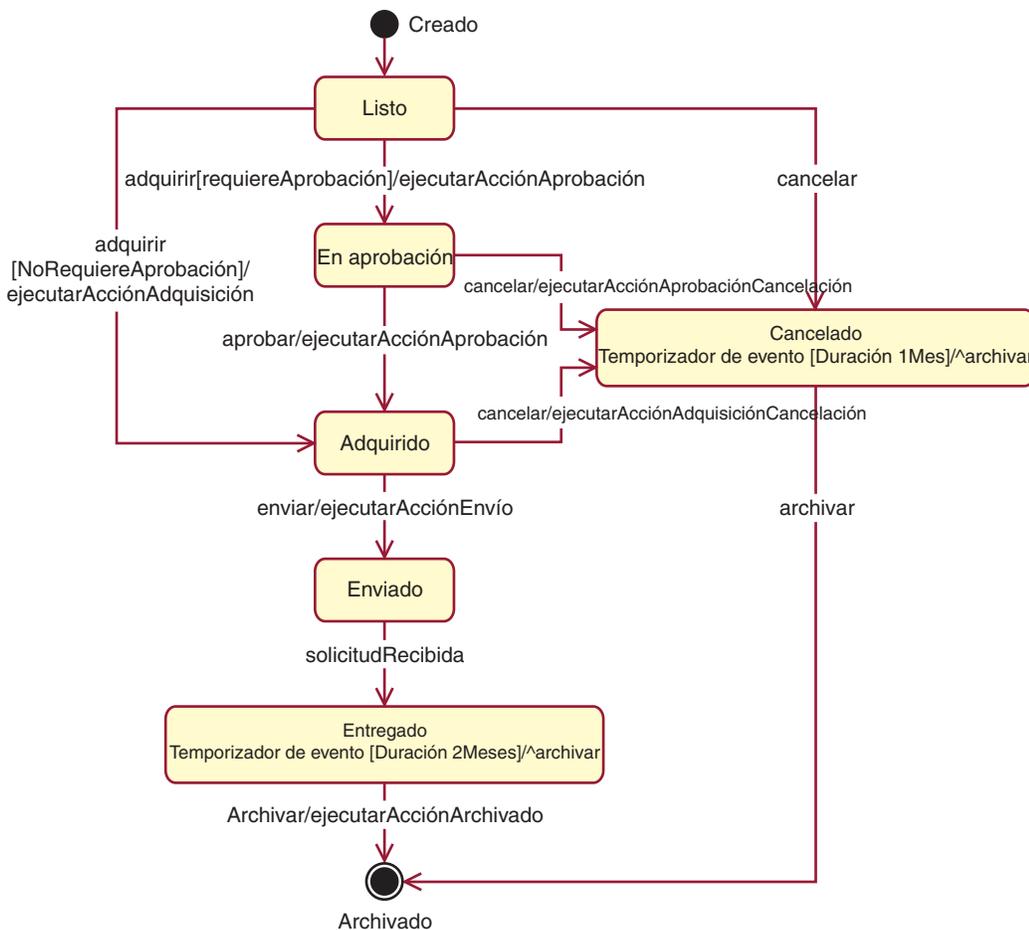


Figura 8. Máquina de estado

Una máquina de estado no se puede utilizar en un módulo de mediación. Sólo se puede desplegar en un WebSphere Process Server.

## Normas empresariales

Las normas empresariales complementan los procesos de negocio y las máquinas de estado. Si hay una condición con una variable, por ejemplo, una *norma empresarial* puede cambiar el valor de dicha variable durante la ejecución. Creada a

través de un lenguaje de programación visual, una norma empresarial toma una decisión basada en el contexto. La decisión puede ser simple o compleja. Las normas empresariales no son de proceso y las normas se pueden cambiar de forma independiente de una aplicación.

Las normas empresariales determinan el resultado de un proceso basado en un contexto. Las normas empresariales se utilizan en las situaciones de negocio cotidianas para tomar una decisión en un conjunto específico dado de circunstancias. Esta decisión puede requerir muchas normas para cubrir todas las circunstancias. Las normas empresariales dentro de un proceso de negocio permiten que las aplicaciones respondan rápidamente a condiciones empresariales cambiantes. En una empresa de seguros, por ejemplo, una norma empresarial para aprobar el seguro de un coche a un solicitante podría ser: *si el solicitante es hombre y mayor de 25 años, y la categoría del coche es deportivo y, durante los últimos 5 años tiene el seguro del coche con nuestra empresa, se aprueba la solicitud de seguro a una tarifa de 100 euros al mes.*

WebSphere Integration Developer ofrece una serie de enfoques para crear normas empresariales. Puede crear reglas if-then o tablas de decisiones, que conforman todas ellas el resultado del proceso. Estas reglas son independientes del propio proceso, lo que significa que puede cambiar las normas en cualquier momento sin tener que rehacer el proceso. Por ejemplo, basándose en la ubicación del negocio, es posible que tenga una norma que indique: *Si la fecha está entre el 26 de diciembre y el 1 de enero, se ofrece un descuento post-Navidad del 20%.* Sin embargo, si el descuento sigue siendo demasiado bajo, lo podría modificar en cualquier momento a un descuento de 40%.

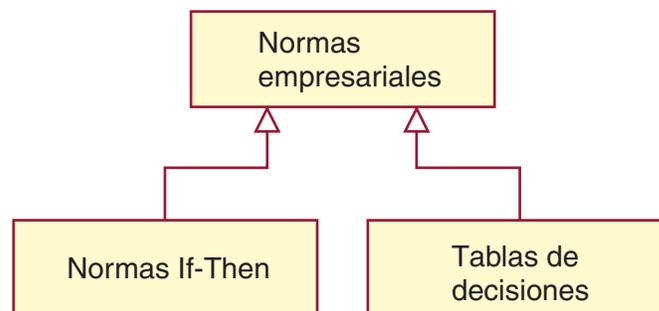


Figura 9. Tipos de normas empresariales

Las normas empresariales no se pueden utilizar en un módulo de mediación. Sólo se pueden desplegar en un servidor WebSphere Process Server.

## Selectores

Las aplicaciones integradas incluyen muchas formas de interactuar. Un *selector* se utiliza para direccionar una operación de una aplicación cliente a uno de los varios posibles componentes para la implementación.

El direccionamiento a un componente se basa en las fechas. Por ejemplo, aquí hay un direccionamiento basado en una fecha: *dos semanas antes de que comience el colegio, ofrece un precio especial de vuelta al colegio en los productos relacionados con el colegio.* Los negocios tienen muchos direccionamientos de este tipo basados en fechas. Un selector decide optar por una ruta en detrimento de otra durante la ejecución basándose en una fecha. Por ejemplo, si es la época justo antes de que

empieza el colegio, se llamará a la oferta de vuelta al colegio anterior. Sin embargo, si se trata de la época cuando acaba el colegio, podría haber una oferta para preparar a los niños para el verano.

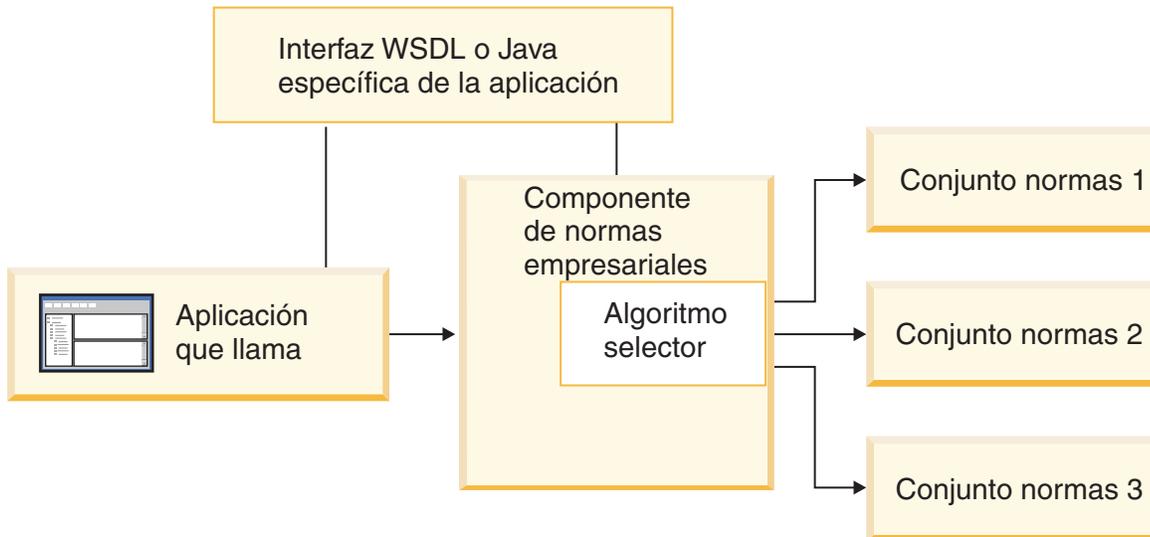


Figura 10. Selección de un conjunto de normas empresariales

Un selector no se puede utilizar en un módulo de mediación. Sólo se puede desplegar en un servidor WebSphere Process Server.

### Tarea de usuario

Un componente de *tarea de usuario* implementa una tarea realizada por un usuario. Representa la implicación de un usuario en un proceso de negocio.

De forma ocasional, los usuarios necesitan intervenir en un proceso de negocio. Por ejemplo, un cliente desea adquirir un artículo que está por encima de su límite de crédito. Una tarea de usuario le permite intervenir y alterar temporalmente una regla empresarial que impide al cliente realizar la compra. Una tarea de usuario puede tener atributos como, por ejemplo, establecer el propietario de la tarea, y proporcionar un proceso de escalada en el caso de que no esté disponible el usuario especificado. El componente de tarea de usuario reconoce la realidad que muchos procesos requieren la intervención de usuario para las tareas como la revisión, la investigación y la aprobación.

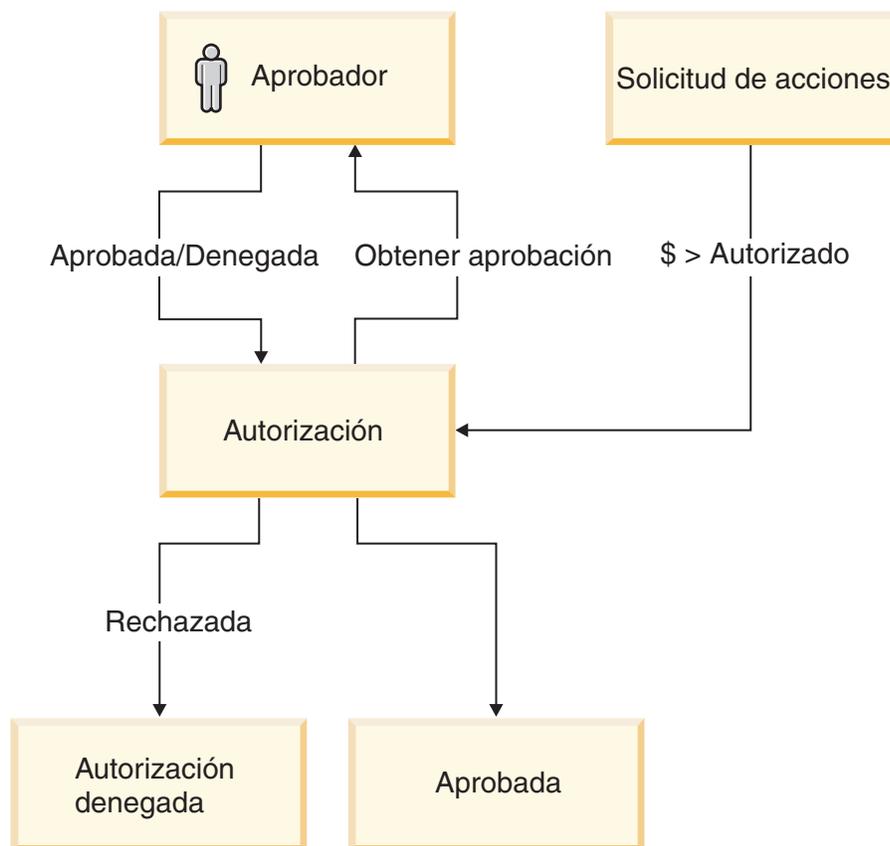


Figura 11. Componente de tarea de usuario

Una tarea de usuario no se puede utilizar en un módulo de mediación. Sólo se puede desplegar en un servidor WebSphere Process Server.

### Correlación de interfaces

Una *correlación de interfaces* resuelve las diferencias entre las interfaces de componentes que interactúan.

**Nota:** Las correlaciones de interfaces están en desuso a partir de WebSphere Process Server versión 7.0. Puede migrar los componentes de correlación de interfaces existentes a WebSphere Integration Developer para utilizar las funciones en el componente de flujo de mediación.

Las diferencias entre las interfaces de los componentes que deben interactuar entre sí son comunes. Estas diferencias surgen porque en WebSphere Integration Developer, a menudo, se ensamblan componentes que se crearon para aplicaciones diferentes. Reutilizarlos para crear una aplicación es uno de los puntos fuertes de WebSphere Integration Developer, ya que, de lo contrario, tendría que volver a codificar los componentes similares. Pero normalmente debe realizar algunos ajustes.

Por ejemplo, dos componentes pueden tener métodos que realizan básicamente la misma acción, pero tienen nombres diferentes como, por ejemplo, `getCredit` y `getCreditRating`. También pueden tener distintos nombres de operación y las operaciones pueden tener diferentes tipos de parámetro. Una correlación de interfaces correlaciona las operaciones y los parámetros de estos métodos, de forma que las diferencias se resuelven y los dos componentes pueden interactuar. Una

correlación de interfaces es como un puente entre las interfaces de dos componentes que les permiten conectarse a pesar de las diferencias.

Una correlación de interfaces existe de forma independiente de los componentes que utiliza, lo que significa que no es necesario modificar los propios componentes.

Una correlación de interfaces no se puede utilizar en un módulo de mediación. Sólo se puede desplegar en un servidor WebSphere Process Server.

### Flujo de mediación

La *mediación* es una forma de mediar o intervenir dinámicamente entre servicios. Un *flujo de mediación* implementa una mediación.

La mediación tiene varias funciones prácticas. Por ejemplo, puede utilizar la mediación cuando deba transformar datos de un servicio en un formato aceptable para un servicio posterior. El registro le permite registrar los mensajes de un servicio antes de que se envíen al siguiente servicio. El direccionamiento le permite direccionar los datos de un servicio en un servicio apropiado determinado por el flujo de mediación. Una mediación opera de forma independiente de los servicios a los que se conecta. Una mediación del editor de ensamblajes aparece como un componente de flujo de mediación entre exportaciones e importaciones.

En el diagrama siguiente, tres solicitantes de servicio o exportaciones envían sus datos de salida a la interfaz del componente de flujo de mediación. El componente de flujo de mediación direcciona los datos apropiados a dos proveedores de servicios o importaciones.

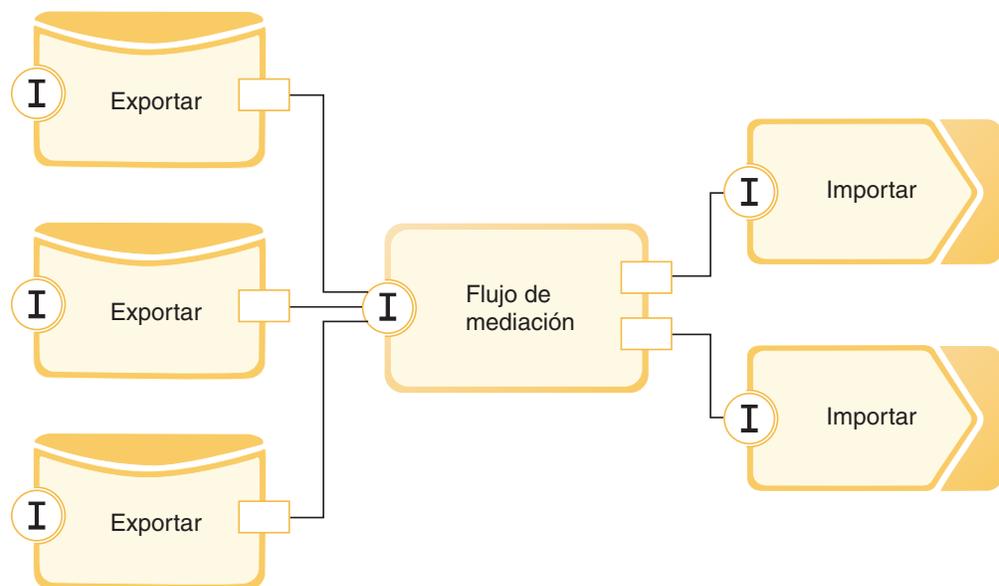


Figura 12. Componente de flujo de mediación entre tres solicitantes de servicios o exportaciones y dos proveedores de servicios o importaciones

Un flujo de mediación es una construcción con aspecto de flujo creada con el editor de flujos de mediación. La selección de un componente de flujo de mediación en el editor de ensamblajes, inicia el editor de flujos de mediación. En el editor de flujos de mediación, una operación de un servicio, el solicitante de servicio o la exportación, se correlaciona con la operación de otro servicio, el proveedor de servicios o la importación, junto con las funciones proporcionadas

por el editor de flujos de mediación. Estas funciones son denominadas *primitivos de mediación* y están conectadas en un flujo de mediación, tal como se muestra en el siguiente diagrama. Los primitivos de mediación son proporcionados por IBM o puede crear sus propios primitivos personalizados. Los primitivos de mediación pueden actuar tanto en el contenido, como en el contexto del mensaje, donde el contexto es información específica del enlace como, por ejemplo, cabeceras SOAP o JMS, o propiedades definidas por el usuario.

En el diagrama siguiente, una operación, `applyforLoan`, envía un mensaje, en primer lugar, a un primitivo de registro, `Log`, que registra el mensaje. `Log` envía el mensaje al primitivo `Filter`, que, en función del mensaje, direcciona el mensaje a una operación `processBusinessLoan` o a una operación `processPersonalLoan`.

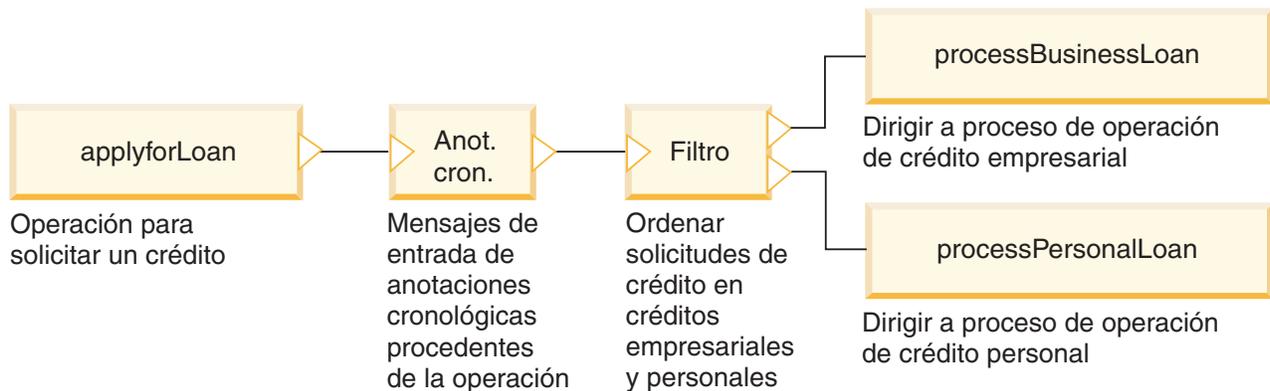


Figura 13. Flujo de mediación entre operaciones

Tal como se describe en la sección Módulos, los flujos de mediación pueden ser un módulo o un módulo de mediación. Ambos tipos de módulos pueden contener uno o más componentes de flujo de mediación además de cero o más componentes Java que aumentan el componente de flujo de mediación. Un módulo se puede desplegar en un servidor WebSphere Process Server. Un módulo de mediación se puede desplegar en un servidor WebSphere Process Server o un servidor WebSphere Enterprise Service Bus.

### Calificadores de servicio

Una aplicación comunica su calidad de servicio (QoS) necesaria en el entorno de ejecución especificando *calificadores de servicio*. Rigen la interacción entre un cliente de servicio y un servicio de destino.

Los calificadores se pueden especificar en las referencias, interfaces e implementaciones de componente de servicio. Puesto que la declaración de los valores QoS es externa a una implementación, puede cambiar estos valores sin cambiar la implementación, o los define de forma diferente cuando se utilizan varias instancias de la misma implementación en contextos diferentes.

Éstas son las categorías de los calificadores:

- Transacción: reglas para el tipo de transacción
- Sesión de actividad: reglas para unirse a la sesión activa
- Seguridad: reglas para permisos
- Fiabilidad asíncrona: reglas para la entrega de mensajes asíncronos

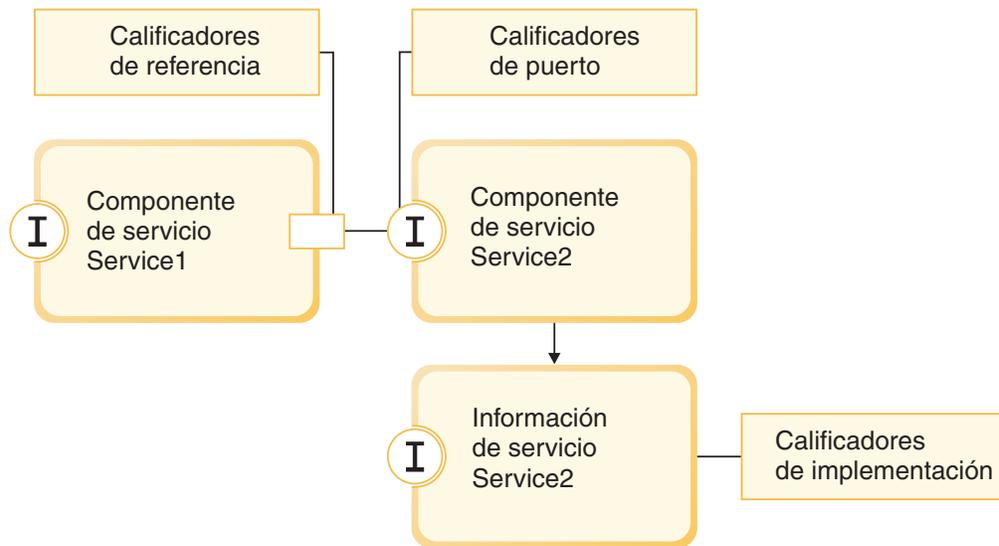


Figura 14. Calificadores

## Referencias autónomas

Las *referencias autónomas* son referencias a aplicaciones que no se han definido como componentes SCA (Service Component Architecture), por ejemplo, JavaServer Pages o servlets). Las referencias autónomas permiten a estas aplicaciones interactuar con componentes SCA (Service Component Architecture).

Las referencias autónomas no tienen ninguna interfaz ni implementación (ya que la implementación está fuera del ámbito del módulo). Un módulo no puede contener ninguna referencia autónoma, ni ningún artefacto de referencia autónoma. Las referencias autónomas tienen el valor práctico de permitirle utilizar las aplicaciones existentes junto con los componentes SCA (Service Component Architecture) creados en WebSphere Integration Developer.

Las referencias autónomas se pueden utilizar en un módulo de mediación. Se pueden desplegar en un servidor WebSphere Process Server o en un servidor WebSphere Enterprise Service Bus Server.

## Objetos de negocio

Los objetos de negocio complementan la arquitectura Service Component Architecture. Service Component Architecture define los servicios como componentes y la conectividad entre ellos. Los *objetos de negocio* definen el flujo de datos entre los componentes.

Cada componente pasa información como entrada y salida. Cuando se invoca un servicio, los datos de objeto se pasan como un documento XML con la codificación de literal de documento cuando se utiliza un tipo de puerto WSDL o como un objeto Java al utilizar una interfaz Java. Los objetos de datos son el formato preferido para los datos y metadatos en los servicios SCA (Service Component Architecture). Al igual que los componentes, los objetos de negocio separan el objeto de datos de su implementación. por ejemplo, un componente interactúa con los pedidos de compra mientras que el propio pedido de compra puede utilizar JDBC, EJB, etc., para realizar las actualizaciones en los datos. Los objetos de negocio permiten al desarrollador de integración centrarse en el trabajo con los

artefactos de negocio. De hecho, los objetos de datos de servicio son transparentes para el desarrollador de integración. Se definen mediante la petición de especificación Java (JSR) de objetos de datos de servicio.

En Figura 15, los objetos de negocio se pasan de un servicio externo a una exportación, de una exportación a un componente, de un componente a otro, de un componente a una importación y de una importación a un servicio. Las importaciones y exportaciones se describen en la sección Enlaces.

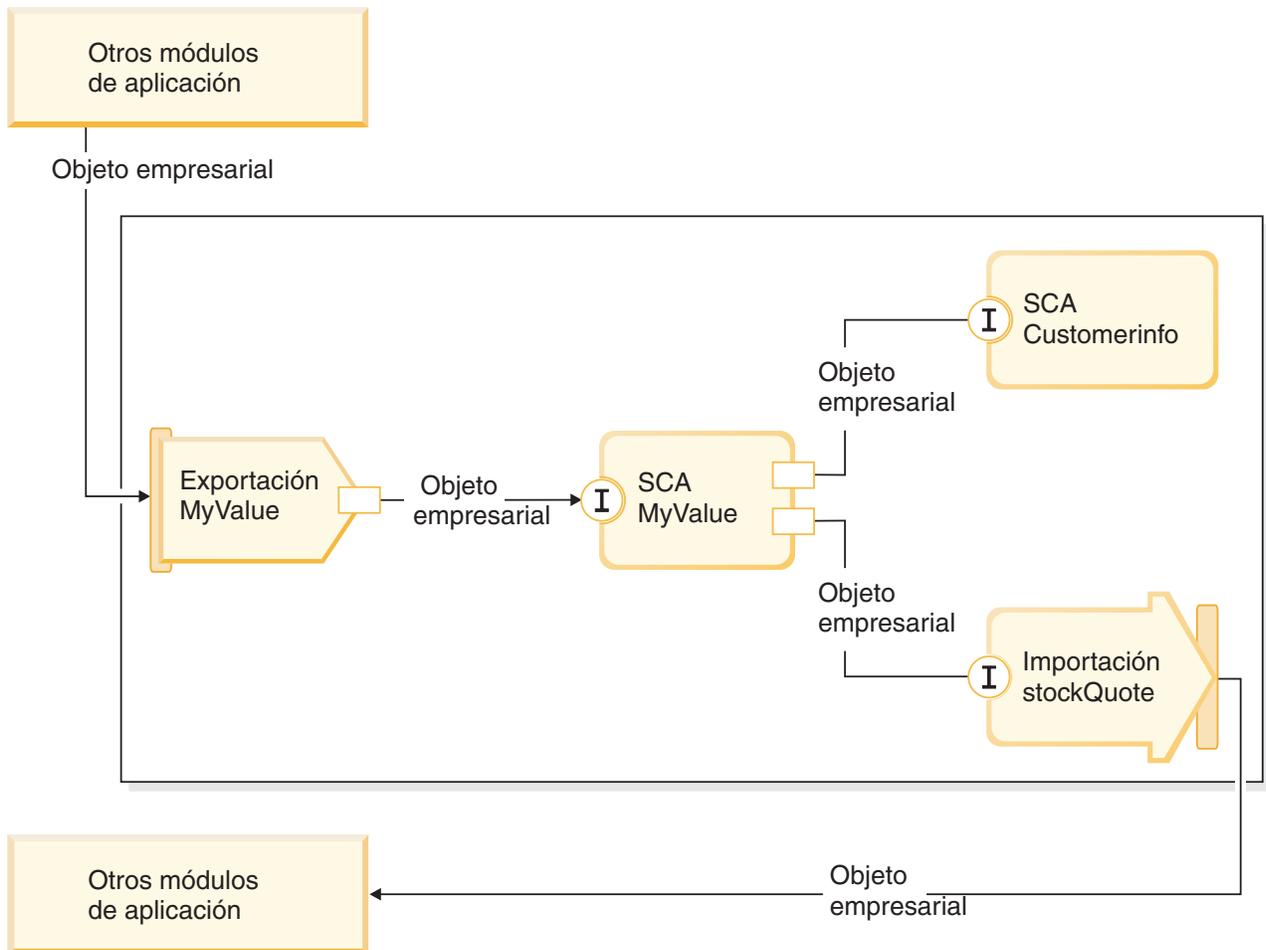


Figura 15. Objetos de negocio

## Objetos de negocio

La industria de software informático ha desarrollado varias infraestructuras y modelos de programación en los que los *objetos de empresa* proporcionan una representación natural de los datos empresariales para el proceso de aplicaciones.

Por lo general, los objetos de empresa:

- Se definen utilizando estándares del sector
- Correlacionan datos de forma transparente con tablas de base de datos o sistemas de información empresarial
- Soportan protocolos de invocación remota

- Proporcionan la base del modelo de programación de datos para programar aplicaciones

Desde la perspectiva de las herramientas, WebSphere Integration Developer proporciona a los desarrolladores un modelo de objetos de empresa común para representar distintos tipos de entidades empresariales desde distintos dominios. En el momento del desarrollo, este modelo permite a los desarrolladores definir objetos de empresa como definiciones de esquemas XML.

En tiempo de ejecución, los datos empresariales definidos por las definiciones de esquema XML se representan como objetos de empresa Java. En este modelo, los objetos de empresa se basan ligeramente en borradores iniciales de la especificación de objetos de datos de servicio (SDO) y proporcionan el conjunto completo de interfaces de aplicación de modelos de programación necesario para manipular datos empresariales.

## Definición de objetos empresariales

Los objetos empresariales se definen utilizando el editor de objetos empresariales en WebSphere Integration Developer. El editor de objetos empresariales almacena los objetos empresariales como definiciones de esquema XML.

Utilizar el esquema XML para definir objetos empresariales proporciona varias ventajas:

- Los esquemas XML proporciona un modelo de definición de datos basado en estándares y una base para la interoperatividad entre sistemas y aplicaciones heterogéneos dispares. Los esquemas XML se utilizan conjuntamente con Web Services Description Language (WSDL) para proporcionar contratos de interfaz basados en estándares entre componentes, aplicaciones y sistemas.
- Los esquemas XML definen un modelo de definición de datos ricos para representar datos empresariales. Este modelo incluye tipos complejos, tipos simples, tipos definidos por el usuario, herencia de tipos y cardinalidad, entre otras características.
- Los objetos empresariales pueden definirse mediante interfaces empresariales y datos definidos en Web Services Description Language, así como mediante esquemas XML de organizaciones estándar del sector, o de otros sistemas y aplicaciones. WebSphere Integration Developer puede importar estos objetos empresariales directamente.

WebSphere Integration Developer también proporciona soporte para descubrir datos empresariales en bases de datos y sistemas de información empresarial y generando, a continuación, la definición de objeto empresarial de esquema XML basado en estándares de esos datos empresariales. Los objetos empresariales generados de esta manera suelen conocerse como *objetos empresariales específicos de la aplicación* ya que imitan la estructura de los datos empresariales definidos en el sistema de información empresarial.

Cuando un proceso manipula datos de muchos sistemas de información distintos, puede resultar útil transformar las distintas representaciones de datos empresariales (por ejemplo, CustomerEIS1 y CustomerEIS2 u OrderEIS1 y OrderEIS2) en una única representación canónica (por ejemplo, Cliente o Pedido). A la representación canónica se la suele conocer como *objeto empresarial genérico*.

Las definiciones de objeto empresarial, concretamente para objetos empresariales genéricos, las utiliza frecuentemente más de una aplicación. Para dar soporte a esta

reutilización, WebSphere Integration Developer permite que se creen objetos empresariales en bibliotecas que entonces pueden asociarse con varios módulos de aplicación.

Los contratos para los servicios proporcionados y consumidos por un módulo de aplicación de Service Component Architecture (SCA), así como los contratos utilizados para crear los componentes dentro de un módulo de aplicación se definen mediante el Web Services Description Language. Un WSDL puede representar tanto las operaciones como los objetos empresariales, que un esquema XML define para representar los objetos empresariales, de un contrato.

## Trabajo con objetos empresariales

Service Component Architecture (SCA) proporciona la infraestructura para definir un módulo de aplicación, los servicios que proporciona, los servicios que consume y la composición de los componentes que proporcionan la lógica empresarial del módulo de aplicación. Los objetos empresariales juegan un papel importante en la aplicación, definiendo los datos empresariales que se utilizan para describir los contratos de servicio y componente y los datos empresariales que los componentes manipulan.

El siguiente diagrama representa un módulo de aplicación SCA e ilustra muchos de los lugares donde el desarrollador trabaja con objetos empresariales.

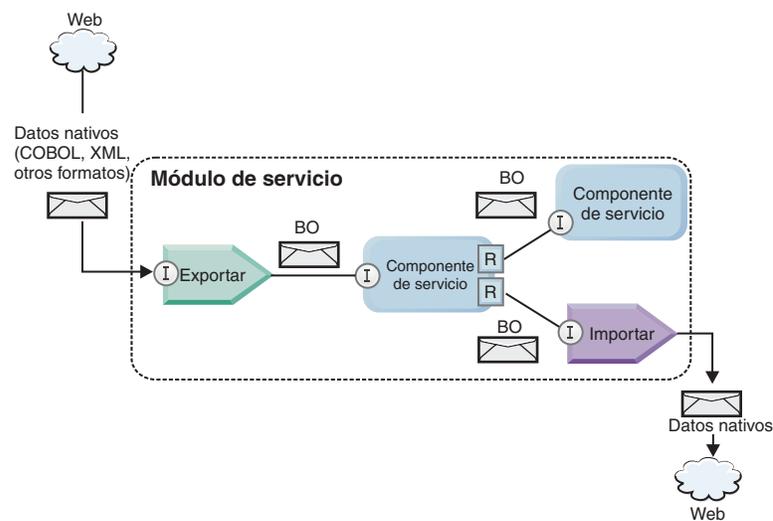


Figura 16. Los objetos empresariales representan los datos que fluyen entre los servicios de una aplicación

**Nota:** En este tema se describe cómo los módulos de aplicación SCA utilizan los objetos empresariales. Si está utilizando interfaces Java los módulos de aplicación SCA también pueden procesar objetos Java.

## Modelo de programación de objetos empresariales

El modelo de programación de objetos empresariales consta de un conjunto de interfaces Java que representan:

- La definición de objeto empresarial y los datos de la instancia
- Un conjunto de servicios que dan soporte a las operaciones en los objetos empresariales

Las definiciones de tipo de objeto empresarial están representadas por las interfaces `commonj.sdo.Type` y `commonj.sdo.Property`. El modelo de programación de objetos empresariales proporciona un conjunto de reglas para correlacionar la información de tipo complejo del esquema XML con la interfaz `Type` y cada uno de los elementos de la definición de tipo complejo con la interfaz `Property`.

Las instancias de objetos empresariales están representadas por la interfaz `commonj.sdo.DataObject`. El modelo de programación de objetos empresariales no tiene tipo, lo que significa que puede utilizarse la misma interfaz `commonj.sdo.DataObject` para representar distintas definiciones de objetos empresariales, tales como `Customer` y `Order`. La definición de qué propiedades pueden establecerse y recuperarse de cada objeto empresarial viene determinada por la información de tipo definida en el esquema XML asociado con cada objeto empresarial.

El comportamiento del modelo de programación de objetos empresariales se basa en la especificación `Service Data Object 2.1`. Para obtener información adicional, consulte la especificación `SDO 2.1 para Java`, las guías de aprendizaje y `javadocs` en la Web: <http://osoa.org/display/Main/Service+Data+Objects+Specifications>.

Los servicios de objetos empresariales dan soporte a diversas operaciones de ciclo de vida (tales como creación, igualdad, análisis y serialización) en los objetos empresariales.

Para conocer detalles sobre el modelo de programación de objetos empresariales, consulte `Programación del uso de servicios de objetos empresariales` y `Paquete com.ibm.websphere.bo`.

## **Enlaces, enlaces de datos y manejadores de datos**

Como se muestra en Figura 16 en la página 19, los datos empresariales que se utilizan para invocar servicios proporcionados por módulos de aplicación SCA se transforman en objetos empresariales para que los componentes SCA puedan manipular los datos empresariales. De forma similar, los objetos empresariales manipulados por componentes SCA se convierten al formato de datos que necesitan los servicios externos.

En algunos casos, como en el enlace de servicio web, el enlace utilizado para servicios de exportación e importación transforma automáticamente los datos al formato adecuado. En otros casos, como en el enlace JMS, los desarrolladores pueden proporcionar un enlace de datos o un manejador de datos que convierte los formatos no nativos a objetos empresariales representados por la interfaz `DataObject`.

Para obtener más información sobre el desarrollo de enlaces de datos y manejadores de datos, consulte `Manejadores de datos` y `Enlaces de datos`.

## **Componentes**

Los componentes SCA definen sus contratos de servicio de suministro y consumo mediante una combinación de Lenguaje de descripción de servicios web y esquemas XML. Los datos empresariales que SCA pasa entre componentes están representados como objetos empresariales mediante la interfaz `DataObject`. SCA verifica que estos tipos de objetos empresariales son compatibles con el contrato de interfaz definido por el componente a invocar.

Las abstracciones del modelo de programación para manipular objetos empresariales varían de componente a componente. El componente POJO y el primitivo Personalizado del componente de flujo de mediación proporciona la manipulación directa de los objetos empresariales al habilitar la programación Java directamente utilizando las interfaces y servicios de programación de objetos empresariales. La mayoría de componentes proporcionan abstracciones de nivel superior para manipular objetos empresariales, pero también proporcionan snippets Java para definir comportamiento personalizado en las interfaces y servicios de objetos empresariales.

Los objetos empresariales pueden transformarse utilizando la combinación de los componentes Mediación de flujo de interfaz y Correlación de objetos empresariales, o el componente de flujo de mediación y su primitivo Correlación XML. Estas prestaciones de transformación de objetos empresariales son de utilidad para convertir objetos empresariales específicos de la aplicación a objetos empresariales genéricos y viceversa.

## Objetos empresariales especiales

Los objetos de mensaje de servicio y los gráficos empresariales son dos tipos especializados de objetos empresariales que se utilizan para finalidades específicas de las aplicaciones.

### Objeto de mensaje de servicio

Un objeto de mensaje de servicio (SMO) es un objeto empresarial especializado que los componentes de flujo de mediación utilizan para representar la colección de datos asociados con una invocación de servicio.

Un SMO tiene una estructura fija de nivel superior que consta de cabeceras, contexto, cuerpo y adjuntos (si los hay).

- Las cabeceras llevan información relacionada con la invocación de servicio por un protocolo o enlace concreto. Algunos ejemplos son las cabeceras SOAP y las cabeceras JMS.
- Los datos de contexto llevan información lógica adicional asociada con la invocación mientras el componente de flujo de mediación los procesa. Esta información no suele formar parte de los datos de aplicación enviados o recibidos por clientes.
- El cuerpo del SMO lleva los datos empresariales de carga útil, que representan el mensaje de aplicación central o datos de invocación en forma de un objeto empresarial estándar.

El SMO también puede llevar datos de adjuntos para invocaciones del servicio web que utilicen SOAP con adjuntos.

Los flujos de mediación realizan tareas tales como direccionamiento de peticiones y transformación de datos, y el SMO proporciona la vista combinada de contenido de cabecera y carga útil en una sola estructura unificada.

### Gráfico empresarial

Un gráfico empresarial es un objeto empresarial especial utilizado para proporcionar soporte para la sincronización de datos en escenarios de integración.

Fíjese en este ejemplo en el que dos sistemas de información empresarial tienen una representación de un pedido específico. Cuando el pedido cambia en un

sistema, puede enviarse un mensaje al otro sistema para sincronizar los datos del pedido. Los gráficos empresariales dan soporte al concepto de enviar solamente la parte del pedido que ha cambiado al otro sistema y anotarlo con información de resumen de cambios para definir el tipo de cambio.

En este ejemplo, un gráfico empresarial Pedido indicaría al otro sistema que uno de los elementos de línea del pedido se ha suprimido y que se ha actualizado la propiedad de fecha de envío prevista del pedido.

Los gráficos empresariales pueden añadirse fácilmente a objetos empresariales existentes en WebSphere Integration Developer. Se encuentran con mayor frecuencia en escenarios en los que se utilizan adaptadores WebSphere y para soportar la migración de aplicaciones WebSphere InterChange Server.

## Modalidad de análisis de objetos empresariales

WebSphere Integration Developer proporciona una propiedad en módulos y bibliotecas que puede utilizar para configurar la modalidad de análisis de XML para objetos empresariales como diligente (*eager*) o lento (*lazy*).

- Si la opción se establece como *diligente*, las corrientes de bytes XML se analizan rápidamente para crear el objeto empresarial.
- Si la opción se establece como *lento*, el objeto empresarial se crea normalmente, pero el análisis de la corriente de bytes XML se aplaza y se analiza parcialmente sólo cuando se accede a las propiedades del objeto empresarial.

En cualquiera de las dos modalidades de análisis de XML, los datos no XML siempre se analizan rápidamente para crear el objeto empresarial.

### Ventajas del uso de la modalidad de análisis lento (*lazy*) frente al diligente (*eager*)

Algunas aplicaciones se benefician de la modalidad de análisis de XML lento mientras que otras ven una mejora del rendimiento con la modalidad de análisis diligente. Es recomendable probar su aplicación en ambas modalidades de análisis para determinar qué modalidad de ajusta mejor a las características específicas de la aplicación.

La sección siguiente proporcione directrices generales sobre los tipos de aplicaciones que se benefician de cada tipo de modalidad de análisis:

- Aplicaciones que se benefician de la modalidad de análisis XML lento  
Las aplicaciones que analizan secuencias de datos XML de gran tamaño son las que tienen más probabilidades de ver mejoras en el rendimiento cuando se utiliza la modalidad de análisis XML lento. Las ventajas de rendimiento aumentan a medida que aumenta el tamaño de la corriente de bytes XML y disminuye la cantidad de datos de la corriente de bytes a que accede la aplicación.

**Nota:** La modalidad de análisis lento de objetos empresariales se soporta en WebSphere Process Server versión 7.0.0.3 y versiones posteriores. No se soportan los módulos y módulos de mediación que incluyen componentes de flujo de mediación.

- Aplicaciones que se benefician de la modalidad de análisis rápida  
Las siguientes aplicaciones seguramente funcionarán mejor en modalidad de análisis diligente:
  - Aplicaciones que analizan corrientes de datos no XML
  - Aplicaciones que se crean mediante el servicio BOFactory

- Aplicaciones que analizan mensajes XML muy pequeños

## **Consideraciones sobre la migración de aplicaciones y el desarrollo**

Si va a configurar una aplicación que se desarrolló originalmente utilizando la modalidad de análisis diligente para que ahora utilice la modalidad de análisis lento, o si tiene pensado cambiar una aplicación entre las modalidades de análisis lento y diligente, tenga en cuenta las diferencias entre las modalidades y las consideraciones al cambiar de modalidad.

### **Manejo de errores**

Si la corriente de bytes XML que se analiza está mal formada, se producen excepciones de análisis.

- En modalidad de análisis XML diligente, estas excepciones se producen en cuanto se analiza el objeto empresarial desde la corriente XML entrante.
- Si se ha configurado la modalidad de análisis XML lento, las excepciones de análisis se producen de forma latente cuando se accede a las propiedades de objetos empresariales y se analiza la parte del XML que está mal formada.

Para tratar XML mal formado, seleccione una de las opciones siguientes:

- Despliegue un bus de servicio empresarial en los límites para validar el XML entrante
- Cree una lógica de detección de errores lenta en el punto en que se accede a las propiedades de objetos empresariales

### **Pilas de excepciones y mensajes**

Debido a que las modalidades de análisis XML diligente y lento tienen distintas implementaciones subyacentes, los seguimientos de pila emitidos por las interfaces de programación y servicios de objetos empresariales tienen el mismo nombre de clase de excepción, pero podrían no contener el mismo mensaje de excepción o conjunto de clases de excepción específicas de la implementación.

### **Formato de serialización XML**

La modalidad de análisis XML lento proporciona una optimización del rendimiento que intenta copiar XML no modificado desde la corriente de bytes de entrada a la corriente de bytes de salida al realizarse la serialización. El resultado es un aumento del rendimiento, pero el formato de serialización de la corriente de bytes XML de salida podría ser distinto si el objeto empresarial entero se actualiza en modalidad de análisis XML lento o si se estaba ejecutando en modalidad de análisis XML diligente.

Aunque es posible que el formato de serialización de XML no sea el equivalente exacto sintácticamente, el valor de semántica proporcionado por el objeto empresarial es equivalente independientemente de las modalidades de análisis, y se puede pasar XML con seguridad entre aplicaciones que se ejecuten en modalidades de análisis distintas con equivalencia semántica.

### **Validador de instancias de objetos empresariales**

El validador de instancias de modalidad de objeto empresarial de análisis XML lento proporciona una validación de objetos empresariales de mayor fidelidad, especialmente la validación de facetas de valores de propiedades. Debido a estas

mejoras, el validador de instancias de modalidad de análisis lento detecta problemas adicionales que no se detectan en la modalidad de análisis diligente y proporciona unos mensajes de error más detallados.

## **Correlaciones XML de la versión 602**

Los flujos de mediación desarrollados originalmente antes de WebSphere Integration Developer versión 6.1 pueden contener primitivos XSLT que utilicen una correlación o una hoja de estilos que no se puedan procesar directamente en la modalidad de análisis XML lento. Cuando se migra una aplicación para su uso en modalidad de análisis XML lento, el asistente de migración puede actualizar los archivos de correlación asociados con primitivos XSLT automáticamente para que se ejecuten en la nueva modalidad. No obstante, si un primitivo XSLT hace referencia directamente a una hoja de estilo que se ha editado manualmente, la hoja de estilo no se migra y no se puede procesar en la modalidad de análisis XML lento.

## **APIs privadas no publicadas**

Si una aplicación aprovecha las interfaces de programación de objetos empresariales no publicadas, privadas y específicas de la implementación, es muy probable que la aplicación no consiga compilar cuando se cambie de modalidad de análisis. En modalidad de análisis diligente, estas interfaces privadas suelen ser clases de implementación de objetos empresariales definidas por la infraestructura de modelado de Eclipse (EMF).

En todos los casos, se recomienda eliminar de la aplicación las API privadas.

## **APIs EMF de objeto de mensaje de servicio**

Un componente de mediación en WebSphere Process Server proporciona la capacidad de manipular el contenido de mensajes utilizando las clases e interfaces Java proporcionadas en el paquete `com.ibm.websphere.sibx.smobo`. En modalidad de análisis XML lento, las interfaces Java del paquete `com.ibm.websphere.sibx.smobo` pueden utilizarse igualmente, pero lo más probable es que fallen los métodos que hacen referencia directamente a clases e interfaces de la infraestructura de modelado de Eclipse (EMF) o que se heredan de interfaces EMF.

El `ServiceMessageObject` y su contenido no puede convertirse a objetos EMF en modalidad de análisis XML lento.

## **Servicio BOMode**

El servicio BOMode se utiliza para determinar si la modalidad de análisis XML que se procesa actualmente es diligente o lenta.

## **Migración**

Todas las aplicaciones anteriores a la versión 7.0.0.0 se ejecutan en modalidad de análisis XML diligente. Cuando se migran en tiempo de ejecución mediante las herramientas de migración en tiempo de ejecución BPM, continúan ejecutándose en modalidad de análisis XML diligente.

Para habilitar una aplicación anterior a la versión 7.0.0.0 para que se configure para utilizar la modalidad de análisis XML lento, primero utilice WebSphere

Integration Developer para migrar los artefactos de la aplicación. Después de la migración configurará la aplicación para que utilice el análisis XML lento.

Consulte el apartado Migración de artefactos de origen para obtener información acerca de la migración de artefactos en WebSphere Integration Developer, y el apartado Configuración de la modalidad de análisis de objetos empresariales y bibliotecas para obtener información acerca de cómo establecer el modo de análisis.

### Tipo de propiedad de objeto comercial QName

Debe modificar el código de la aplicación para que funcione con objetos empresariales que contengan la propiedad del tipo QName si desea que la aplicación que utilizaba análisis diligente utilice análisis lento. En la modalidad de análisis diligente, WebSphere Process Server utiliza la clase de Java `org.eclipse.emf.ecore.xml.type.internal.QName` para definir el valor de la propiedad de tipo QName. La modalidad de análisis lento utiliza la clase Java `javax.xml.namespace.QName` para establecer el valor para la propiedad de tipo QName. Modifique el código de la aplicación al cambiar la modalidad de diligente a lenta para un módulo, para ello, sustituya la referencia a la clase Java `org.eclipse.emf.ecore.xml.type.internal.QName` por `javax.xml.namespace.QName`.

---

## Relaciones

Una relación es una asociación entre dos o más entidades de datos, que suelen ser objetos empresariales. Las relaciones pueden utilizarse para transformar datos que son equivalentes entre objetos empresariales y otros datos pero que se representan de maneras distintas, o pueden utilizarse para trazar asociaciones entre distintos objetos encontrados en aplicaciones distintas. Pueden compartirse entre aplicaciones, entre soluciones e incluso entre productos.

El servicio de relaciones en WebSphere Process Server proporciona la infraestructura y las operaciones para gestionar relaciones. Al permitirle tratar con objetos empresariales independientemente de dónde residan, puede proporcionar un vista integral unificada a través de todas las aplicaciones de una empresa, y servir como bloque de creación para soluciones BPM. Dado que las relaciones son ampliables y gestionables, pueden utilizarse en soluciones de integración complejas.

### ¿Qué son las relaciones?

Una relación es una asociación entre objetos empresariales. Cada objeto empresarial de una relación se denomina *participante* en la relación. Cada participante en la relación se distingue de otros participantes según la función o *rol*, que desempeña en esa relación. Una relación contiene una lista de roles.

La *definición* de la relación describe cada rol y especifica cómo están relacionados los roles. También describe la "forma" global de la relación. Por ejemplo, este rol puede tener sólo un participante, pero este otro rol puede tener todos los participantes necesarios. Podría definir una relación *automóvil-propietario*, por ejemplo, donde un propietario podría poseer varios automóviles. Por ejemplo, una instancia podría tener los siguientes participantes para cada uno de estos roles:

- Automóvil (Ferrari)
- Propietario (John)

La definición de la relación es una plantilla para la *instancia* de la relación. La instancia es la instanciación en tiempo de ejecución de la relación. En el ejemplo anterior de *automóvil-propietario*, una instancia podría describir cualquiera de las siguientes asociaciones:

- John posee Ferrari
- Sara posee Mazda
- Bob posee Ferrari

Utilizar relaciones le libera de la necesidad de crear de forma personalizada una persistencia de seguimiento de relaciones dentro de la lógica empresarial. Para determinados escenarios, el servicio de relaciones realiza todo el trabajo automáticamente. Vea el ejemplo descrito en la sección sobre Relaciones de identidad.

## Escenarios

Este es un ejemplo típico de una situación en la que una solución de integración podría utilizar relaciones. Una gran corporación adquiere varias empresas, o unidades empresariales. Cada unidad empresarial utiliza distinto software para supervisar el personal y los portátiles. La empresa necesita un método para supervisar a los empleados y sus portátiles. Necesita una solución que le permita:

- Ver todos los empleados de las distintas unidades empresariales como si estuvieran en una base de datos
- Tener una única vista de todos los portátiles
- Permitir a los empleados iniciar sesión en el sistema y adquirir un portátil
- Acomodar los distintos sistemas de aplicaciones empresariales en las diversas unidades empresariales

Para conseguirlo, la empresa necesita un método para asegurar, por ejemplo, que John Smith y John A. Smith en distintas aplicaciones se perciben como el mismo empleado. Por ejemplo, necesitan una manera de consolidar una única entidad por diversos espacios de aplicación.

Los escenarios de relaciones más complejos implican crear procesos empresariales que trazan relaciones entre distintos objetos que se encuentran en varias aplicaciones. Con los escenarios de relaciones complejos, los objetos empresariales residen en la solución de integración y no en las aplicaciones. El servicio de relaciones proporciona una plataforma para gestionar relaciones de forma persistente. Antes del servicio de relaciones, tendría que crear su propio servicio de persistencia de objetos. Estos son dos ejemplos de escenario de relaciones complejos:

- Tiene un objeto empresarial car con un número VIN en una aplicación SAP y desea realizar el seguimiento del hecho que este automóvil es propiedad de otra persona. No obstante, la relación de propiedad es con alguien de la aplicación PeopleSoft. En este patrón de relaciones tiene dos soluciones y necesita construir un puente entre ellas.
- Una gran empresa de ventas desea poder supervisar la mercancía devuelta a cambio de efectivo o crédito. Existen dos aplicaciones distintas implicadas: un sistema de gestión de pedidos (OMS) para compras, y un sistema de gestión de devoluciones (RMS) para devoluciones. Los objetos empresariales residen en más de una aplicación y necesita una manera de mostrar las relaciones que existen entre ellos.

## Patrones de uso común

Los patrones de relaciones más comunes son los patrones de *equivalencia*. Estos se basan en las referencias cruzadas o la correlación. Existen dos tipos de relaciones que encajan con este patrón: *no identidad* e *identidad*.

- Las **relaciones de no identidad** establecen asociaciones entre objetos empresariales u otros datos de uno a muchos o de muchos a muchos. Para cada instancia de relación puede haber una o más instancias de cada participante. Un tipo de relación de no identidad es una relación de búsqueda estática. Un ejemplo es una relación en la que CA en una aplicación SAP está relacionado con California en una aplicación Siebel.

Las **relaciones de identidad** establecen asociaciones entre objetos empresariales u otros datos de uno a uno. Para cada instancia de relación sólo puede haber una instancia de cada participante. Las relaciones de identidad capturan referencias cruzadas entre objetos empresariales que son equivalentes semánticamente, pero que se identifican de forma distinta dentro de distintas aplicaciones. Cada participante en la relación está asociado a un objeto empresarial que tiene un valor (o una combinación de valores) que identifica al objeto de forma exclusiva. Las relaciones de identidad normalmente transforman los atributos clave de objetos empresariales, tales como números de ID y códigos de producto.

Por ejemplo, si tiene objetos empresariales car en aplicaciones SAP, PeopleSoft y Siebel, y desea crear una solución que los sincronice, normalmente necesitaría introducir lógica de sincronización de relaciones creada personalmente en seis correlaciones:

```
SAP -> generic
generic -> SAP
PeopleSoft-> generic
generic-> PeopleSoft
Siebel-> generic
generic-> Siebel
```

No obstante, si utiliza relaciones en la solución, el servicio de relaciones proporciona implementaciones de patrones prefabricadas que mantienen todas estas instancias de relaciones automáticamente.

## Herramientas para trabajar con relaciones

El *editor de relaciones* en WebSphere Integration Developer es la herramienta que utiliza para modelar y diseñar relaciones y roles de integración empresarial. Si desea información de fondo y de tareas sobre cómo crear relaciones y el uso del editor de relaciones, consulte el Information Center de WebSphere Integration Developer.

El *servicio de relaciones* es un servicio de infraestructura en WebSphere Process Server que mantiene relaciones y roles en el sistema y proporciona operaciones para la gestión de relaciones y roles.

El *gestor de relaciones* es la interfaz de administración para gestionar relaciones. Se accede a él mediante las páginas Gestor de relaciones de la consola de administración.

Las relaciones pueden invocarse de forma programática mediante las API de servicio de relaciones.

## Servicio de relaciones

El servicio de relaciones almacena datos de relaciones en tablas de relaciones, donde realiza el seguimiento de valores específicos de la aplicación a través de aplicaciones y a través de soluciones. El servicio de relaciones proporciona operaciones para la gestión de roles y relaciones.

### Cómo funcionan las relaciones

Las relaciones y los roles se definen utilizando la interfaz gráfica de la herramienta del editor de relaciones en WebSphere Integration Developer. El servicio de relaciones almacena los datos de correlación en tablas en la base de datos de relaciones en el origen de datos predeterminado que especifique al configurar el servicio de relaciones. Una tabla aparte (denominada a veces tabla participante) almacena información para cada participante en la relación. El servicio de relaciones utiliza estas tablas de relaciones para hacer el seguimiento de los valores específicos de la aplicación relacionados y propagar información actualizada por todas las soluciones.

Las relaciones, que son artefactos empresariales, se despliegan dentro de un proyecto o en una biblioteca compartida. En el primer despliegue, el servicio de relaciones llena los datos.

Durante la ejecución, cuando las correlaciones u otros componentes de WebSphere Process Server necesitan una instancia de relación, las instancias de las relaciones se actualizan o se recuperan, en función del escenario.

Los datos de la instancia de rol y de relación se pueden manipular mediante los siguientes métodos:

- Invocaciones de snippets Java del componente WebSphere Process Server por parte de las API del servicio de relaciones
- Transformaciones de relaciones en el servicio de correlación de objetos empresariales de WebSphere Process Server.
- La herramienta del gestor de relaciones

Si desea información de fondo y de tareas sobre cómo crear relaciones, identificar tipos de relación y el uso del editor de relaciones, consulte el Information Center de WebSphere Integration Developer.

## Gestor de relaciones

El gestor de relaciones es la interfaz de administración para gestionar relaciones. Se accede a él mediante las páginas Gestor de relaciones de la consola de administración.

El gestor de relaciones proporciona una interfaz gráfica de usuario para crear y manipular datos de roles y relaciones en tiempo de ejecución. Puede gestionar entidades de relaciones en todos los niveles: instancia de relación, instancia de rol y los niveles de datos de atributos y de datos de propiedades. Con el gestor de relaciones, puede hacer lo siguiente:

- Ver una lista de las relaciones del sistema e información detallada para relaciones individuales
- Gestionar instancias de relaciones:

- Consultar datos de relación para ver subconjuntos de datos de instancia
- Consultar datos de relación para ver subconjuntos de datos de instancia utilizando vistas de base de datos
- Ver una lista de instancias de relación que coincidan con una consulta de relación e información detallada sobre una instancia
- Editar los valores de propiedades de una instancia de relación
- Crear y suprimir instancias de relaciones
- Gestionar roles e instancias de roles:
  - Ver los detalles sobre un rol o una instancia de rol
  - Editar propiedades de instancias de rol
  - Crear y suprimir instancias de roles para una relación
  - Retrotraer datos de instancias de relaciones a un punto en el tiempo en que sepa que los datos fiables
- Importar datos de una relación estática existente al sistema, o exportar datos de una relación estática existente a un archivo RI o CSV
- Eliminar esquemas de relaciones y datos del repositorio cuando la aplicación que los utiliza se ha desinstalado

## Relaciones en entornos de despliegue de red

Pueden utilizarse relaciones en entornos de despliegue de red (Network Deployment (ND)) sin realizar configuración adicional.

En entornos Network Deployment (ND), las relaciones se instalan en un clúster de aplicación. Las relaciones son visibles entonces dentro del clúster, y todos los servidores del clúster tienen acceso a los datos de instancia almacenados en la base de datos de relaciones. La capacidad de ejecutar el servicio de relaciones en un entorno ND la hace escalable y de alta disponibilidad.

El gestor de relaciones permite que se gestionen las relaciones entre distintos clústeres a través de una interfaz administrativa centralizada. El gestor de relaciones se conecta a un servidor en un clúster seleccionando su MBean de relación.

## APIs de servicio de relaciones

Las relaciones pueden invocarse de forma programática mediante las API de servicio de relaciones, dentro o fuera de correlaciones de objetos empresariales.

Hay tres tipos de API disponibles:

- APIs de manipulación de instancia de relación (incluido crear, actualizar, suprimir datos de la instancia directamente)
- APIs de soporte de patrones de relación (incluidas `correlate()`, `correlateforeignKeyLookup`)
- Patrones de búsqueda de relaciones (APIs de búsqueda)

---

## El bus de integración de servicio en WebSphere Process Server

WebSphere Process Server soporta la integración de servicios de aplicaciones, incluyendo las mismas posibilidades que WebSphere Enterprise Service Bus.

## Conexión de servicios a través de un bus de servicio de empresa

Con un Enterprise Service Bus (ESB), puede maximizar la flexibilidad de una SOA. Los participantes de una interacción de servicio se conectan al ESB en lugar de conectarse directamente entre sí.

Cuando un solicitante de servicios se conecta al ESB, el ESB es responsable de entregar sus peticiones, mediante mensajes, a un proveedor de servicios que ofrezca la función y la calidad de servicio necesarias. El ESB facilita las interacciones entre solicitante y proveedor, y direcciona los protocolos no coincidentes, los patrones de interacción o las posibilidades de servicio. Un ESB también puede habilitar o mejorar la supervisión y la gestión. El ESB proporciona las características de virtualización y gestión que implementan y amplían las posibilidades principales de la SOA.

El ESB resume las características siguientes:

### Ubicación e identidad

No es necesario que los participantes conozcan la ubicación o la identidad de los demás participantes. Por ejemplo, no es necesario que los solicitantes sepan que una petición puede recibir respuesta de varios proveedores; es posible añadir o eliminar proveedores sin que ello afecte de ningún modo.

### Protocolo de interacción

No es necesario que los participantes compartan el mismo protocolo de comunicación ni el mismo estilo de interacción. Por ejemplo, una petición expresada como SOAP a través de HTTP puede recibir respuesta de un proveedor que sólo utiliza SOAP a través de JMS (Java Message Service).

### Interfaz

No es necesario que los solicitantes y los proveedores utilicen una interfaz común. Un ESB concilia las diferencias transformando los mensajes de petición y de respuesta en el formato esperado por el proveedor.

### No es necesario que los solicitantes y los proveedores utilicen una interfaz común

Un ESB concilia las diferencias transformando los mensajes de respuesta en el formato esperado por el proveedor.

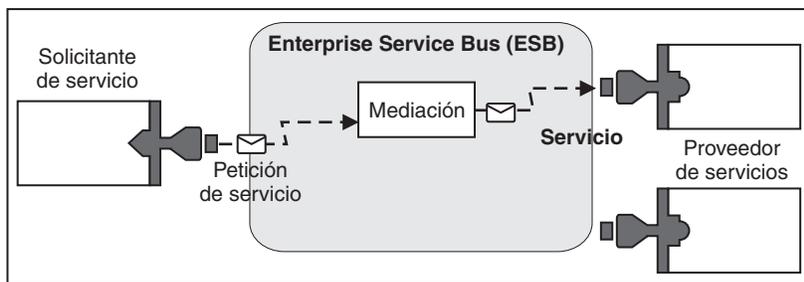
### Calidades de servicio (interacción)

Los participantes, o administradores de sistemas, declaran sus requisitos de calidad de servicio, incluida la autorización de peticiones, el cifrado y descifrado del contenido de los mensajes, la auditoría automática de interacciones de servicios, y el modo en que deben direccionarse las peticiones (por ejemplo, optimizándolas en cuanto a velocidad o coste).

Interponer el ESB entre los participantes permite modular su interacción a través de un constructor lógico que se conoce como *mediación*. Las mediaciones operan en mensajes que se encuentran de camino entre los solicitantes y los proveedores. Por ejemplo, pueden utilizarse mediaciones para encontrar servicios con características específicas que pida un solicitante, o para resolver las diferencias de interfaz entre solicitantes y proveedores. En el caso de las interacciones complejas, pueden encadenarse mediaciones secuencialmente.

Un Enterprise Service Bus con mediaciones realiza las acciones siguientes entre el solicitante y el servicio:

- *Direccionar* mensajes entre servicios. Un Enterprise Service Bus ofrece una infraestructura de comunicación común que puede utilizarse para conectar servicios y, por consiguiente, las funciones de empresa funciones que representan, sin que los programadores deban escribir ni realizar el mantenimiento de una lógica de conexión compleja.
- *Convertir* protocolos de transporte entre solicitante y servicio. Un Enterprise Service Bus proporciona un método coherente y basado en estándares para integrar funciones de empresa que utilizan distintos estándares de TI. Esto permite la integración de funciones de empresa que generalmente no pueden comunicarse, como por ejemplo, para conectar aplicaciones de silos de departamento o para permitir que aplicaciones de distintas compañías participen en interacciones de servicios.
- *Transformar* formatos de mensajes entre solicitante y servicio. Un Enterprise Service Bus permite que las funciones de empresa intercambien información en distintos formatos; el bus garantiza que la información que se entrega a la función de empresa tenga el formato necesario para la aplicación.
- *Gestionar* sucesos de empresa de distintas fuentes. Un Enterprise Service Bus da soporte a interacciones basadas en sucesos además de los intercambios de mensajes para gestionar solicitudes de servicios.



*Figura 17. Bus de servicio de empresa.* El bus de servicio de empresa direcciona los mensajes entre las aplicaciones, que son solicitantes o proveedores de servicios. El bus convierte los protocolos de transporte y transforma los formatos de mensajes entre los solicitantes y los proveedores. En esta figura, cada aplicación utiliza un protocolo distinto (representado por las distintas formas geométricas de los conectores) y utiliza formatos de mensajes distintos.

El uso de Enterprise Service Bus le permite centrarse en los aspectos principales de la empresa, y no en los sistemas informáticos. Puede modificar o añadir servicios cuando sea necesario; por ejemplo, para responder a cambios en los requisitos de la empresa, para añadir más capacidad de servicio o para añadir nuevas posibilidades. Los cambios necesarios se realizan configurando de nuevo el bus, sin que ello afecte en absoluto, o con un efecto insignificante, a los servicios y las aplicaciones existentes que utilizan dicho bus.

## Infraestructura de mensajería de Enterprise Service Bus

WebSphere Process Server incluye posibilidades de Enterprise Service Bus. WebSphere Process Server da soporte a la integración de tecnologías orientadas a servicios, orientadas a mensajes y controladas por sucesos para proporcionar una infraestructura de mensajería basada en estándares en un Enterprise Service Bus integrado.

Las posibilidades de Enterprise Service que puede utilizar para sus aplicaciones de empresa proporcionan no sólo una capa de transporte, sino soporte de mediación para facilitar las interacciones de servicio. Enterprise Service Bus se crea en función de estándares abiertos y una arquitectura orientada a servicios (SOA). Se basa en la

potente infraestructura de Java EE y los servicios de plataforma asociados proporcionados por IBM WebSphere Application Server Network Deployment.

WebSphere Process Server está basado en la misma tecnología que está disponible en IBM WebSphere Enterprise Service Bus. Esta posibilidad forma parte de la funcionalidad subyacente de WebSphere Process Server, y no es necesario tener una licencia adicional de WebSphere Enterprise Service Bus para aprovechar estas posibilidades.

No obstante, puede desplegar licencias autónomas adicionales de WebSphere Enterprise Service Bus en su empresa para ampliar el alcance de la conectividad de las soluciones de integración de procesos basadas en WebSphere Process Server. Por ejemplo, WebSphere Enterprise Service Bus se puede instalar más cerca de una aplicación SAP para alojar IBM WebSphere Adapter for SAP y para transformar los mensajes SAP antes de enviar la información a través de la red a un proceso de negocio coreografiado por WebSphere Process Server.

Puede desplegar WebSphere Enterprise Service Bus en su empresa para ampliar el alcance de la conectividad de las soluciones de integración de procesos basadas en instalaciones separadas de WebSphere Process Server o de otras soluciones de integración como parte de un ESB federado. Por ejemplo, WebSphere Enterprise Service Bus se puede instalar más cerca de una aplicación SAP para alojar IBM WebSphere Adapter for SAP y para transformar los mensajes SAP antes de enviar la información a través de la red a un proceso de negocio coreografiado por WebSphere Process Server.

### **Sistemas principales de destino de cola o de mensajería**

Un sistema principal de destino de cola o de mensajería proporciona la función de mensajería dentro de un servidor. Un servidor se convierte en el sistema principal de destino de mensajería cuando se configura como el destino de mensajería.

Un motor de mensajería se ejecuta dentro de un servidor. El motor de mensajería proporciona funciones de mensajería y un punto de conexión con el que las aplicaciones se conectan al bus. La comunicación asíncrona de SCA (Service Component Architecture), las importaciones y exportaciones JMS, así como el proceso interno asíncrono utilizan colas de mensajes en el motor de mensajería.

El entorno de despliegue conecta el origen del mensaje con el destino del mensaje a través del bus si los módulos de aplicación están desplegados. Conocer el origen del mensaje y el destino del mensaje ayuda a determinar qué tipo de entorno de despliegue se necesita.

Las aplicaciones pueden almacenar datos persistentes en un almacén de datos, que es un conjunto de tablas de una base de datos o un esquema, o en un almacén de archivos. El motor de mensajería utiliza una instancia de un origen de datos JDBC para interactuar con esa base de datos.

Configure el sistema principal de destino de mensajería cuando defina el entorno de despliegue utilizando **Servidor** en la consola administrativa o designe el servidor como el sistema principal de destino durante la instalación de software.

#### **Almacenes de datos:**

Cada motor de mensajería puede utilizar un almacén de datos, que es un conjunto de tablas en una base de datos o esquema que almacena los datos persistentes.

Todas las tablas del almacén de datos se mantienen en el mismo esquema de base de datos. Puede crear cada almacén de datos en una base de datos diferente. O bien, puede crear varios almacenes de datos en la misma base de datos, donde cada almacén de datos utiliza un esquema diferente.

Un motor de mensajería utiliza una instancia de un almacén de datos JDBC para interactuar con la base de datos que contiene el almacén de datos del motor de mensajería.

## Orígenes de datos

Los orígenes de datos proporcionan un enlace entre las aplicaciones y las bases de datos relacionales.

Las aplicaciones utilizan un origen de datos para obtener conexiones con una base de datos relacional. Un origen de datos es análogo a la fábrica de conexiones de JCA (Java EE Connector Architecture), que proporciona conectividad a otros tipos de EIS (Enterprise Information Systems).

Un origen de datos se asocia a un proveedor JDBC, que suministra las clases de implementación de controlador que son necesarias para la conectividad JDBC con un tipo de base de datos específico. Los componentes de aplicación realizan transacciones directamente con el origen de datos para obtener instancias de conexión con una base de datos. La agrupación de conexiones que corresponde a cada origen de datos proporciona gestión de conexiones.

Puede crear varios orígenes de datos con valores distintos y asociarlos al mismo proveedor JDBC. Por ejemplo, puede utilizar varios orígenes de datos para acceder a distintas bases de datos dentro de la misma aplicación de base de datos. WebSphere Process Server requiere que los proveedores JDBC implementen una de las siguientes interfaces de origen de datos, o ambas interfaces, definidas por Sun Microsystems. Estas interfaces permiten que la aplicación se ejecute en un protocolo de transacción de una sola fase o de dos fases.

**Nota:** Los orígenes de datos de Business Process Choreographer se crean utilizando las herramientas de configuración de Business Process Choreographer. Consulte Configuración de Business Process Choreographer.

- `ConnectionPoolDataSource` - un origen de datos que da soporte a la participación de aplicaciones en las transacciones locales y globales, a excepción de las transacciones de compromiso de dos fases. Cuando un origen de datos de agrupación de conexiones está implicado en una transacción global, el gestor de transacciones no proporciona la función de recuperación de transacciones. La aplicación es responsable de proporcionar el proceso de recuperación de copia de seguridad si hay varios gestores de recursos implicados.
- `XADataSource` - un origen de datos que da soporte a la participación de aplicaciones en cualquier entorno de transacciones de una sola fase o de dos fases. Cuando este origen de datos está implicado en una transacción global, el gestor de transacciones de WebSphere Application Server proporciona la función de recuperación de transacciones.

Las tablas siguientes proporcionan ejemplos de configuraciones de un entorno de despliegue típico autónomo y de un entorno de despliegue típico:

Tabla 2. Configuración de entorno de despliegue autónomo típico

| Origen de datos                                      | Componente                 | Ámbito   | Nombre JNDI  |
|--|----------------------------|----------|--|
| Origen de datos WBI                                  | CommonDB                   | Nodo     | jdbc/WPSDB   |
| Origen de datos ME de bus de aplicación SCA          | Motor de mensajería SCA    | Servidor | jdbc/com.ibm.ws.sib/nlNode01.server1-SCA.APPLICATION.localhostNode01Cell.Bus |
| Origen de datos de Business Process Choreographer    | BPC                        | Servidor | jdbc/BPEDB   |
| Origen de datos ME de Business Process Choreographer | Motor de mensajería de BPC | Servidor | jdbc/com.ibm.ws.sib/nlNode01.server1-BPC.localhostNode01Cell.Bus             |
| Suceso   | CEI                        | Servidor | jdbc/cei   |
| Origen de datos CEI ME                               | Motor de mensajería CEI    | Servidor | jdbc/com.ibm.ws.sib/nlNode01.server1-CEI.nombre_célula.BUS                   |

Tabla 3. Configuración de entorno de despliegue típico

| Origen de datos                                      | Componente                 | Ámbito  | Nombre JNDI  |
|--|----------------------------|---------|--|
| Origen de datos WBI                                  | CommonDB                   | Célula  | jdbc/WPSDB   |
| Origen de datos ME de bus de aplicación SCA          | Motor de mensajería SCA    | Clúster | jdbc/com.ibm.ws.sib/clusterone-SCA.APPLICATION.enduranceTestCell01.Bus |
| Origen de datos de Business Process Choreographer    | BPC                        | Clúster | jdbc/BPEDB   |
| Origen de datos ME de Business Process Choreographer | Motor de mensajería de BPC | Clúster | jdbc/com.ibm.ws.sib/clusterone-BPC.enduranceTestCell01.Bus             |
| Suceso   | CEI                        | Clúster | jdbc/cei   |
| Origen de datos CEI ME                               | Motor de mensajería CEI    | Clúster | jdbc/com.ibm.ws.sib/clusterone-CEI.nombre_célula.BUS                   |

Para obtener más información sobre orígenes de datos, consulte “Orígenes de datos” en el centro de información de WebSphere Application Server.

#### Proveedores de JDBC:

Puede utilizar proveedores JDBC para interactuar con aplicaciones con bases de datos relacionales.

Las aplicaciones utilizan proveedores JDBC para interactuar con las bases de datos relacionales. Un proveedor de JDBC suministra la clase de implementación específica de controlador JDBC para acceder a un tipo específico de base de datos. Para crear una agrupación de conexiones en dicha base de datos, asocie un origen

de datos al proveedor de JDBC. El proveedor de JDBC y los objetos de origen de datos funcionan, conjuntamente, de forma equivalente a la fábrica de conexiones de JCA (Java EE Connector Architecture), que proporciona conectividad con una base de datos no relacional.

Consulte los ejemplos de la configuración de entorno autónomo típico y de la configuración de entorno de despliegue típico en el tema anterior.

Para obtener más información sobre los proveedores JDBC, consulte “Proveedores de JDBC” en el Centro de información de WebSphere Application Server.

## **Buses de integración de servicios para WebSphere Process Server**

Un bus de integración de servicios es un mecanismo de comunicaciones gestionado que da soporte a la integración de servicios mediante la mensajería síncrona y asíncrona. Un bus está formado por la interconexión de mensajería de interconexión que gestionan recursos de bus. Esta es una de las tecnologías de WebSphere Application Server en las que está basado WebSphere Process Server.

Algunos buses se crean automáticamente para que los utilicen el sistema, las aplicaciones de Service Component Architecture (SCA) que despliega y otros componentes. También puede crear buses para dar soporte a la lógica de integración de servicio u otras aplicaciones, por ejemplo, para dar soporte a las aplicaciones que actúan como solicitantes de servicios y proveedores dentro de WebSphere Process Server o para enlazar con WebSphere MQ.

un destino de bus es una dirección lógica a la que pueden asociarse aplicaciones como, por ejemplo, un productor, consumidor o ambos. Un destino de cola es un destino de bus que se utiliza para la mensajería de punto a punto.

Cada bus puede tener uno o varios miembros, siendo cada uno de ellos un servidor o un clúster.

La *topología de bus* es la disposición física de los servidores de aplicaciones, motores de mensajería, los gestores de colas de WebSphere MQ y el patrón de conexiones del bus y los enlaces entre los mismos que forman el Enterprise Service Bus.

Algunos buses de integración se crean automáticamente para dar soporte a WebSphere Process Server. Se pueden crear hasta cuatro buses cuando crea el entorno de despliegue o configura un servidor o clúster para dar soporte a aplicaciones SCA. Cada uno de estos buses tiene tres alias de autenticación que debe configurar.

### **Bus del sistema SCA:**

El *bus del sistema SCA* es un bus de integración de servicio que se utiliza para alojar los destinos de cola para módulos de Service Component Architecture (SCA). El tiempo de ejecución SCA, que da soporte a módulos de mediación, utiliza destinos de cola del bus del sistema como infraestructura para dar soporte a interacciones entre componentes y módulos.

El bus del sistema se crea automáticamente al crear un entorno de despliegue o al configurar un servidor o un clúster para dar soporte a aplicaciones SCA. El bus del sistema proporciona un ámbito en el que se configuran recursos, como los destinos de cola, para módulos de mediación y puntos finales de interacción. El bus permite

el direccionamiento de mensajes entre puntos finales. Puede especificar la calidad de servicio para el bus, incluidas la prioridad y la fiabilidad.

El nombre del bus es SCA.SYSTEM.busID.Bus. El alias de autenticación utilizado para proteger este bus es SCA\_Auth\_Alias.

#### **Bus de aplicaciones SCA:**

Los destinos del bus de aplicaciones soportan la comunicación entre los adaptadores de WebSphere Business Integration Adapters y otros componentes de System Component Architecture.

El bus de aplicaciones se crea automáticamente al crear un entorno de despliegue o al configurar un servidor o un clúster para dar soporte a aplicaciones SCA. El bus de aplicaciones es parecido a los buses de integración de servicios que podría crear para dar soporte a la lógica de integración de servicios u otras aplicaciones.

El nombre del bus es SCA.APPLICATION.busID.Bus. El alias de autenticación utilizado para proteger este bus es SCA\_Auth\_Alias.

#### **El bus CEI (Common Event Infrastructure):**

El bus CEI (Common Event Infrastructure) se utiliza para transmitir sucesos de base común de forma asíncrona al servidor CEI (Common Event Infrastructure) configurado.

El nombre del bus es CommonEventInfrastructure\_Bus. El alias de autenticación que se utiliza para proteger este bus es CommonEventInfrastructureJMSAuthAlias

#### **El bus de Business Process Choreographer:**

Utilice el nombre y la autenticación del bus de Business Process Choreographer para la transmisión de mensajes internos.

El bus de Business Process Choreographer se utiliza para transmitir mensajes internamente y para la API de JMS (Java Message Service) de Business Flow Manager.

El nombre del bus es BPC.cellName.Bus. El alias de autenticación es BPC\_Auth\_Alias.

## **Aplicaciones de servicio y módulos de servicio**

Un módulo de servicio es un módulo de Service Component Architecture (SCA) que proporciona servicios en tiempo de ejecución. Cuando despliega un módulo de servicio en WebSphere Process Server, crea una aplicación de servicio asociado que se empaqueta como un archivo EAR (Enterprise Archive).

Los módulos de servicio son las unidades básicas de desarrollo, y pueden contener componentes, bibliotecas y módulos intermedios que utiliza la aplicación de servicio asociada. Los módulos de servicio tienen exportaciones y, opcionalmente, importaciones, para definir las relaciones entre los módulos y los solicitantes y proveedores de servicio. WebSphere Process Server da soporte a los módulos para servicios de empresa y módulos de mediación. Tanto los módulos como los módulos de mediación son tipos de módulos SCA. Un módulo de mediación permite la comunicación entre aplicaciones transformando la invocación de servicio a un formato que sólo entiende el destino, pasando la petición al destino y

devolviendo el resultado al originador. Un módulo para un servicio de empresa implementa la lógica de un proceso empresarial. No obstante, un módulo también puede incluir la misma lógica de mediación que se puede empaquetar en un módulo de mediación.

## **Despliegue de una aplicación de servicio**

El proceso para desplegar un archivo EAR que contiene una aplicación de servicio es el mismo proceso que se utiliza para desplegar cualquier archivo EAR. Los valores de los parámetros de mediación se pueden modificar durante el despliegue. Una vez que se ha desplegado un archivo EAR que contiene un módulo SCA , pueden visualizarse detalles sobre la aplicación de servicio y su módulo asociado. Puede ver cómo un módulo de servicio se conecta a los solicitantes de servicios (a través de las exportaciones) y a los proveedores de servicios (a través de las importaciones).

## **Visualización de los detalles de un módulo SCA**

Los detalles del módulo de servicio que pueden visualizarse dependen del módulo SCA . Pueden incluir los atributos siguientes.

- Nombre del módulo SCA
- Descripción del módulo SCA
- Nombre de la aplicación asociada
- Información de la versión del módulo SCA , si el módulo tiene versión
- Importaciones de módulo SCA :
  - Las interfaces de importación son definiciones abstractas que describen cómo accede un módulo SCA a un servicio.
  - Los enlaces de importación son definiciones concretas que especifican el mecanismo físico por el cual un módulo SCA accede a un servicio. Por ejemplo, mediante SOAP/HTTP.
- Exportaciones de módulo SCA :
  - Las interfaces de exportación son definiciones abstractas que describen cómo acceden los solicitantes de servicios a un módulo SCA .
  - Los enlaces de exportación son definiciones concretas que especifican el mecanismo físico por el que un solicitante de servicio accede a un módulo SCA e, indirectamente, a un servicio.
- Propiedades de módulo SCA

## **Importaciones y enlaces de importación**

Las importaciones definen interacciones entre módulos de Service Component Architecture (SCA) y proveedores de servicios. Los módulos SCA utilizan las importaciones para permitir que los componentes accedan a servicios externos (servicios que se encuentran fuera del módulo SCA ) mediante una representación local. Los enlaces de importación definen la forma específica de acceder a un servicio externo.

Si módulos SCA no necesita acceder a servicios externos, éstos no son necesarios para tener importaciones. Los módulos de mediación suelen tener una o más importaciones que se utilizan para pasar mensajes o solicitudes a sus destinos previstos.

## Interfaces y enlaces

Una importación de módulo SCA necesita como mínimo una interfaz y una importación de módulo SCA sólo tiene un enlace.

- Las interfaces de importación son definiciones abstractas que define un conjunto de operaciones mediante WSDL (Web Services Description Language), un lenguaje XML para describir servicios Web. Un módulo SCA puede tener varias interfaces de importación.
- Los enlaces de importación son definiciones concretas que especifican el mecanismo físico que los módulos SCA utilizan para acceder a un servicio externo.

## Enlaces de importación soportados

WebSphere Process Server da soporte a los siguientes enlaces de importación:

- Los enlaces SCA conectan los módulos SCA a otros módulos SCA . Los enlaces SCA también se denominan como enlaces por omisión.
- Los enlaces de servicio Web permiten que los componentes accedan a servicios Web. Los protocolos soportados son SOAP1.1/HTTP, SOAP1.2/HTTP y SOAP1.1/JMS.

Puede utilizar un enlace SOAP1.1/HTTP o SOAP1.2/HTTP basado en la API Java para servicios web XML (JAX-WS), que permite la interacción con servicios que utilizan los enlaces de documento o literal RPC y que utiliza los manejadores JAX-WS para personalizar invocaciones. Se proporciona un enlace SOAP1.1/HTTP individual para permitir la interacción con servicios que utilizan un enlace codificado con RPC o donde hay un requisito para utilizar manejadores JAX-RPC para personalizar invocaciones.

- Los enlaces HTTP permiten acceder a aplicaciones mediante el protocolo HTTP.
- Los enlaces de importación EJB (Enterprise JavaBeans) permiten a los componentes SCA invocar servicios proporcionados por la lógica empresarial Java EE que se ejecuta en un servidor Java EE.
- Los enlaces de sistema de información empresarial (EIS) proporcionan conectividad entre componentes SCA y un EIS externo. Esta comunicación se consigue a través del uso de los adaptadores de recursos.
- Los enlaces JMS (Java Message Service) 1.1 permiten la interoperatividad con el proveedor de mensajería por omisión de WebSphere Application Server. JMS puede explotar distintos tipos de transporte, incluido TCP/IP y HTTP o HTTPS. De forma automática se da soporte a la clase Mensaje JMS y a sus cinco subtipos (Texto, Bytes, Objeto, Corriente y Correlación).
- Los enlaces JMS genéricos permiten la interoperatividad con proveedores JMS de terceros que se integran con WebSphere Application Server utilizando la JMS ASF (Application Server Facility) .
- Los enlaces JMS de WebSphere MQ permiten la interoperatividad con los proveedores JMS basados en WebSphere MQ. La clase Mensaje de JMS y sus cinco subtipos (Texto, Bytes, Objeto, Corriente y Correlación) están soportados de forma automática. Si desea utilizar WebSphere MQ como un proveedor JMS, utilice los enlaces JMS de WebSphere MQ.
- Los enlaces de WebSphere MQ permiten la interoperatividad con WebSphere MQ. Puede utilizar los enlaces WebSphere MQ sólo con los gestores de colas remotas a través de una conexión de cliente de WebSphere MQ; no puede utilizarlos con gestores de colas locales. Utilice los enlaces WebSphere MQ si desea comunicarse con las aplicaciones nativas de WebSphere MQ.

## Invocación dinámica de servicios

Los servicios pueden invocarse utilizando cualquier enlace de exportación soportado. El servicio se encuentra normalmente en un punto final especificado en la importación. Este punto final se denomina punto final estático. Puede invocar otro servicio alterando temporalmente el punto final estático. La alteración temporal dinámica de puntos finales estáticos permite invocar un servicio en otro punto final, utilizando un enlace de importación soportado cualquiera. La invocación dinámica de servicios también le permite invocar un servicio donde el enlace de importación soportado no tiene un punto final estático.

Se utiliza una importación con un enlace asociado para especificar el protocolo y su configuración para la invocación dinámica. La importación utilizada por la invocación dinámica se puede conectar al componente que realiza la llamada; o se puede seleccionar de forma dinámica en el tiempo de ejecución.

Para las invocaciones SCA y de servicio web, también es posible realizar una invocación dinámica sin una importación, con el protocolo y la configuración deducidos a partir del URL de punto final. El tipo de destino de invocación se identifica en el URL de punto final. Si se utiliza una importación, el URL debe ser compatible con el protocolo del enlace de importación.

- Un URL de SCA indica la invocación de otro módulo SCA.
- De forma predeterminada, un URL de HTTP o de JMS indica la invocación de un servicio web; para estos URL, es posible proporcionar un valor de tipo de enlace adicional que indica que el URL representa una invocación a través de un enlace HTTP o JMS.
- Para un URL HTTP de servicio web, el valor predeterminado es utilizar SOAP 1.1 y se puede especificar un valor de tipo de enlace que indica el uso de SOAP 1.2.

## Exportaciones y enlaces de exportación

Las exportaciones definen interacciones entre módulos de Service Component Architecture (SCA) y solicitantes de servicios. módulos SCA utilizan las exportaciones para ofrecer servicios a otros. Los enlaces de exportación definen la forma específica en que los solicitantes de servicios acceden a un módulo SCA .

## Interfaces y enlaces

Una exportación de módulo SCA necesita como mínimo una interfaz.

- Las interfaces de exportación son definiciones abstractas que definen un conjunto de operaciones mediante WSDL (Web Services Description Language), un lenguaje XML para describir servicios Web. Un módulo SCA puede tener varias interfaces de exportación.
- Los enlaces de exportación son definiciones concretas que especifican el mecanismo físico que los solicitantes de servicios utilizan para acceder a un servicio. Generalmente, una exportación de módulo SCA tiene un enlace especificado. Una exportación sin enlaces especificados es interpretada por el entorno de ejecución como una exportación con un enlace SCA.

## Enlaces de exportación soportados

WebSphere Process Server da soporte a los siguientes enlaces de exportación:

- Los enlaces SCA conectan los módulos SCA a otros módulos SCA . Los enlaces SCA también se denominan como enlaces por omisión.

- Los enlaces de servicio Web permiten invocar exportaciones como servicios Web. Los protocolos soportados son SOAP1.1/HTTP, SOAP1.2/HTTP y SOAP1.1/JMS. Puede utilizar un enlace SOAP1.1/HTTP o SOAP1.2/HTTP basándose en la API Java para servicios web XML (JAX-WS), que permite la interacción con servicios que utilizan enlaces de documento o de literal RPC y que utiliza los manejadores JAX-WS para personalizar invocaciones. Se proporciona un enlace SOAP1.1/HTTP individual para permitir la interacción con servicios que utilizan un enlace codificado con RPC o donde hay un requisito para utilizar manejadores JAX-RPC para personalizar invocaciones.
- Los enlaces HTTP permiten acceder a exportaciones mediante el protocolo HTTP.
- Los enlaces de exportación EJB (Enterprise JavaBeans) permiten exponer los componentes SCA como EJB, de forma que la lógica empresarial Java EE puede invocar componentes SCA que no estarían disponibles de otra forma.
- Los enlaces de sistema de información empresarial (EIS) proporcionan conectividad entre componentes SCA y un EIS externo. Esta comunicación se consigue a través del uso de los adaptadores de recursos.
- Los enlaces JMS (Java Message Service) 1.1 permiten la interoperatividad con el proveedor de mensajería por omisión de WebSphere Application Server. JMS puede explotar distintos tipos de transporte, incluido TCP/IP y HTTP o HTTPS. De forma automática se da soporte a la clase Mensaje JMS y a sus cinco subtipos (Texto, Bytes, Objeto, Corriente y Correlación).
- Los enlaces JMS genéricos permiten la interoperatividad con proveedores JMS de terceros que se integran con WebSphere Application Server utilizando la JMS ASF (Application Server Facility) .
- Los enlaces JMS de WebSphere MQ permiten la interoperatividad con los proveedores JMS basados en WebSphere MQ. La clase Mensaje de JMS y sus cinco subtipos (Texto, Bytes, Objeto, Corriente y Correlación) están soportados de forma automática. Si desea utilizar WebSphere MQ como un proveedor JMS, utilice los enlaces JMS de WebSphere MQ.
- Los enlaces de WebSphere MQ permiten la interoperatividad con WebSphere MQ. Utilice una conexión remota (o cliente) para conectarse a un gestor de colas MQ en una máquina remota. Una conexión local (o enlaces) es una conexión directa a WebSphere MQ. Sólo se puede utilizar para una conexión a un gestor de colas MQ en la misma máquina. WebSphere MQ permitirá ambos tipos de conexión, pero los enlaces MQ sólo soportan la conexión "remota" (o "cliente").

## Módulos de mediación

Los Módulos de mediación son módulos de Service Component Architecture (SCA) que pueden cambiar el formato, el contenido o el destino de las solicitudes de servicio.

Los Módulos de mediación operan en mensajes que se encuentran de camino entre los solicitantes de servicios y los proveedores de servicios. Puede direccionar los mensajes a distintos proveedores de servicio y corregir el contenido o el formato de los mensajes. Los Módulos de mediación pueden proporcionar funciones tales como la anotación cronológica de mensajes y el proceso de errores, que se adaptan a sus requisitos.

Determinados aspectos de los módulos de mediación se pueden modificar desde la consola administrativa de WebSphere Process Server, sin tener que desplegar el módulo.

## Componentes de módulos de mediación

Los módulos de mediación contienen los elementos siguientes:

- Importaciones, que definen interacciones entre módulos SCA y proveedores de servicios. Permiten que los módulos SCA llamen a servicios externos como si fueran locales. Puede ver las importaciones del módulo de mediación desde WebSphere Process Server y modificar el nuevo enlace.
- Exportaciones, que definen interacciones entre módulos SCA y solicitantes de servicios. Permiten que un módulo SCA ofrezca un servicio y defina las interfaces externas (puntos de acceso) de un módulo SCA . Puede ver las exportaciones de módulo de mediación desde WebSphere Process Server.
- Componentes de SCA, que son elementos básicos de los módulos SCA como, por ejemplo, módulos de mediación. Puede crear y personalizar los módulos SCA y los componentes gráficamente mediante WebSphere Integration Developer. Después de desplegar un módulo de mediación puede personalizar determinados aspectos del mismo desde la consola administrativa de WebSphere Process Server, sin tener que volver a desplegar el módulo.

Generalmente, los módulos de mediación contienen un tipo específico de componente SCA denominado *componente de flujo de mediación*. Los componentes de flujo de mediación definen flujos de mediación.

Un componente de flujo de mediación puede contener varios primitivos de mediación, uno o ninguno. WebSphere Process Server da soporte a un conjunto de primitivos de mediación suministrados que proporcionan funcionalidad para el direccionamiento y transformación de mensajes. Si necesita la flexibilidad adicional de las primitivas de mediación, puede utilizar la primitiva Mediación personalizada para invocar la lógica personalizada.

El objetivo de un módulo de mediación que no contiene un componente de flujo de mediación es transformar las solicitudes de servicio de un protocolo a otro. Por ejemplo, se puede realizar una solicitud de servicio mediante SOAP/JMS, pero puede que sea necesario transformar a SOAP/HTTP antes de enviarla.

**Nota:** Puede efectuar determinados cambios en los módulos de mediación desde WebSphere Process Server. No obstante, no puede ver ni cambiar los componentes SCA dentro de un módulo WebSphere Process Server. Utilice WebSphere Integration Developer para personalizar componentes SCA.

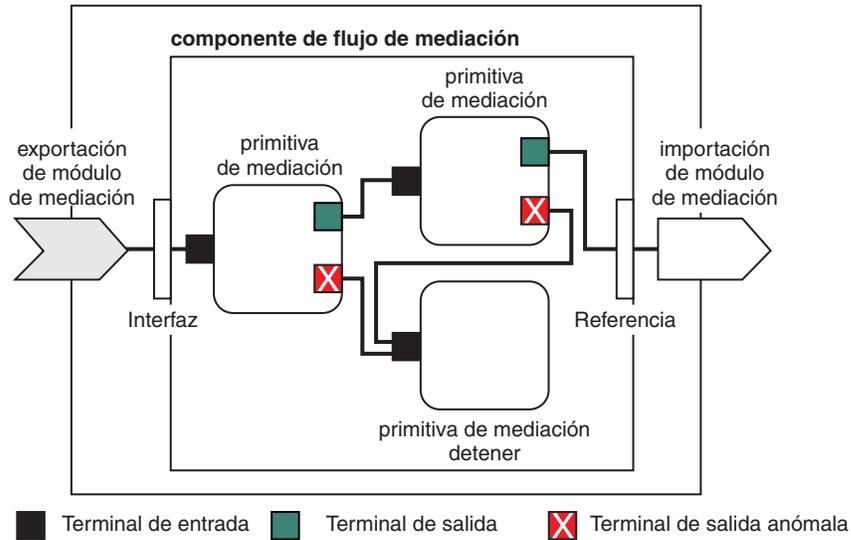


Figura 18. Ejemplo simplificado de un módulo de mediación. El módulo de mediación contiene un componente de flujo de mediación, que contiene primitivas de mediación.

- Propiedades

Los Primitivos de mediación tienen propiedades, algunas de las cuales pueden visualizarse en la consola administrativa como propiedades adicionales de un módulo SCA .

Para que se puedan ver las propiedades de un primitivo de mediación desde la consola administrativa de WebSphere Process Server, el desarrollador de la integración debe promover las propiedades. Determinadas propiedades se prestan a ser configuradas de forma administrativa y WebSphere Integration Developer las describe como propiedades promovibles, porque se pueden promover desde el ciclo de integración al ciclo administrativo. Otras propiedades no son adecuadas para la configuración administrativa, normalmente porque su modificación puede afectar al flujo de mediación de tal forma que sea necesario volver a desplegar el módulo de mediación. WebSphere Integration Developer lista las propiedades que puede elegir para su promoción bajo las propiedades promovidas de un primitivo de mediación.

Puede utilizar la consola administrativa de WebSphere Process Server para cambiar el valor de las propiedades promovidas sin tener que volver a desplegar un módulo de mediación ni reiniciar el servidor o el módulo.

En general, los flujos de mediación utilizan los cambios de propiedad inmediatamente. No obstante, si se producen cambios en una célula del gestor de despliegue, esos cambios entran en vigor en cada nodo después de que se ha sincronizado el nodo. Asimismo, los flujos de mediación en camino siguen utilizando los valores anteriores.

**Nota:** En la consola administrativa, sólo puede cambiar valores de las propiedades, no grupos, nombres ni tipos de propiedades. Si desea cambiar grupos, nombres o tipos de propiedades, debe utilizar WebSphere Integration Developer.

- Un módulo de mediación o una biblioteca dependiente también puede definir subflujos. Un subflujo encapsula un conjunto de conexiones de primitivas de mediación como una parte de lógica de integración reutilizable. Se puede añadir una primitiva a un flujo de mediación para invocar un subflujo.

## Despliegue de módulos de mediación

Los Módulos de mediación se crean mediante WebSphere Integration Developer y generalmente se despliegan a WebSphere Process Server dentro de un archivo EAR (Enterprise Archive).

Puede cambiar el valor de las propiedades promovidas en el momento del despliegue.

Puede exportar un módulo de mediación desde WebSphere Integration Developer y hacer que WebSphere Integration Developer empaquete el módulo de mediación dentro de un archivo JAR (Java ARchive) y el archivo JAR dentro de un archivo EAR. A continuación podrá desplegar el archivo EAR instalando una nueva aplicación desde la consola administrativa.

Los Módulos de mediación pueden considerarse como una entidad. Sin embargo, los módulos SCA se definen mediante varios archivos XML almacenados en un archivo JAR.

Ejemplo de archivo EAR, que contiene un módulo de mediación



Figura 19. Ejemplo simplificado de un archivo EAR que contiene un módulo de mediación. El archivo EAR contiene varios archivos JAR. El archivo JAR de programa de utilidad contiene un módulo de mediación.

## Primitivas de mediación

Los componentes de flujos de mediación operan sobre flujos de mensaje entre componentes de servicio. Las posibilidades de un componente de mediación se implementan mediante *primitivas de mediación*, que implementan tipos de implementación de servicio estándar.

Un componente de flujos de mediación tiene uno o varios flujos. Por ejemplo, uno para la petición y otro para la respuesta.

WebSphere Process Server admite un conjunto de primitivas de mediación suministrado, que implementa las posibilidades de mediación estándar para los módulos de mediación o los módulos desplegados en WebSphere Process Server. Si necesita posibilidades de mediación especiales, puede desarrollar sus propios primitivos de mediación personalizados.

Una primitiva de mediación define una operación de “entrada” que procesa o gestiona los mensajes que están representados por SMO (Objetos de mensajes de servicio). Una primitiva de mediación también puede definir operaciones de “salida” que envían mensajes a otro componente o módulo.

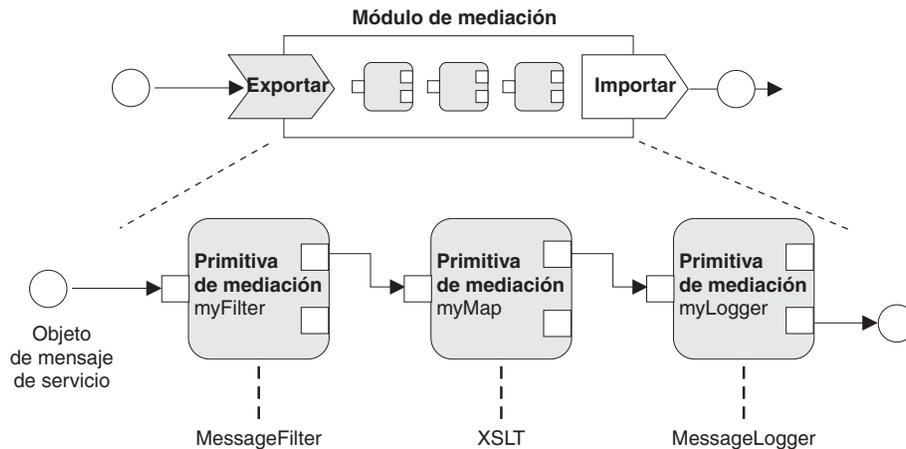


Figura 20. Módulo de mediación que contiene tres primitivos de mediación

Puede utilizar WebSphere Integration Developer para configurar la primitivas de mediación y definir sus propiedades. El administrador del entorno de ejecución puede visualizar algunas de estas propiedades las si las promueve. Toda propiedad de primitiva de mediación que se solicite puede ser también una propiedad dinámica. Una propiedad dinámica puede alterarse temporalmente, durante la ejecución, utilizando un archivo de políticas.

WebSphere Integration Developer también permite crear un modelo gráfico y ensamblar los componentes de flujo de mediación de primitivos de mediación y ensamblar los módulos de mediación o los módulos de los componentes de flujo de mediación. La consola administrativa hace referencia a los módulos de mediación o los módulos como módulos SCA.

WebSphere Integration Developer también permite definir subflujos en los módulos o sus bibliotecas dependientes. Un subflujo puede contener cualquier primitiva de mediación excepto la primitiva de mediación Resolución de la política. Un subflujo se invoca desde un flujo petición o de respuesta a través de una primitiva de mediación Subflujo. Las propiedades promovidas desde las primitivas de mediación de un subflujo se exponen como propiedades en las primitivas de mediación Subflujo. A continuación, se pueden volver a promocionar hasta que alcanzan el nivel de módulo en el que el administrador del tiempo de ejecución las puede modificar.

### Primitivas de mediación soportadas

El siguiente conjunto de primitivos de mediación está soportado por WebSphere Process Server:

#### Correlación de objetos empresariales

Transforma mensajes.

- Define las transformaciones de los mensajes utilizando una correlación de objetos empresariales, que puede reutilizarse.

- Le permite definir gráficamente transformaciones de los mensajes, utilizando el editor de correlaciones de objetos empresariales.
- Puede modificar el contenido de un mensaje.
- Puede transformar el tipo de mensaje de entrada a un tipo de mensaje de salida diferente.

### **Mediación personalizada**

Le permite implementar su propia lógica de mediación en código Java. La primitiva Mediación personalizada combina la flexibilidad de una primitiva de mediación definida por el usuario, con la simplicidad de una primitiva de mediación predefinida. Puede crear complejas transformaciones y patrones de direccionamiento mediante las acciones siguientes:

- Crear código Java.
- Crear sus propias propiedades.
- Añadir nuevos terminales.

Puede llamar a un servicio desde una primitiva Mediación personalizada, pero la primitiva de mediación Invocación de servicio se ha diseñado para llamar a los servicios, y proporciona funciones adicionales como, por ejemplo, poder llevar a cabo reintentos.

### **Manejador de datos**

Permite transformar una parte de un mensaje. Se utiliza para convertir el elemento de un mensaje de un formato físico a una estructura lógica, o una estructura lógica en un formato físico. El principal uso de la primitiva es la conversión a un formato físico, por ejemplo, una serie de texto en un objeto de mensaje de texto JMS se convierte en una estructura de objeto empresarial lógica, y viceversa. Esta mediación se utiliza normalmente para:

- Transformar una sección del mensaje de entrada de una estructura definida a otra, por ejemplo, cuando SMO incluye un valor de serie delimitado por comas y desea analizarlo en un determinado objeto empresarial.
- Alterar el tipo de mensaje, por ejemplo, cuando una exportación JMS que se ha configurado para utilizar un enlace de datos de tipo básico JMS y en el módulo de mediación el desarrollador de la integración decide que el contenido debe inflarse a una determinada estructura BO.

### **Búsqueda de base de datos**

Modifica mensajes, utilizando información de una base de datos proporcionada por el usuario.

- Debe configurar una base de datos, un origen de datos y los valores de autenticación de servidor para que el primitivo de mediación de búsqueda de base de datos los utilice. La consola administrativa le ayudará a llevar a cabo dichos pasos.
- El primitivo de mediación de Búsqueda de base de datos sólo se puede leer de una tabla.
- La columna de clave especificada debe contener un valor exclusivo.
- Los datos en las columnas de valor deben ser un tipo de esquema XML simple o un tipo de esquema XML que amplíe el tipo de esquema XML simple.

### **Búsqueda de puntos finales**

Permite el direccionamiento dinámico de peticiones, buscando puntos finales de servicio en un depósito.

- La información de punto final de servicio se recupera de WSRR (WebSphere Service Registry and Repository). El registro WSRR puede ser local o remoto.
- Puede realizar cambios en el registro desde la consola administrativa de WSRR.
- WebSphere Process Server necesita saber qué registro utilizar y, por tanto, se deben crear las definiciones de acceso de WSRR mediante la consola administrativa de WebSphere Process Server.

#### **Emisor de sucesos**

Mejora la supervisión al permitirle enviar los sucesos desde el interior de un componente de flujo de mediación.

- Puede suspender la acción mediar desmarcando el recuadro de selección.
- Puede consultar los sucesos del Emisor de sucesos utilizando el navegador de CBE (Common Base Events) en WebSphere Process Server.
- Sólo debería enviar sucesos al llegar a un punto que sea suficientemente importante en un flujo de mediación, por cuestiones de rendimiento.
- Puede definir las parte del mensaje que contiene el suceso.
- Los sucesos se envían en forma de Common Base Events y se envían a un servidor de Common Event Infrastructure.
- Para utilizar completamente la información del Emisor de sucesos, los consumidores de sucesos tienen que comprender la estructura de Common Base Events. Common Base Events tiene un esquema general pero éste no crea un modelo de los datos específicos de la aplicación, contenidos en los elementos de datos ampliados. Para crear un modelo de los elementos de datos ampliados, las herramientas de WebSphere Integration Developer generan un archivo de definición de catálogo de sucesos de Common Event Infrastructure para cada una de las primitivos de mediación del Emisor de sucesos que se haya configurado. Los archivos de definición de catálogo de sucesos son artefactos de exportación proporcionados para ayudarle; no se utilizan en WebSphere Integration Developer ni durante la ejecución de WebSphere Process Server. Debe consultar los archivos de definición de catálogo de sucesos cuando cree aplicaciones para consumir sucesos de emisor de sucesos.
- Puede especificar otra supervisión desde WebSphere Process Server. Por ejemplo, puede supervisar sucesos para que se emitan desde importaciones y exportaciones.

**Error** Detiene una ruta en particular de un flujo, y genera una excepción.

#### **Abanico de entrada**

Ayuda a agregar (combinar) mensajes.

- Sólo puede utilizarse en combinación con la primitiva de mediación Diseminación.
- Juntas, las primitivas de mediación Diseminación y Abanico de entrada permiten agregar datos a un mensaje de salida.
- La primitiva de mediación Abanico de entrada recibe mensajes hasta que se alcanza un punto de decisión; entonces se emite un mensaje.
- Para contener los datos de agregación, utilice el contexto compartido.

#### **Diseminación**

Ayuda a dividir y agregar (combinar) mensajes.

- Juntas, las primitivas de mediación Diseminación y Abanico de entrada permiten agregar datos a un mensaje de salida.

- En la modalidad de iteración, la primitiva de mediación Diseminación le permite iterar a través de un único mensaje de entrada que contenga un elemento que se repita. Para cada aparición del elemento que se repite, se envía un mensaje.
- Para contener los datos de agregación, utilice el contexto compartido.

#### **Método setter de cabecera HTTP**

Proporciona un mecanismo para gestionar cabeceras en mensajes HTTP.

- Puede crear, establecer, copiar o suprimir cabeceras de mensaje HTTP.
- Puede establecer varias acciones para cambiar varias cabeceras HTTP.

#### **Método setter de cabecera MQ**

Proporciona un mecanismo para gestionar cabeceras en mensajes MQ.

- Puede crear, establecer, copiar o suprimir cabeceras de mensaje MQ.
- Puede establecer varias acciones para cambiar varias cabeceras MQ.

#### **Método setter de cabecera SOAP**

Proporciona un mecanismo para gestionar cabeceras en mensajes SOAP.

- Puede crear, establecer, copiar o suprimir cabeceras de mensaje SOAP.
- Puede establecer varias acciones para cambiar varias cabeceras SOAP.

#### **Método setter del elemento de mensaje**

Proporciona un mecanismo sencillo para establecer el contenido de los mensajes.

- Puede cambiar, añadir o suprimir elementos de un mensaje.
- No cambia el tipo del mensaje.
- Los datos en las columnas de valor deben ser un tipo de esquema XML simple o un tipo de esquema XML que amplíe el tipo de esquema XML simple.

#### **Filtro de mensajes**

Direcciona mensajes por distintas vías de acceso, en función del contenido de los mensajes.

- Puede suspender la acción mediar desmarcando el recuadro de selección.

#### **Registrador de mensajes**

Registra mensajes en una base de datos relacional o mediante su propio anotador personalizado. Los mensajes se almacenan en formato XML, por consiguiente, las aplicaciones compatibles con XML pueden procesar posteriormente los datos.

- Puede suspender la acción mediar desmarcando el recuadro de selección.
- El esquema (estructura de las tablas) de la base de datos relacional lo define IBM.
- Por omisión, la primitiva de mediación del Anotador de mensajes utiliza la base de datos común. El tiempo de ejecución correlaciona el origen de los datos de jdbc/mediation/messageLog a la base de datos común.
- Puede establecer clases de implementación de manejador para personalizar el comportamiento del anotador personalizado. De manera opcional, puede proporcionar clases de implementación de formateador, clases de implementación de filtro o ambas para personalizar el comportamiento del anotador personalizado.

#### **Resolución de la política**

Permite la configuración dinámica de peticiones, buscando puntos finales de servicio, y los archivos de política asociados, en un depósito.

- Puede utilizar un archivo de política para alterar temporalmente de forma dinámica las propiedades promovidas de otras primitivas de mediación.
- La información de punto final de servicio y la información de políticas se recupera de WSRR (WebSphere Service Registry and Repository). El registro WSRR puede ser local o remoto.
- Puede realizar cambios en el registro desde la consola administrativa de WSRR.
- WebSphere Process Server necesita saber qué registro utilizar y, por tanto, se deben crear las definiciones de acceso de WSRR mediante la consola administrativa de WebSphere Process Server.

#### **Invocación de servicio**

Llama a un servicio desde dentro de un flujo de mediación, en lugar de esperar hasta el final del flujo de mediación y utilizar el mecanismo de devolución de llamada.

- Si el servicio devuelve un error, puede reintentar el mismo servicio o bien llamar a otro servicio.
- La primitiva de mediación Invocación de servicio es una potente primitiva de mediación que puede utilizarse por sí sola en llamadas de servicio simples, o bien puede combinarse con otras primitivas de mediación, en caso de tratarse de mediaciones complejas.

#### **Establecer tipo de mensaje**

Durante el desarrollo de la integración, le permite tratar los mensajes de tipo débil como si fueran de tipo fuerte. Un campo es de tipo débil si puede contener más de un tipo de datos. Un campo es de tipo fuerte si se conocen su tipo y su estructura interna.

- Durante la ejecución, la primitiva de mediación Establecer tipo de mensaje le permite comprobar que el contenido de un mensaje coincida con los tipos de datos esperados.

#### **Detener**

Detiene una vía de acceso concreta en el flujo, sin generar una excepción.

#### **Filtro de tipo**

Permite dirigir mensajes a otra vía de acceso de un flujo, según el tipo.

#### **Transformación XSL**

Transforma mensajes.

- Le permite llevar a cabo transformaciones XSL (Extensible Stylesheet Language).
- Los mensajes se transforman utilizando una transformación XSLT 1.0. La transformación opera en una serialización XML del mensaje.

#### **Direccionamiento dinámico**

Puede direccionar mensajes de diversas maneras utilizando puntos finales definidos durante la integración o puntos finales determinados dinámicamente durante la ejecución.

El direccionamiento dinámico abarca el direccionamiento de mensajes donde el flujo es dinámico pero todos los puntos finales posibles están predefinidos en un módulo de Service Component Architecture (SCA) y el direccionamiento de mensajes en el que el flujo es dinámico y la selección de punto final también es dinámica. En el último caso, los puntos finales de servicio se seleccionan de un origen externo durante la ejecución.

## Selección de punto final dinámico

El tiempo de ejecución tiene la capacidad de direccionar los mensajes de petición y respuesta a una dirección de punto final identificada por un elemento de cabecera de mensaje. El primitivo de mediación puede actualizar este elemento de la cabecera del mensaje en un flujo de mediación. La dirección de punto final se puede actualizar con la información de un registro, una base de datos o con información del propio mensaje. El direccionamiento de mensajes de respuesta se aplica sólo cuando la respuesta está siendo enviada por una exportación JAX-WS de servicio web.

Para que el tiempo de ejecución implemente el direccionamiento dinámico en una petición o respuesta, el módulo SCA debe tener establecida la propiedad Utilizar punto final dinámico si está establecido en la cabecera de mensaje. Los desarrolladores de integración pueden establecer la propiedad Utilizar punto final dinámico si está establecido en la cabecera del mensaje o pueden promocionarla (hacer que se pueda ver en el entorno de ejecución), para que el administrador del entorno de ejecución pueda establecerla. Puede ver las propiedades de los módulos en la ventana Propiedades de módulo. Para ver la ventana, pulse **Aplicaciones > Módulos SCA > Propiedades de módulo**. El desarrollador de integración proporciona los nombres de alias de propiedades promocionada y estos nombres son los que se muestran en la consola administrativa.

## Registro

Puede utilizar IBM WSSR (WebSphere Service Registry and Repository) para almacenar la información de los puntos finales de servicio y, a continuación, crear módulos SCA para recuperar los puntos finales del registro WSRR.

Cuando desarrolle módulos SCA, el primitivo de mediación de Búsqueda de puntos finales se utiliza para que un flujo de mediación consulte un registro WSRR para un punto final de servicio o un conjunto de puntos de finales de servicio. Si un módulo SCA recupera un conjunto de puntos finales, debe utilizar otra primitiva de mediación para seleccionar el preferido.

## Control de política de mediación de peticiones de servicio

Puede utilizar las políticas de mediación para controlar los flujos de mediación entre solicitantes de servicios y proveedores de servicios.

Puede controlar los flujos de mediación mediante las políticas de mediación almacenadas en IBM WebSphere Service Registry and Repository (WSRR). La implementación de la gestión de políticas de servicio en WSRR se basa en Web Services Policy Framework (WS-Policy).

Para controlar las peticiones de servicio mediante políticas de mediación, necesita disponer de los módulos Service Component Architecture (SCA) adecuados y los documentos de política de mediación en el registro WSRR.

## Cómo adjuntar una política de mediación a una petición de servicio

Cuando se desarrolla un módulo SCA que tiene que utilizar una política de mediación, debe incluir un primitivo de mediación Resolución de políticas en el flujo de mediación. En tiempo de ejecución, el primitivo de mediación Resolución de políticas obtiene la información de política de mediación del registro. Por lo

tanto, un módulo SCA debe contener un componente de flujo de mediación para dar soporte al control de política de mediación de las peticiones de servicio.

En el registro, puede conectar una o más políticas de mediación a un módulo SCA o a un servicio de destino utilizado por el módulo SCA. Las políticas de mediación adjuntadas se pueden utilizar (están dentro del ámbito) para todos los mensajes de servicio procesados por el módulo SCA. Las políticas de mediación pueden tener adjunciones de política que definen condiciones. Las condiciones de política de mediación permiten aplicar diferentes políticas de mediación en distintos contextos. Además, las políticas de mediación pueden tener clasificaciones, que se pueden utilizar para especificar un estado de gobierno.

## **WebSphere Service Registry and Repository**

El producto WebSphere Service Registry and Repository (WSRR) le permite almacenar, acceder y gestionar información sobre puntos finales de servicio y políticas de mediación. Puede utilizar WSRR para hacer que sus aplicaciones de servicio sean más dinámicas, y más adaptables a las condiciones empresariales, en cambio constante.

### **Introducción**

Los flujos de mediación utilizan WSRR como un mecanismo de bucle dinámico, que proporciona información sobre puntos finales de servicio o políticas de mediación.

Para configurar el uso de WSRR, y cree documentos de definición de WSRR utilizando la consola administrativa. De forma alternativa, puede utilizar los mandatos de administración de WSRR desde el cliente de scripts wsadmin. Las definiciones de WSRR y sus propiedades de conexión son los mecanismos utilizados para conectarse a una instancia de registro, y recuperar un punto final de servicio o una política de mediación.

### **Puntos finales de servicio**

Puede utilizar WSRR para almacenar información sobre servicios que ya utiliza, que tiene previsto utilizar o que desea reconocer. Estos servicios podrían estar en sus sistemas o en otros sistemas. Por ejemplo, una aplicación podría utilizar WSRR para localizar el servicio más apropiado para satisfacer sus necesidades funcionales y de rendimiento.

Cuando se desarrolla un módulo SCA que tiene que acceder a puntos finales de servicio de WSRR, debe incluir una primitiva de mediación búsqueda de puntos finales en el flujo de mediación. En tiempo de ejecución, la primitiva de mediación de búsqueda de puntos finales obtiene los puntos finales de servicio del registro.

### **Políticas de mediación**

Puede utilizar también WSRR para almacenar información de política de mediación. Las políticas de mediación le ayudan a controlar las solicitudes de servicio, alterando temporalmente las propiedades del módulo. Si WSRR contiene políticas de mediación que están conectadas a un objeto que representa el módulo SCA o el servicio de destino, las políticas de mediación podrían alterar temporalmente las propiedades del módulo. Si desea que políticas de mediación distintas se apliquen en contextos distintos, puede crear condiciones de política de mediación.

**Nota:** Las políticas de mediación se ocupan del control de flujos de mediación, y no de la seguridad.

Cuando se desarrolla un módulo SCA que tiene que utilizar una política de mediación, debe incluir una primitiva de mediación de resolución de políticas en el flujo de mediación. En tiempo de ejecución, la primitiva de mediación de resolución de políticas obtiene la información de política de mediación del registro.

## **Cientes de servicios de mensajes**

Message Service Clients está disponible para C/C++ y .NET para permitir a las aplicaciones no Java conectarse a Enterprise Service Bus.

Message Service Clients para C/C++ y .NET proporcionan una API denominada XMS que tiene el mismo conjunto de interfaces de la API de JMS (Java Message Service). Message Service Client para C/C++ contiene dos implementaciones de XMS, una para ser utilizada por aplicaciones en C y otra para ser utilizada por aplicaciones en C++. Message Service Client para .NET contiene una implementación totalmente gestionada de XMS, que puede ser utilizada por cualquier lenguaje compatible con .NET.

Puede obtener Message Service Clients para .NET desde [http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en\\_US&cs=utf-8&cc=us&lang=en](http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en_US&cs=utf-8&cc=us&lang=en)

Puede obtener Message Service Clients para C/C++ desde [http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en\\_US&cs=utf-8&cc=us&lang=en](http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en_US&cs=utf-8&cc=us&lang=en).

También puede instalar y utilizar el soporte de cliente Java EE desde WebSphere Application Server Network Deployment, incluidos el cliente de servicios web, el cliente EJB y el cliente JMS.

