

IBM Cúram Social Program Management



Cúram - Guía de compatibilidad para desarrollo

Versión 6.05

IBM Cúram Social Program Management



Cúram - Guía de compatibilidad para desarrollo

Versión 6.05

Nota

Antes de utilizar esta información y el producto al que hace referencia, lea la información que figura en el apartado "Avisos" en la página 29

Revisión: mayo de 2013

Esta edición se aplica a IBM Cúram Social Program Management v6.0 5 y a todos los releases posteriores hasta que se indique lo contrario en nuevas ediciones.

Materiales con licencia - Propiedad de IBM.

© Copyright IBM Corporation 2012, 2013.

© Cúram Software Limited. 2011. Reservados todos los derechos.

Contenido

Figuras v

Tablas vii

Capítulo 1. Introducción 1

1.1 Finalidad 1
1.2 A quién va destinado 1
1.3 Requisitos previos 1

Capítulo 2. Desarrollo en conformidad con Cúram 3

2.1 Descripción general 3
2.2 Inicio de un proyecto nuevo 3
 2.2.1 Comprender la estructura de directorios de desarrollo 3
 2.2.2 Control de código fuente 3
2.3 Modificación de artefactos de origen de servidor 4
 2.3.1 Escribir código fuente para nuevos métodos y clases 4
 2.3.2 Modificación de conjuntos de reglas de Cúram Express Rules (CER) 4
 2.3.3 Ampliación de tablas de códigos 5
2.4 Código fuente y API 5
 2.4.1 API internas 5
 2.4.2 API externas 6
 2.4.3 Mecanismos de extensión 6
 2.4.4 Resumen 8
2.5 Cómo evitar los escollos comunes de la conformidad 9
 2.5.1 Utilizar prefijos específicos de proyecto en nombres de artefacto 9
 2.5.2 Utilizar identificadores numéricos en datos de demostración, iniciales personalizadas 10
 2.5.3 No realice nunca modificaciones de sustitución en archivos de aplicación 10
 2.5.4 No cree nunca dependencias de artefactos de demostración o ejemplo 11

2.5.5 Cómo reflejar los cambios de los tipos de artefactos dinámicos de nuevo en el sistema de desarrollo 11
2.5.6 No cree dependencias nuevas en API internas 11

Apéndice A. Detalles de conformidad de componente 13

A.1 Introducción 13

Apéndice B. Mecanismos de extensión no recomendados 15

B.1 Introducción 15
B.2 Clases de extensión 15
 B.2.1 Entidad 15
 B.2.2 Estructura 16
 B.2.3 Process, Facade, WebService, WSInbound 17
B.3 Subclase con sustitución 18
 B.3.1 Entidad 18
 B.3.2 Process, Facade, WebService, WSInbound 20
B.4 Subclase sin sustitución 22
 B.4.1 Entidad 22
 B.4.2 Process, Facade, WebService, WSInbound 23
B.5 Modificación de dominio 25
 B.5.1 Definiciones de dominio 25
B.6 Relaciones 26
 B.6.1 Asignable 26
 B.6.2 Agregación 27
 B.6.3 Clave foránea 27
 B.6.4 Índice 27
 B.6.5 Índice exclusivo 27
B.7 Otros mecanismos 28
 B.7.1 Exclusiones 28

Avisos 29

Marcas registradas 31

Figuras

Tablas

1. Detalles de conformidad de componente	13	7. Subclase sin sustitución tal como se aplica a	
2. Clases de extensión según se aplican a clases		clases de entidad.	22
de entidad	15	8. Subclase sin sustitución tal como se aplica a	
3. Clases de extensión tal como se aplican a		otras clases modeladas.	23
clases de estructura	16	9. Sustitución de definiciones de dominio	25
4. Clases de extensión tal como se aplican a otras		10. Relaciones asignables	26
clases modeladas.	17	11. Agregaciones	27
5. Subclase con sustitución tal como se aplica a		12. Claves foráneas	27
clases de entidad.	18	13. Índices	27
6. Subclase con sustitución tal como se aplica a		14. Índices exclusivos	27
otras clases modeladas.	20	15. Exclusiones	28

Capítulo 1. Introducción

1.1 Finalidad

Este documento proporciona directrices sobre cómo crear funcionalidad personalizada que sea compatible.

Tenga en cuenta que a partir de la versión 6.0.3 algunas de estas directrices han cambiado. Mientras que todos los mecanismos de personalización de aplicaciones continúan estando soportados para los clientes que ya los han utilizado, ahora algunos de ellos no se recomiendan para nuevo desarrollo.

1.2 A quién va destinado

Este documento está pensado para que lo lean los diseñadores y desarrolladores de equipos de proyectos que creen aplicaciones Cúram.

1.3 Requisitos previos

Para la lectura de este documento se requieren conocimientos prácticos del entorno de desarrollo de aplicaciones. En toda esta guía se proporcionan referencias a la documentación sobre desarrollo correspondiente.

Capítulo 2. Desarrollo en conformidad con Cúram

2.1 Descripción general

Este capítulo describe consideraciones importantes que es necesario tener en cuenta para realizar desarrollo en conformidad con Cúram. Estas consideraciones son básicas para que el soporte técnico pueda ayudar a los clientes en la aplicación de sus propias personalizaciones de la aplicación. Siguiendo estas consideraciones, a los clientes también les resultará más fácil la actualización a futuras versiones.

2.2 Inicio de un proyecto nuevo

Al iniciar un proyecto nuevo, es importante comprender la estructura de directorios de desarrollo. También es importante colocarlo bajo control del código fuente.

2.2.1 Comprender la estructura de directorios de desarrollo

Se requieren conocimientos de la estructura de directorios de desarrollo para comprender dónde se encuentran los artefactos de desarrollo, cómo se organizan y dónde se almacenan los cambios en estos artefactos. Tenga en cuenta que para acceder a la estructura de directorios de desarrollo, en primer lugar debe instalar una versión de desarrollo de la aplicación.

La lista siguiente describe los directorios en los que se instalan los artefactos de desarrollo de cliente y servidor:

- Los artefactos de desarrollo de cliente se instalan en el directorio `webclient`. Para obtener información sobre cómo desarrollar aplicaciones de cliente de forma compatible, consulte el *Manual de consulta del cliente web de Cúram*.
- Los artefactos de desarrollo de servidor se instalan en el directorio `EJBServer`.

En los directorios `webclient` y `EJBServer` existe un subdirectorio `components`, que contiene un subdirectorio adicional denominado `custom`. El subdirectorio `custom` es dónde se deben colocar los artefactos de desarrollo específicos del proyecto. Los otros subdirectorios `components` contienen todos los artefactos de desarrollo de aplicaciones proporcionados con el producto.

Importante: La carpeta `custom` contiene una estructura de inicio para primer uso y se hace referencia a ella en toda la documentación del desarrollador como el área en la que se deben desarrollar todos los artefactos. Se debe tener en cuenta que esto no es una imposición y es elección del proyecto realizar el desarrollo en este componente o crear un componente con un nuevo nombre adecuado para su proyecto.

En el directorio `EJBServer\components\custom\model` hay un archivo de modelo de inicio y algunos fragmentos de modelo.

2.2.2 Control de código fuente

Para mantener un seguimiento de todos los cambios realizados en los artefactos de origen, la estructura de directorios de desarrollo se debe colocar bajo control de código fuente. Una vez bajo control de código fuente, todos los artefactos de desarrollo deben estar etiquetados. Asegúrese de que la etiqueta haga referencia a la versión de la aplicación. De esta forma, en cualquier momento será posible generar un informe, mediante la funcionalidad `diff`, de todos los archivos añadidos o modificados para implementar funcionalidad del proyecto. Este informe resulta útil cuando se adopta un nuevo release de la aplicación.

Tenga en cuenta que a partir de la versión 6.0.3 se han realizado cambios sobre cómo se proporciona el código fuente Java™. Para obtener más información consulte la siguiente sección 4.

2.3 Modificación de artefactos de origen de servidor

Existen muchos tipos de artefactos de servidor, algunos de los cuales son clases de aplicación. Algunos de estos se representan en un modelo de aplicación. Otras interfaces Java son "especializadas". Existen implementaciones de aplicación de estas dos categorías de clase, y es importante poder distinguirlas. Aunque es posible cambiar aspectos de una interfaz modelada modificando el modelo y volviendo a generar el código, no es posible cambiar una interfaz especializada.

Interfaces modeladas:

- Aparecen en el modelo UML de aplicación

Interfaces especializadas:

- No aparecen en el modelo UML de aplicación
- Aparecen en los directorios de los componentes del entorno de desarrollo
- No se pueden personalizar
- Contiene la anotación de Google Guice `@ImplementedBy` para indicar la clase de implementación de la aplicación

Algunos componentes pueden contener interfaces que no correspondan a ninguna de las dos categorías anteriores, pero siempre se describirán como documentación específica del componente. Tanto las interfaces de aplicación modeladas como especializadas pueden tener implementaciones que se puedan personalizar.

Para obtener información sobre cómo implementar artefactos de origen, consulte la *Guía para el desarrollador del servidor de Cúram* para las implementaciones de interfaces modeladas y la publicación *Persistence Cookbook* para implementaciones de interfaces especializadas; es necesario comprobar la interfaz implementada para determinar la categoría. Las recomendaciones sobre cómo modificar los artefactos de origen de servidor han cambiado con la versión 6.0.3. Tenga en cuenta que las recomendaciones contenidas en este documento (la Guía de Conformidad de desarrollo de Cúram) son definitivas.

2.3.1 Escribir código fuente para nuevos métodos y clases

Se deben escribir nuevas clases específicas de cliente, clases que envuelven clases existentes, o en un conjunto limitado de circunstancias nuevas subclases de clases existentes en nuevos archivos de origen. Todos los nuevos archivos de origen se deben colocar en el subdirectorio `source` del directorio `EJBServer\components\custom`.

Para clases modeladas, la jerarquía de clases generada dictará la estructura de paquete de los nuevos archivos de origen. Consulte la *Guía para el desarrollador del servidor de Cúram* para obtener información sobre cómo modelar nuevas clases y sustituir implementaciones existentes.

Para implementaciones especializadas, depende del usuario cómo se empaqueta la nueva clase. Consulte la publicación *Persistence Cookbook* para obtener información sobre cómo configurar nuevas subclases mediante Google Guice.

2.3.2 Modificación de conjuntos de reglas de Cúram Express Rules (CER)

CER Editor almacena sus conjuntos de reglas en la base de datos y no en el sistema de archivos. Los conjuntos de reglas proporcionados en el componente principal NO se deben personalizar. Las soluciones deben tener sus propias declaraciones de conformidad sobre sus conjuntos de reglas.

2.3.3 Ampliación de tablas de códigos

Tenga en cuenta que ahora se proporciona documentación para indicar qué tablas de códigos se pueden ampliar sin problemas, y cuáles requieren que los clientes pregunten al soporte técnico antes de personalizarlas. En la estructura de directorios de la documentación del proyecto correspondiente a cada instalación (en una carpeta denominada `RestrictedCodeTables`) se proporciona una lista de tablas de códigos que no se pueden ampliar sin ponerse en contacto con el soporte técnico. Si desea personalizar una tabla de códigos que aparece en esta lista, debe emitir un caso de soporte.

2.4 Código fuente y API

Actualmente toda la funcionalidad Java de la aplicación se distribuye como archivos jar precompilados. Así ha sido siempre para los módulos empresariales introducidos desde la versión 5.0 (para los que nunca se proporcionó código fuente) pero ahora ha pasado a ser cierto de manera universal. Actualmente la funcionalidad de la aplicación sólo se volverá a generar y compilar en una instalación de cliente si así lo requiere la utilización de los mecanismos de extensión del cliente. Esto significa que el proceso de compilación del cliente ya no necesita volver a compilar toda la base de código fuente Java; ahora sólo será necesario volver a compilar el código fuente específico del proyecto y el código fuente Java regenerado dependiente.

Para un número limitado de áreas funcionales principales de la versión 6.0.3 en adelante, el código fuente Java ya no se distribuye de ninguna forma. Se sigue proporcionando código fuente para el resto de la aplicación (como 'ejemplo'), pero sólo para fines de documentación. Este código no está directamente implicado en el proceso de compilación de la versión 6.0.3. Este código fuente de ejemplo se distribuye en archivos jar para cada componente individual de la forma siguiente: `EJBServer\components\\sample\src.zip`. Las versiones compiladas de cada componente se pueden encontrar en la siguiente ubicación: `EJBServer\components\\lib\.jar`.

Además, a partir de la versión 6.0.3, las operaciones de clase se han marcado como internas o externas mediante anotaciones.

Las operaciones externas forman la API oficial a la aplicación en adelante, la cual se anima a los clientes a utilizar e invocar desde su propio código.

Importante: De forma predeterminada, las clases sin anotaciones son internas.

2.4.1 API internas

Aunque es posible invocar y crear subclases de API internas desde el código personalizado, a partir de la versión 6.0.3 esto no se recomienda. Estas API se anotan con `@AccessLevel (INTERNAL)`.

Importante: En este contexto 'no se recomienda' significa que se sigue dando soporte a su utilización, pero que es posible que estas API se modifiquen o eliminen en releases posteriores, una vez que se haya dado a los clientes un período mínimo de 1 año de aviso previo respecto a cualquier cambio o eliminación.

Nota: Este aviso no se proporciona para las API marcadas como internas en la versión 6.0.3. (es decir, no hay planes actuales de cambiar ninguna de la API marcadas como internas en la versión 6.0.3) y por lo tanto debería haber tiempo suficiente para que los clientes planificaran dichas migraciones.

Las referencias de clientes existentes a las API marcadas como internas desde la versión 6.0.3 continuará funcionando como anteriormente, con la excepción de que se generarán avisos que desaconsejarán su uso en proyectos Eclipse que tengan estas dependencias.

Los proyectos deberían esforzarse en alejarse de estas dependencias de API internas con el tiempo, y no deberían introducir nuevas dependencias de ellas (dentro de lo razonable - en función de dónde se

encuentra un proyecto de cliente en su proceso de desarrollo/diseño, puede ser inevitable a corto plazo). La mayoría de los clientes existentes verán avisos de referencias no recomendables después de adoptar la versión 6.0.3 o versiones posteriores, y no se espera que los clientes las corrijan inmediatamente como parte de la actividad de adaptación. Como se ha mencionado anteriormente, esto no afectará a su derecho a recibir soporte técnico.

Tenga en cuenta que con las versiones anteriores de la aplicación, algunas API internas se han configurado para generar errores de 'restricción de acceso' en Eclipse si se hace referencia a ellas (estas API se anotan con `@AccessLevel (RESTRICTED)`), y tales referencias no estarán soportadas en los proyectos de cliente. Estas API siempre han sido internas y no han estado nunca soportadas para uso de cliente; será evidente cuál es cuál - las API de acceso restringido generan errores de Eclipse, las API no recomendadas generan avisos de Eclipse.

2.4.2 API externas

Los proyectos de cliente pueden hacer referencia directamente a las API externas. Estas API se anotan con `@AccessLevel (EXTERNAL)`. Se proporciona Javadoc para todas las API externas para cada componente individual, que se puede encontrar en `EJBServer\components\\doc\api.zip`. Tenga en cuenta que puede que algunos componentes no tengan ningún Javadoc ya que no tienen API externas. En el código de cliente sólo se puede hacer referencia a las clases que se documentan en el Javadoc; si se hace referencia a otras clases se generarán avisos sobre la recomendación de no hacerlo o errores de acceso restringido, no es compatible.

Tenga en cuenta también que, como con todas las API, se espera que las que están marcadas como externas evolucionarán con el tiempo (aunque continuarán siendo compatibles con las versiones anteriores). Si tiene un requisito que cree que no se puede cumplir mediante una combinación de la utilización de API externas y los mecanismos de extensión permitidos, debe plantearlo mediante el soporte técnico. Si corresponde, un nuevo enfoque basado en configuración o patrón de estrategia, enganche de personalización o API pasará a estar disponible, y estas nuevas API se pueden proporcionar en paquetes de características. Como alternativa, en determinadas circunstancias se puede designar de nuevo una API interna como externa, si corresponde.

2.4.3 Mecanismos de extensión

La eliminación de código fuente de las áreas de funcionalidad principal a las que se ha hecho referencia anteriormente ha producido un cambio en el enfoque recomendado de utilización de mecanismos de extensión en proyectos de cliente. Anteriormente, si los clientes deseaban utilizar los diversos mecanismos de extensión de aplicación (por ejemplo, clases de extensión, subclase con y sin sustitución, agregación), podían buscar en la base de códigos para ver dónde y cómo se estaban invocando las clases de destino en el código de aplicación. A continuación, podían realizar una evaluación de los efectos funcionales de la extensión que se estaba considerando.

A partir de la versión 6.0.3, los clientes ya no tendrán el código fuente de algunas áreas de funcionalidad principal y además una gran cantidad de API se han marcado como internas. En la siguiente sección se resume el cambio en los métodos de extensiones recomendados para proyectos de cliente.

Tenga en cuenta que esta sección sólo hace referencia a las restricciones de extensión de artefactos de aplicación. Se pueden continuar utilizando todos los mecanismos de extensión en clases definidas por el cliente, y por supuesto todos estos artefactos pueden ser externos por naturaleza e invocarse desde cualquier parte de una implementación de cliente.

Importante: Esta sección sólo proporciona un resumen de alto nivel. Todos los detalles sobre qué mecanismos se permiten en qué tipos de clases de la versión 6.0.3 se proporcionan en Apéndice B, "Mecanismos de extensión no recomendados", en la página 15. En los casos en los que no se recomienda la utilización de un mecanismo, en este apéndice se proporcionarán donde corresponda mecanismos alternativos que pueden emplear los clientes.

2.4.3.1 Clases de entidad

Con algunas excepciones, actualmente no se recomienda que el cliente utilice y modifique directamente las clases de entidad de aplicación. En muchos casos, las operaciones de clase de entidad de aplicación tienen equivalentes directos en el nivel de fachada que se han marcado como externos, y estos pueden ser utilizados por los clientes. Sin embargo, aún se permite la adición de operaciones estereotipadas y no estereotipadas en entidades de aplicación, así como la definición de diversas opciones de entidad.

Con anterioridad a la versión 6.0.3, se podían añadir atributos a las clases de entidad de aplicación mediante extensión. Sin embargo, habiéndose eliminado el código fuente de áreas de funcionalidad principal, los clientes ya no tienen visibilidad respecto a si los atributos añadidos mediante clases de extensión se correlacionarán con API externas. Por este motivo, actualmente no se recomienda añadir atributos a clases de entidad de aplicación.

Los clientes que deseen añadir datos a las pantallas de la aplicación deben añadir nuevas clases de entidad específicas de cliente y deben envolver las operaciones de mantenimiento de aplicación externas en sus propias clases de proceso para mantener ambas tablas atómicamente. A continuación, las pantallas de la aplicación se pueden modificar para que apunten a las nuevas clases de proceso.

Nota: Las entidades que representan tipos de pruebas son una excepción a esta regla. Los clientes pueden continuar añadiendo atributos a estas entidades de pruebas de la aplicación mediante extensión, ya que esto lo requiere el generador de pruebas.

En la versión 6.0.3, las entidades de pruebas de la aplicación se han marcado incorrectamente como internas; esto se corregirá en un release posterior. Por ahora, los clientes que utilicen extensión en entidades de pruebas para añadir atributos podrían ver avisos que les aconsejen no hacerlo en Eclipse en relación con estas clases; estos avisos específicos se pueden ignorar.

Esta nota sólo se aplica a entidades que representan tipos de pruebas y no para ninguna otra clase de entidad de aplicación.

2.4.3.2 Definiciones de dominio

En general, los clientes utilizan definiciones de dominio de aplicación y su modificación aún se permite. Sin embargo, actualmente no se recomienda cambiar el tipo fundamental de una definición de dominio, así como tampoco cambiar diversas opciones relacionadas con tablas de códigos.

2.4.3.3 Clases de estructura

Las clases de estructura de aplicaciones son todas básicamente externas por naturaleza, en el sentido de que se puede hacer referencia a ellas en funcionalidad específica de cliente.

No se recomienda que los clientes creen directamente agregaciones desde estructuras de aplicaciones a cualquier otra estructura (dado que ya no tienen completa visibilidad sobre dónde se utilizan estas estructuras de aplicaciones). Sin embargo, los clientes pueden continuar utilizando la agregación para incluir estructuras de aplicaciones en las estructuras específicas de su propio proyecto.

2.4.3.4 Otras clases modeladas

Para otras clases modeladas en la aplicación (por ejemplo, Process, Facade, WSInbound y WebService), actualmente no se recomienda la utilización de ningún mecanismo de extensión.

Con anterioridad a la versión 6.0.3, Subclase con sustitución era un mecanismo utilizado habitualmente para la adición y modificación de operaciones en las clases de aplicación Process y Facade. Sin embargo, de la misma forma que con la extensión de clases de entidad de aplicación, actualmente esto puede resultar inseguro, dado que los clientes ya no tienen necesariamente completa visibilidad respecto a dónde se utilizan estas clases.

De forma similar a las clases de entidad, en su lugar los clientes deben modelar y modificar sus propias clases Process, Facade o WSInbound, envolviendo API externas existentes o implementando nueva funcionalidad. Para operaciones de fachada, si se desea se puede volver a apuntar las páginas UIM afectadas a las nuevas operaciones de fachada.

2.4.3.5 Clases no modeladas

Algunos componentes contienen clases no modeladas. Para estas clases, el uso de cada clase o interfaz externa se describe en el Javadoc de la clase.

Algunas clases no modeladas se proporcionan con restricciones de acceso de Eclipse aplicadas para proporcionar a los clientes algunas directrices en relación a qué API pueden o no pueden invocar o personalizar. Determinados paquetes y clases están marcados como restringidos; estas clases no se deben utilizar ya que son clases internas que pueden cambiar con el tiempo. Las restricciones de acceso no se deben eliminar del archivo Eclipse.classpath, ya que esto podría hacer que se consumieran clases restringidas, lo que podría causar problemas durante las actualizaciones.

Algunos componentes no modelados contienen clases protegidas de paquete; estas clases no se deben utilizar en código personalizado. Los clientes no deben colocar código personalizado en la misma estructura del paquete para llamar o hacer referencia a clases protegidas de paquete.

Muchas API no modeladas no son directamente personalizables. Sólo las interfaces/clases etiquetadas con la anotación `@Implementable` se pueden ampliar o implementar. Estas clases tendrán un JavaDoc que detallará cómo personalizarlas o implementarlas. Las clases no modeladas que no están etiquetadas con la anotación `@Implementable` no se deben ampliar o implementar ya que se pueden añadir nuevas operaciones con el tiempo que pueden afectar a las actualizaciones.

Para las clases etiquetadas con la anotación `@Implementable`, los mecanismos de personalización típicos para estos tipos de clase son sucesos y estrategias.

Los sucesos permiten a los clientes añadir lógica personalizada en diversos puntos de la aplicación. Para obtener más información sobre cómo añadir escuchas de sucesos, consulte la publicación Persistence Cookbook. Las clases de sucesos normalmente se denominan 'xxxEvent', así que se pueden identificar fácilmente.

Los patrones de estrategia permiten a los clientes cambiar el comportamiento predeterminado de determinadas funciones de la aplicación. Se proporciona una implementación predeterminada para cada clase de estrategia; sin embargo, los clientes pueden elegir sustituir la implementación predeterminada de cualquiera de las operaciones de estrategia mediante la utilización de enlaces de Guice. Para obtener más información sobre cómo utilizar los enlaces de Guice, consulte la publicación Persistence Cookbook. Las clases de estrategia normalmente se denominan 'xxxStrategy', así que se pueden identificar fácilmente.

Nota: Para obtener más información sobre la conformidad para cada componente individual, consulte Apéndice A, "Detalles de conformidad de componente", en la página 13.

2.4.4 Resumen

En resumen:

Dónde desea hacer referencia a una clase de aplicación en el código personalizado:

- Si la clase es externa, se permite hacer referencia a ella.
- Si la clase es interna, puede hacer referencia a ella en su código existente pero no se recomienda que lo haga. No debe hacer referencia a ella en el nuevo código.
- Si la clase es de acceso restringido, no se permite hacer referencia a ella.

Dónde desea personalizar una clase de aplicación:

- Si la clase es modelada, consulte el Apéndice B para obtener información sobre las personalizaciones permitidas.
- Si la clase no es modelada, consulte su JavaDoc y/o guía de configuración/desarrollo de su componente padre para obtener información sobre los puntos de personalización.

2.5 Cómo evitar los escollos comunes de la conformidad

En esta sección se describen problemas de conformidad que podrían surgir y se proporcionan reglas generales para evitarlos. Resulta relativamente fácil seguir las reglas generales que se presentan en estas secciones desde las etapas iniciales de un proyecto. Sin embargo, si no se siguen, pueden producir posteriormente trastornos graves y corregirlo pueden resultar difícil y costoso.

2.5.1 Utilizar prefijos específicos de proyecto en nombres de artefacto

Debe añadir como prefijo a todos los nuevos nombres de artefacto de origen (clases de modelo, archivos de origen, mensajes, archivos de mensajes, etc.) una palabra abreviada o acrónimo relevante. Utilice siempre el mismo acrónimo o palabra abreviada. A medida que el proyecto avance, esto hará que las adiciones del proyecto a artefactos principales resulten más evidentes. Esta distinción pasa a ser muy útil a medida que aumenta el esfuerzo de desarrollo. Normalmente, la mayoría de proyectos se describirán mediante alguna clase de acrónimo. Este acrónimo es un buen candidato para utilizar como prefijo.

La utilización de un prefijo específico del proyecto evita que se produzcan conflictos de nombres entre artefactos nuevos que añada y artefactos nuevos que Cúram añada con el tiempo; los conflictos de nombres pueden resultar costosos y difíciles de corregir cuando se producen.

Por ejemplo, considere adoptar un Service Pack y descubrir que una de las adiciones de campos de base de datos personalizadas tiene el mismo nombre que un nuevo campo de la aplicación que no tiene el mismo tipo de datos o significado empresarial. Como alternativa, considere adoptar un Service Pack y descubrir que se ha añadido una nueva aplicación Elemento de tabla de códigos que está en conflicto con un elemento personalizado que también ha añadido con el mismo nombre, pero con distinto significado. Estos tipos de conflictos se pueden evitar si se asegura de que siempre se añade un prefijo coherente a los nuevos nombres de artefactos personalizados.

Algunos tipos de artefactos tienen más de un identificador y estos se deben tener en cuenta al darles nombre. Por ejemplo, las clases de entidad y los elementos de tabla de códigos. Un clase de entidad personalizada tiene un nombre de tabla que comparte el mismo espacio de nombres sin formato (el esquema de base de datos) como tablas de aplicaciones y debe tener un nombre de tabla exclusivo en ese espacio de nombres. También tiene un nombre de clase Java, que comparte un espacio de nombres jerárquico (estructura de paquetes) con clases Java de la aplicación. De la misma forma, un elemento de tabla de códigos personalizado tiene tanto un valor como un identificador Java, y ambos comparten un espacio de nombres sin formato con elementos de aplicación en la misma tabla de códigos.

Es importante tener en cuenta que la utilización de prefijos específicos de proyecto no es aplicable cuando está sustituyendo un artefacto de aplicación, ya que el mecanismo de sustitución normalmente se basa en nombrar el artefacto personalizado exactamente con el mismo nombre que el artefacto de aplicación que sustituye.

Consideraciones adicionales:

- Los identificadores se proporcionan de muchas formas, por ejemplo, un nombre de archivo, un ID XML, un nombre de clase Java o una combinación de estos.
- Se aconseja un prefijo corto ya que hay muchos lugares con restricciones de longitud (por ejemplo, determinados tipos de identificadores de base de datos).

Nota: Además de los artefactos de origen, también es importante considerar los valores de identificador que puedan estar en conflicto con valores utilizados por IBM.

Las API `TransactionInfo.setFacadeScopeObject` y `TransactionInfo.getFacadeScopeObject` permiten a los desarrolladores acceder a objetos asociados con la transacción actual. Cuando utilice esta API, para asegurarse de que los datos de la transacción no estén en conflicto con datos pertenecientes a IBM, debe utilizar una Serie como su identificador de objeto y añadir como prefijo a esta serie una palabra adecuada, como se ha descrito anteriormente.

2.5.2 Utilizar identificadores numéricos en datos de demostración, iniciales personalizados

Los datos de demostración e iniciales predefinidos se cargan en la base de datos de la aplicación mediante archivos DMX. Estos datos se instalan en la base de datos cuando un sistema se configura por primera vez o cuando se actualiza un sistema. En la aplicación se proporciona un conjunto de datos de demostración e iniciales. Es posible que los clientes deban añadir sus propios datos de demostración y/o iniciales.

Para evitar que el sistema se cuelgue con los datos de demostración e iniciales que se proporcionan en la aplicación y con los datos creados por el sistema de ejecución, es importante que los identificadores (por ejemplo, las claves primarias) de los datos de demostración e iniciales de los clientes se obtengan de los rangos reservados. Por lo tanto, se ha reservado un conjunto de rangos para uso de los clientes.

2.5.2.1 Rangos reservados

Los proyectos deben utilizar identificadores (claves primarias) en sus datos de demostración e iniciales personalizados que se obtengan de los siguientes rangos reservados:

- Claves primarias no legibles: de 45.000 a 49.999 (ambos inclusive)
- Claves primarias legibles: de 11.521 a 12.799 (ambos inclusive)
- Conjuntos de reglas: de 4.500 a 4.999 (ambos inclusive)

Se ayudará a los clientes que ya hayan utilizado identificadores que queden fuera de estos rangos para corregir esto antes de realizar su siguiente actualización.

2.5.2.2 Conjuntos de datos grandes

Es posible que de vez en cuando sea necesario generar conjuntos de datos muy grandes. Por ejemplo, esto puede ser necesario para las pruebas de carga. En estos casos, el número de registros necesarios sobrepasaría en mucho los rangos de claves asignados que se documentan aquí. En esta situación, se debe adoptar un enfoque distinto.

En lugar de utilizar claves de los rangos asignados, se debe utilizar el servidor de claves para generar los valores de claves necesarios. Si estos datos se importarán en una base de datos que se ha vuelto a crear, el valor final del conjunto de claves también se debe extraer y cargar en la tabla de conjuntos de claves, sustituyendo el valor de conjunto de claves inicial proporcionado en la aplicación. Si tiene preguntas en torno a este proceso, póngase en contacto con el soporte técnico para obtener más información.

2.5.2.3 Excepción para las tablas de códigos

Tenga en cuenta que la declaración anterior no es aplicable a las tablas de códigos.

2.5.3 No realice nunca modificaciones de sustitución en archivos de aplicación

Los Service Packs y parches de emergencia deben poder mover, reestructurar o sobrescribir sin problemas los archivos de aplicación. Si modifica estos archivos, los Service Packs o parches de emergencia pueden sobrescribirlos sin aviso. No hay ninguna garantía de que estos cambios sean compatibles con las modificaciones que ha realizado, por lo que podría ser que no fuera posible volver a aplicar los cambios de sustitución posteriormente.

Existe un número muy reducido de excepciones a esta regla y éstas se listan a continuación:

- EJBServer

- /project/config/datamanager_config.xml
- /project/config/deployment_packaging.xml
- /project/properties/Bootstrap.properties
- .classpath
- .project
- Webclient
 - /JavaSource/curam/omega3/ApplicationConfiguration.properties
 - /JavaSource/curam/omega3/il8n/CDEJResources.properties
 - .classpath
 - .project

2.5.4 No cree nunca dependencias de artefactos de demostración o ejemplo

Los artefactos de demostración y ejemplo no son adecuados para las dependencias personalizadas, es decir, API referencia-a, archivos UIM, tablas de códigos, archivos de mensajes, etc. en tales componentes de código personalizados. Estos artefactos pueden cambiar sin previo aviso.

Distintas áreas de producto de Cúram tienen distintos enfoques respecto a cómo marcar los artefactos como internos/de ejemplo/de demostración, por lo que esta guía no puede precisar exactamente cómo identificarlos. Sin embargo, existen algunas reglas generales fiables:

- Artefactos cuyo nombre, paquetes de código, paquete de modelo o vía de acceso de archivo contengan las palabras 'Internal', 'Sample' o 'Demo' (o derivados evidentes de estas palabras)

En caso de duda, póngase en contacto con el soporte técnico.

Importante: La carpeta CPMSample es interna; todo el código y todos los artefactos de esta carpeta pueden cambiar sin aviso previo. Si los clientes desean utilizar funcionalidad de CPMSample, deberán duplicarlo en su base de código.

2.5.5 Cómo reflejar los cambios de los tipos de artefactos dinámicos de nuevo en el sistema de desarrollo

Si modifica tipos de artefactos dinámicos en sistemas de producción o prueba, debe asegurarse siempre de que estas modificaciones se reflejen de nuevo en el sistema de desarrollo.

Existen diversos artefactos de desarrollo 'dinámicos' en la aplicación que se pueden modificar en tiempo de ejecución en un sistema de producción o prueba (por ejemplo, tablas de códigos, flujos de trabajo, etc.). Los cambios en tiempo de ejecución realizados en estos artefactos siempre se deben sincronizar de nuevo con la base de códigos de desarrollo de forma que los cambios de desarrollo simultáneos se pueden integrar con estos cambios en tiempo de ejecución antes del despliegue.

Pueden producirse cambios simultáneos realizados en estos artefactos durante el desarrollo de hitos rutinarios del proyecto, o al adoptar Service Packs o realizar actualizaciones de versión principal o secundaria. En todos los casos, debe haber un espacio central donde los cambios simultáneos se fusionen y validen y este es la base de códigos de desarrollo. El sistema de registro de estos artefactos es la base de códigos de desarrollo.

2.5.6 No cree dependencias nuevas en API internas

A partir de la versión 6.0.3, los clientes deben evitar invocar o personalizar operaciones y clases de aplicaciones marcadas como 'internas', ya que estas API pueden cambiar en versiones posteriores de la aplicación.

Apéndice A. Detalles de conformidad de componente

A.1 Introducción

Esta sección contiene información de conformidad para cada componente individual.

Importante: A menos que se indique lo contrario, para todos los componentes (independientemente de si se listan aquí o no) se supone que se aplican las siguientes declaraciones generales de conformidad:

Dónde desea hacer referencia a una clase de aplicación en el código personalizado:

- Si la clase es externa, se permite hacer referencia a ella.
- Si la clase es interna, puede hacer referencia a ella en su código existente pero no se recomienda que lo haga. No debe hacer referencia a ella en el nuevo código.
- Si la clase es de acceso restringido, no se permite hacer referencia a ella.

Dónde desea personalizar una clase de aplicación en su código personalizado:

- Si la clase es modelada, consulte el Apéndice B para ver lo que se permite hacer.
- Si la clase es no modelada, consulte su Javadoc (EJBServer\components\

Tabla 1. Detalles de conformidad de componente.

Esta tabla lista los componentes con API no modeladas.

Componente	Detalles
Cúram Client Development Environment	Para obtener más indicaciones sobre cómo personalizar/utilizar este componente, consulte el <i>Manual de consulta del cliente web de Cúram</i> . Tenga en cuenta que los archivos de la carpeta CuramCDEJ se copiarán en carpetas de compilación temporales durante el proceso de compilación de la aplicación. La presencia de estos archivos fuera de la carpeta CuramCDEJ no hace que estén disponibles para su personalización.
Cúram Server Development Environment	<p>El Javadoc de este componente detalla todos los puntos de personalización y API externas. En el código de cliente sólo se puede hacer referencia a las clases que se documentan en el Javadoc; si se hace referencia a otras clases se generarán avisos sobre la recomendación de no hacerlo o errores de acceso restringido. (La función de codificación de Cúram no es compatible con el uso del cliente que vaya más allá del uso que se documenta en las publicaciones <i>Guía para el desarrollador del servidor de Cúram</i> y <i>Cúram Security Handbook</i>.)</p> <p>El directorio bin de este componente contiene scripts de compilación Apache Ant que no se deben modificar directamente. Las actualizaciones de estos scripts se pueden realizar creando nuevos scripts ant personalizados y utilizando la funcionalidad de herencia Ant.</p> <p>La carpeta de controladores de este componente contiene controladores de base de datos utilizados para acceder a la base de datos de la aplicación. Si es necesario, estos controladores se pueden sustituir por el controlador correspondiente de la base de datos que se esté utilizando, siempre y cuando la base de datos sea una versión de base de datos soportada tal como se especifica en el documento <i>Requisitos previos admitidos de Cúram</i>.</p> <p>Nota: Si surge un problema con un controlador que no se ha proporcionado con el producto (es decir, que no se ha probado y verificado para su uso con la aplicación), se puede solicitar al cliente que sustituya el controlador por una versión que se haya probado, mientras que se notifica el problema específico al proveedor de terceros. Tenga en cuenta que los archivos de la carpeta CuramSDEJ se copiarán en carpetas de compilación temporales durante el proceso de compilación de la aplicación. La presencia de estos archivos fuera de la carpeta CuramSDEJ no hace que estén disponibles para su personalización.</p>
Cúram Administration Suite	Tenga en cuenta que a partir de la versión 6.0.3, la declaración de conformidad para las clases de Cúram Administration Suite no es distinta a las de cualquier otro componente. Las API externas de la suite de administración se pueden envolver e invocar desde código personalizado.
Infraestructura de persistencia	La infraestructura de persistencia no se puede personalizar. Los clientes no deben colocar código personalizado en los paquetes de código de la infraestructura de persistencia (curam.util.persistencia y todos sus subpaquetes). Para obtener información adicional sobre cómo utilizar estas API, consulte la publicación <i>Persistence Cookbook</i> .

Tabla 1. Detalles de conformidad de componente (continuación).

Esta tabla lista los componentes con API no modeladas.

Componente	Detalles
Infraestructura de CER	La declaración de conformidad de la infraestructura de CER se puede encontrar en el <i>Manual de consulta de Cúram Express Rules</i> . Las entidades de CER (es decir, cualquier entidad cuyo nombre tenga como prefijo la palabra Creole) se deben considerar internas y sujetas a cambios, y los clientes no deben actualizarlas o consultarlas excepto mediante archivos DMX o la API de CER.
Gestor de dependencias	El Gestor de dependencias abarca todos los artefactos de servidor del paquete de código curam.dependency y todos sus subpaquetes. Los siguientes componentes contribuyen al paquete de código del Gestor de dependencias: <ul style="list-style-type: none"> • la Infraestructura de CER; y • la aplicación principal. El Gestor de dependencias no se puede personalizar de ninguna forma. Todas las API del Gestor de dependencias son sólo para uso interno. La declaración de conformidad del Gestor de dependencias se puede encontrar en el <i>Manual de consulta de Cúram Express Rules</i> .
API del motor de titularidad y elegibilidad	Para ver las directrices sobre cómo configurar y personalizar este componente, lea el documento <i>Comprensión de la elegibilidad y la titularidad con Cúram Express Rules</i> .
Generador de pruebas	El Generador de pruebas es una infraestructura de la aplicación que se proporciona como parte de la estructura de directorios de Herramientas (EGTools). Para obtener más información sobre cómo utilizar el Generador de pruebas, consulte las <i>Especificaciones del generador de pruebas de Cúram</i> .
DocMaker, Herramientas de ayuda en línea	No se puede personalizar ninguna parte de la herramienta DocMaker o de las Herramientas de ayuda en línea excepto como se indica en la <i>Guía de desarrollo para el sistema de ayuda en línea de Cúram</i> .
Infraestructura de pods	La Infraestructura de pods se proporciona en los archivos widget-inf.jar y widget-utility.jar. La Infraestructura de pods no se puede personalizar. Los cargadores de pods no se pueden personalizar. Para obtener más información sobre cómo desarrollar pods, consulte la publicación <i>Guía del desarrollador de módulos de Cúram</i> .
Funded Program Management	Para ver las directrices sobre cómo personalizar este componente, consulte la <i>Guía del desarrollador de Funded Program Management</i> y el javadoc del componente.
Incidentes de Cúram	Para ver las directrices sobre cómo personalizar entidades de incidentes o sustituir implementaciones de incidentes, consulte la publicación <i>Persistence Cookbook</i> y el java del componente.
Cúram Citizen Context Viewer	Para ver directrices adicionales sobre cómo personalizar este componente, consulte la <i>Guía de configuración de Cúram Citizen Context Viewer</i> y el javadoc del componente.
Asesor de Cúram	Los siguientes componentes de servidor se proporcionan con Asesor de Cúram: Asesor.
Cúram Common Intake	Los siguientes componentes de servidor se proporcionan con Cúram Common Intake: Intake, PCR, CREOLEProgramRRRecommendation, ReferralsLite y CPMReferralsLite
Bandeja de entrada	Para ver las directrices sobre cómo configurar y personalizar este componente, consulte la Parte VI de la <i>Guía de referencia de flujo de trabajo de Cúram</i> .
Cúram Waitlists	Para ver las directrices sobre cómo personalizar este componente, consulte la <i>Guía de personalización de Cúram Waitlist</i> y el javadoc del componente.
IBM Cúram Business Intelligence and Analytics	Para ver las directrices sobre cómo personalizar este componente, consulte la <i>Guía del desarrollador de informes de Cúram Business Intelligence</i>
IBM Cúram Social Enterprise Collaboration	Los siguientes componentes de servidor se proporcionan con Social Enterprise Collaboration: SocialEnterpriseCollaboration, CaseParticipantIndex y ClientAccess.
IBM Cúram Universal Access	Para ver directrices adicionales sobre cómo personalizar este componente, consulte la <i>Guía del desarrollador de Cúram Universal Access</i> y el javadoc del componente.
IBM Cúram Outcome Management	Los siguientes componentes de servidor se proporcionan con Cúram Outcome Management: AssessmentPlanning, AssessmentPlanningCPM, DecisionAssistAssessments y SimpleOutcomeManagement
IBM Cúram Provider Management	Para ver las directrices sobre cómo personalizar este componente, consulte la <i>Guía del desarrollador de Cúram Provider Management</i> y el javadoc del componente.
IBM Cúram Youth Services(CYS)	Para ver las directrices sobre cómo personalizar entidades de CYS o sustituir implementaciones de CYS, consulte la publicación <i>Persistence Cookbook</i> y el javadoc del componente.
IBM Cúram Child Care (CCC)	Para ver las directrices sobre cómo personalizar entidades de CCC o sustituir implementaciones de CCC, consulte la publicación <i>Persistence Cookbook</i> y el javadoc del componente.

Apéndice B. Mecanismos de extensión no recomendados

B.1 Introducción

Como se ha descrito anteriormente, muchos de los mecanismos recomendados anteriormente en las versiones anteriores a la versión 6.0.3 para ampliar o reducir clases de aplicaciones ahora no se recomiendan. En este apéndice se detallan qué mecanismos están permitidos y cuáles no se recomiendan cuando se aplican a qué tipos de clases. También se sugiere qué hacer si se encuentra que una combinación de mecanismo/tipo de clase que desea utilizar ahora no se recomienda.

B.2 Clases de extensión

B.2.1 Entidad

Tabla 2. Clases de extensión según se aplican a clases de entidad

Acción	Opción del modelo	¿No recomendada?	Alternativa
Añadir una operación de entidad estereotipada (por ejemplo, <<ns>>, <<nsreadmulti>>)	Ninguna	No recomendada	En lugar de utilizar una clase <<extension>>, añada la operación estereotipada mediante la utilización de subclase con sustitución.
Cambiar una operación de entidad (por ejemplo, parámetros)	Ninguna	No recomendada	Crear una nueva operación estereotipada con la estructura deseada mediante subclase sin sustitución. Si cree que tiene una necesidad válida de cambiar la estructura de una operación de entidad de aplicación, emita un caso de soporte.
Cambiar una opción de operación de entidad	Campo de ID automático Clave de ID automático Sin SQL generado Bloqueo optimista Ordenar por SQL Where	No recomendada	Cree una nueva operación estereotipada mediante subclase sin sustitución. Si cree que tiene una necesidad válida de cambiar estas opciones en operaciones de entidad de aplicación, emita un caso de soporte.
	Auditoría a nivel de tabla de base de datos	No recomendada	Esta opción se puede establecer mediante las propiedades de tiempo de ejecución. Consulte la sección 12.3.6 de la Guía de modelado del servidor para obtener más información sobre cómo hacerlo.

Tabla 2. Clases de extensión según se aplican a clases de entidad (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
	Operación en caso de error Operación posterior al acceso a los datos Operación anterior al acceso a los datos Tratar Readmulti Max como informativos Excepción Registros de Readmulti Max devueltos	No recomendada	Los clientes sólo deben cambiar estas opciones en operaciones de entidad de aplicación mediante subclase con sustitución.
Cambiar una opción de clase de entidad	Habilitar validación	No recomendada	Los clientes sólo deben cambiar esta opción en operaciones de entidad de aplicación mediante subclase con sustitución.
	Abstracta Permitir bloqueo optimista Sin SQL generado Sustituir superclase	No recomendada	Si cree que tiene una necesidad válida de cambiar estas opciones en operaciones de entidad de aplicación, emita un caso de soporte.
	Auditar campos Último campo actualizado	Permitida	Actualmente sólo está soportado mediante clases de extensión y este continuará siendo el caso a partir de la versión 6.0.3
Añadir un atributo de entidad	Ninguna	No recomendada	Los clientes que deseen añadir datos a pantallas de la aplicación deben añadir nuevas clases de entidad específicas de cliente, y deben envolver las operaciones de CRUD de Cúram en sus propias clases de proceso para mantener ambas tablas atómicamente. A continuación, las pantallas de Cúram se pueden cambiar para que apunten a las nuevas clases de proceso.
Cambiar una opción de atributo de entidad	Permitir nulos	No recomendada	Si cree que tiene una necesidad válida de cambiar esta opción en atributos de entidad de aplicación, emita un caso de soporte.

B.2.2 Estructura

Tabla 3. Clases de extensión tal como se aplican a clases de estructura

Acción	Opción del modelo	¿No recomendada?	Alternativa
Añadir un atributo a una estructura	Ninguna	No recomendada	Crear una nueva estructura específica del proyecto y agregar la estructura de la aplicación desde la estructura específica del proyecto a la estructura de la aplicación (no a la inversa). Utilice la nueva estructura 'compuesta' en funcionalidad específica del cliente necesaria.
Cambiar un atributo de estructura	Ninguna	No recomendada	Crear una nueva estructura específica del proyecto y agregar la estructura de la aplicación desde la estructura específica del proyecto a la estructura de la aplicación (no a la inversa). Utilice la nueva estructura 'compuesta' en funcionalidad específica del cliente necesaria. Si cree que tiene una necesidad válida de modificar un atributo de una estructura de aplicación, emita un caso de soporte.

Tabla 3. Clases de extensión tal como se aplican a clases de estructura (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de estructura	Auditar campos	No recomendada	Si necesita propagar Auditar campos de una entidad a una pantalla, deberá crear nuevas operaciones estereotipadas que mantengan Auditar campos, crear una nueva fachada que reinicie las operaciones CRUD de entidad existentes y llame a las nuevas operaciones estereotipadas, y actualizar las páginas UIM según se requiera.

B.2.3 Process, Facade, WebService, WSInbound

Tabla 4. Clases de extensión tal como se aplican a otras clases modeladas

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de clase	Abstracta Generar FID Sustituir superclase Estilo de enlace WS WS es documento XML Tipo de documento Generar bean de fachada Nombre de proveedor Manejadores de solicitudes Manejadores de respuestas Validar solicitud Documento XML Esquema XML	No recomendada	Si cree que tiene una necesidad válida de cambiar estas opciones en las clases Process, Facade, WebService o WSInbound de la aplicación, emita un caso de soporte.
Añadir una operación	Ninguna	No recomendada	Esto no se ha recomendado nunca mediante clases de extensión, ya que habría necesitado que los clientes realizaran modificación in situ del código Java de la aplicación. Si desea añadir una operación a una clase Process, Facade, WebService o WSInbound, aísle la clase y operación en su propia operación y clase específica de proyecto.
Cambiar una operación (por ejemplo, visibilidad de la operación)	Ninguna	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación, si corresponde. Si no existe ningún punto de ampliación adecuado, pero cree que tiene una necesidad válida de cambiar el funcionamiento o la estructura de una operación de la aplicación, emita un caso de soporte.
Cambiar una opción de operación	Auditar llamadas de BI	No recomendada	Esta opción se puede establecer mediante las propiedades de tiempo de ejecución. Consulte la sección 12.3.6 de la Guía de modelado del servidor para obtener más información sobre cómo hacerlo.

Tabla 4. Clases de extensión tal como se aplican a otras clases modeladas (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
	Campo de fecha hábil Juego de caracteres de codificación de mensaje de bytes Generar seguridad Es XA Transaccional Tipo de mensaje Nombre JNDI de fábrica de conectores de cola Nombre JNDI de cola de respuestas Tiempo de espera de mensaje de respuesta Tipo de duplicación Transaccional Nombre JNDI de cola de transmisión	No recomendada	Si cree que tiene una necesidad válida de cambiar alguna de estas opciones en las operaciones Process, Facade, WSInbound o WebService de la aplicación, emita un caso de soporte.
	Campos seguros	No recomendada	Los clientes que deseen modificar qué campos de una operación de la aplicación se van a tratar como seguros deben envolver la operación en su propia clase y operación Facade y establece la opción Campos seguros en esta nueva operación en el valor que deseen. Las definiciones de pantalla de UIM afectadas se deben apuntar de nuevo en la nueva operación, si es necesario.
Cambiar una opción de parámetro de operación	Campos obligatorios	No recomendada	Los clientes que deseen modificar qué campos de una operación de la aplicación se tratarán como obligatorios deben envolver la operación en su propia clase y operación Facade y establecer la opción Campos obligatorios en esta nueva operación en el valor que deseen. Las definiciones de pantalla de UIM afectadas se deben apuntar de nuevo en la nueva operación, si es necesario.

B.3 Subclase con sustitución

B.3.1 Entidad

Tabla 5. Subclase con sustitución tal como se aplica a clases de entidad

Acción	Opción del modelo	¿No recomendada?	Alternativa
Añadir una operación de entidad estereotipada (por ejemplo, <<ns>>, <<nsreadmulti>>, etc.)	Ninguna	No recomendada	En lugar de utilizar subclase con sustitución, añada la operación estereotipada mediante la utilización de subclase sin sustitución. Esto garantizará que su subclase (y por lo tanto sus nuevas operaciones estereotipadas) se tratarán como 'externas' y que no verá avisos sobre su uso desaconsejado en Eclipse cuando haga referencia a ellas. Tenga en cuenta que continuará recibiendo avisos sobre su uso desaconsejado si hace referencia directamente a operaciones estereotipadas en la entidad base, ya que éstas son internas (debido al diseño).

Tabla 5. Subclase con sustitución tal como se aplica a clases de entidad (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
Añadir o modificar una operación de entidad no estereotipada	Ninguna	No recomendada	<p>En lugar de utilizar Subclase con sustitución, añada una operación no estereotipada mediante la utilización de Subclase sin sustitución. Esto garantizará que su subclase (y por lo tanto sus nuevas operaciones no estereotipadas) se tratarán como 'externas' y que no recibirá avisos sobre su uso desaconsejado en Eclipse cuando haga referencia a ellas.</p> <p>Tenga en cuenta que continuará recibiendo avisos sobre su uso desaconsejado si hace referencia directamente a operaciones de la entidad base, ya que éstas son internas (debido al diseño).</p> <p>No se recomienda que los clientes proporcionen nuevas implementaciones para operaciones de entidad de aplicación no estereotipadas.</p>
Cambiar la estructura de una operación de entidad	Ninguna	No recomendada	<p>Cree una nueva operación estereotipada mediante subclase sin sustitución.</p> <p>Si cree que tiene una necesidad válida de cambiar la estructura de una operación de entidad de aplicación, emita un caso de soporte.</p>
Cambiar una opción de operación de entidad	<p>Campo de ID automático</p> <p>Clave de ID automático</p> <p>Sin SQL generado</p> <p>Bloqueo optimista</p> <p>Ordenar por SQL</p> <p>Where</p>	No recomendada	<p>Cree una nueva operación estereotipada mediante subclase sin sustitución.</p> <p>Si cree que tiene una necesidad válida de cambiar estas opciones en operaciones de entidad de aplicación, emita un caso de soporte.</p>
	Auditoría a nivel de tabla de base de datos	No recomendada	<p>Esta opción se puede establecer mediante las propiedades de tiempo de ejecución. Consulte la sección 12.3.6 de la Guía de modelado del servidor para obtener más información sobre cómo hacerlo.</p>
	<p>Operación en caso de error</p> <p>Operación posterior al acceso a los datos</p> <p>Operación anterior al acceso a los datos</p>	Permitida (parcialmente)	<p>Se sigue permitiendo al cliente implementar puntos de salida de entidad de aplicación.</p> <p>Si los clientes desean realizar proceso en puntos de salida para los que existe una implementación predeterminada, se debe invocar la implementación predeterminada al inicio de la implementación del punto de salida del cliente (es decir, debe haber una llamada a 'super()' al principio).</p> <p>No se permite a los clientes desactivar implementaciones de punto de salida de la aplicación.</p>
	<p>¡Tratar Readmulti Max como informativos</p> <p>Excepción</p> <p>Registros de Readmulti Max devueltos</p>	Permitida	
Cambiar una opción de clase de entidad	Habilitar validación	Permitida (parcialmente)	<p>Se sigue permitiendo al cliente implementar puntos de salida de entidad de aplicación.</p> <p>Si los clientes desean realizar proceso en puntos de salida para los que existe una implementación predeterminada, se debe invocar la implementación predeterminada al inicio de la implementación del punto de salida del cliente (es decir, debe haber una llamada a 'super()' al principio).</p>
	<p>Abstracta</p> <p>Permitir bloqueo optimista</p> <p>Sin SQL generado</p>	No recomendada	<p>Si cree que tiene una necesidad válida de cambiar estas opciones en operaciones de entidad de aplicación, emita un caso de soporte.</p>

Tabla 5. Subclase con sustitución tal como se aplica a clases de entidad (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
	Auditar campos Último campo actualizado	No recomendada	Utilice clases de extensión para sustituir estas opciones en una clase de entidad de aplicación.
	Sustituir superclase	Permitida (parcialmente)	Se permite implícitamente dar soporte a otras acciones 'Permitidas' descritas en esta sección.

B.3.2 Process, Facade, WebService, WSInbound

Tabla 6. Subclase con sustitución tal como se aplica a otras clases modeladas

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de clase	Abstracta Generar FID Sustituir superclase Estilo de enlace WS WS es documento XML Tipo de documento Generar bean de fachada Nombre de proveedor Manejadores de solicitudes Manejadores de respuestas Validar solicitud Documento XML Esquema XML	No recomendada	<p>Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación si corresponde, y estableciendo las opciones adecuadas en la nueva clase.</p> <p>Si cree que tiene una necesidad válida de cambiar directamente estas opciones en las clases Process, Facade, WebService o WSInbound de la aplicación, emita un caso de soporte.</p>
Añadir una operación	Ninguna	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación, si corresponde.
Cambiar una operación	Ninguna	No recomendada	<p>Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación, si corresponde.</p> <p>Si cree que tiene una necesidad válida de cambiar directamente la estructura de las operaciones en las clases Process, Facade, WebService o WSInbound de la aplicación, emita un caso de soporte.</p>
Cambiar una opción de operación	Auditar llamadas de BI	No recomendada	Esta opción se puede establecer mediante las propiedades de tiempo de ejecución. Consulte la sección 12.3.6 de la Guía de modelado del servidor para obtener más información sobre cómo hacerlo.

Tabla 6. Subclase con sustitución tal como se aplica a otras clases modeladas (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
	Campo de fecha hábil Juego de caracteres de codificación de mensaje de bytes Generar seguridad Es XA Transaccional Tipo de mensaje Nombre JNDI de fábrica de conectores de cola Nombre JNDI de cola de respuestas Tiempo de espera de mensaje de respuesta Tipo de duplicación Transaccional Nombre JNDI de cola de transmisión	No recomendada	Si cree que tiene una necesidad válida de cambiar alguna de estas opciones en las operaciones Process, Facade, WSInbound o WebService de la aplicación, emita un caso de soporte.
	Campos seguros	No recomendada	Los clientes que deseen modificar qué campos de una operación de aplicación se tratarán como seguros deben envolver la operación en su propia operación y establecer la opción Campos seguros en la nueva operación en el valor que deseen. Las definiciones de pantalla de UIM afectadas se deben apuntar de nuevo en la nueva operación, si es necesario.
Cambiar una opción de parámetro de operación	Campos obligatorios	No recomendada	Los clientes que deseen modificar qué campos de una operación de la aplicación se tratarán como obligatorios deben envolver la operación en su propia operación y establecer la opción Campos obligatorios en la nueva operación en el valor que deseen. Las definiciones de pantalla de UIM afectadas se deben apuntar de nuevo en la nueva operación, si es necesario.

B.4 Subclase sin sustitución

B.4.1 Entidad

Tabla 7. Subclase sin sustitución tal como se aplica a clases de entidad

Acción	Opción del modelo	¿No recomendada?	Alternativa
Añadir una operación de entidad estereotipada (por ejemplo, <<ns>>, <<nsreadmulti>>, etc.)	Ninguna	Permitida	<p>En lugar de utilizar subclase con sustitución, añada la operación estereotipada mediante la utilización de subclase sin sustitución. Esto garantizará que su subclase (y por lo tanto sus nuevas operaciones estereotipadas) se tratarán como 'externas' y que no verá avisos sobre su uso desaconsejado en Eclipse cuando haga referencia a ellas.</p> <p>Tenga en cuenta que continuará recibiendo avisos sobre su uso desaconsejado si hace referencia directamente a operaciones de la entidad base, ya que éstas son internas (debido al diseño).</p>
Añadir una operación de entidad no estereotipada	Ninguna	Permitida	<p>En lugar de utilizar subclase con sustitución, añada la operación no estereotipada mediante la utilización de subclase sin sustitución. Esto garantizará que su subclase (y por lo tanto sus nuevas operaciones no estereotipadas) se tratarán como 'externas' y que no recibirá avisos sobre su uso desaconsejado en Eclipse cuando haga referencia a ellas.</p> <p>Tenga en cuenta que continuará recibiendo avisos sobre su uso desaconsejado si hace referencia directamente a operaciones de la entidad base, ya que éstas son internas (debido al diseño).</p>
Cambiar la estructura de una operación de entidad	Ninguna	No recomendada	Cree una nueva operación estereotipada mediante subclase sin sustitución.
Cambiar una opción de operación de entidad	Campo de ID automático Clave de ID automático Sin SQL generado Bloqueo optimista Ordenar por SQL Where	No recomendada	Cree una nueva operación estereotipada mediante subclase sin sustitución.
	Auditoría a nivel de tabla de base de datos	No recomendada	<p>Esta opción se puede establecer mediante las propiedades de tiempo de ejecución, si desea cambiar el comportamiento de las operaciones de la aplicación.</p> <p>De lo contrario, cree una nueva operación estereotipada para implementar la funcionalidad necesaria utilizando subclase sin sustitución.</p>
	Operación en caso de error Operación posterior al acceso a los datos Operación anterior al acceso a los datos Tratar Readmulti Max como Informational Excepción Registros de Readmulti Max devueltos	No recomendada	<p>Utilice subclase con sustitución para sustituir estas opciones en una clase de entidad de la aplicación.</p> <p>De lo contrario, cree una nueva operación estereotipada para implementar la funcionalidad necesaria utilizando subclase sin sustitución.</p>

Tabla 7. Subclase sin sustitución tal como se aplica a clases de entidad (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de clase de entidad	Habilitar validación	No recomendada	Utilice subclase con sustitución para sustituir esta opción en una clase de entidad de la aplicación. De lo contrario, cree una nueva operación estereotipada para implementar la funcionalidad necesaria utilizando subclase sin sustitución.
	Abstracta Permitir bloqueo optimista Sin SQL generado	No recomendada	Cree una nueva operación estereotipada mediante subclase sin sustitución.
	Auditar campos Último campo actualizado	No recomendada	Utilice clases de extensión para sustituir estas opciones en una clase de entidad de aplicación.

B.4.2 Process, Facade, WebService, WSInbound

Tabla 8. Subclase sin sustitución tal como se aplica a otras clases modeladas

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de clase	Abstracta Generar FID Sustituir superclase Estilo de enlace WS WS es documento XML Tipo de documento Generar bean de fachada Nombre de proveedor Manejadores de solicitudes Manejadores de respuestas Validar solicitud Documento XML Esquema XML	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación si corresponde, y estableciendo las opciones adecuadas en la nueva clase.
Añadir una operación	Ninguna	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación si corresponde y estableciendo las opciones adecuadas en la nueva operación de clase.
Cambiar una operación	Ninguna	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación, si corresponde.

Tabla 8. Subclase sin sustitución tal como se aplica a otras clases modeladas (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una opción de operación	Auditar llamadas de BI Campo de fecha hábil Juego de caracteres de codificación de mensaje de bytes Generar seguridad Es XA Transaccional Tipo de mensaje Nombre JNDI de fábrica de conectores de cola Nombre JNDI de cola de respuestas Tiempo de espera de mensaje de respuesta Tipo de duplicación Transaccional Nombre JNDI de cola de transmisión Campos seguros	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación si corresponde y estableciendo las opciones adecuadas en la nueva operación de clase.
Cambiar una opción de parámetro de operación	Campos obligatorios	No recomendada	Crear una nueva operación en una clase específica de proyecto, envolviendo API externas de la funcionalidad de clase de aplicación si corresponde, y estableciendo las opciones adecuadas en la nueva clase.

B.5 Modificación de dominio

B.5.1 Definiciones de dominio

Tabla 9. Sustitución de definiciones de dominio

Acción	Opción del modelo	¿No recomendada?	Alternativa
Cambiar una definición de dominio de aplicación específica		No recomendada (parcialmente)	No se permite la personalización por parte de los clientes de las siguientes definiciones de dominio de aplicación: TRUNCATED_NOTE_TEXT NOTE_TEXT INCIDENT_DESCRIPTION SERVICE_DELIVERY_NOTE_TEXT_SMALL INJURY_DESCRIPTION ACTION_TAKEN CITIZEN_ACCOUNT_RICH_TEXT CW_RICH_STRING VIEW_LIFE_EVENTS_POST_SUBMIT_XML_DATA RICH_TEXT_EDITOR_WIDGET SEC_RICH_TEXT_VIEW_WIDGET RICH_TEXT PROGRESS_RICH_TEXT_SMALL
Cambiar una opción de definición de dominio	Nombre de tabla de códigos Raíz de tabla de códigos	No recomendada	Crear una nueva definición de dominio con la raíz y nombre de tabla de códigos adecuados, y envolver en su propio proceso. No se permite a los clientes cambiar estas opciones de definiciones de dominio de aplicación.

Tabla 9. Sustitución de definiciones de dominio (continuación)

Acción	Opción del modelo	¿No recomendada?	Alternativa
	Comprimir espacios intermedios Convertir a mayúscula Nombre de función de validación personalizado Predeterminado Valor máximo Tamaño mínimo Valor mínimo Coincidencia de patrón Eliminar espacios iniciales Eliminar espacios finales Tipo de almacenamiento	Permitida	
	Tamaño máximo	Permitida (parcialmente)	Permitida sólo para aumentar el tamaño. Si desea reducir el tamaño de una definición de dominio de aplicación, emita un caso de soporte. No se debe utilizar para cambiar el tamaño máximo de la definición de dominio USERNAME.
Cambiar el tipo de una definición de dominio	Ninguna	No recomendada	Crear una nueva definición de dominio con el tipo adecuado y envolver en su propio proceso. No se permite a los clientes cambiar los tipos fundamentales de definiciones de dominio de aplicación.
Cambiar la longitud de serie de una definición de dominio	Ninguna	Permitida (parcialmente)	Permitida sólo para aumentar el tamaño. Si desea reducir el tamaño de una definición de dominio de aplicación, emita un caso de soporte. No se debe utilizar para cambiar la longitud de serie de la definición de dominio USERNAME.
Crear una nueva definición de dominio basada en una definición de dominio de aplicación	Ninguna	Permitida	

B.6 Relaciones

B.6.1 Asignable

Tabla 10. Relaciones asignables

Acción	¿No recomendada?
Hacer que una estructura proporcionada por el cliente sea asignable a una entidad o estructura de aplicación	Permitida

Tabla 10. Relaciones asignables (continuación)

Acción	¿No recomendada?
Hacer que una estructura de aplicación sea asignable a otra entidad o estructura de aplicación	No recomendada

B.6.2 Agregación

Tabla 11. Agregaciones

Acción	¿No recomendada?
Agregar una estructura de aplicación en una estructura proporcionada por el cliente (por ejemplo, crear una estructura de cliente que 'contenga' una estructura de aplicación)	Permitida
Agregar una estructura de aplicación o proporcionada por el cliente en una estructura de aplicación (por ejemplo, añadir cualquier estructura a una estructura de aplicación mediante agregación)	No recomendada

B.6.3 Clave foránea

Tabla 12. Claves foráneas

Acción	¿No recomendada?
Crear una nueva clave foránea donde una entidad proporcionada por el cliente es el hijo	Permitida
Crear una nueva clave foránea donde una entidad de aplicación es el hijo	No recomendada

B.6.4 Índice

Tabla 13. Índices

Acción	¿No recomendada?
Crear un nuevo índice (en una aplicación o entidad proporcionada por el cliente) utilizando una estructura proporcionada por el cliente	Permitida
Crear un nuevo índice (en una aplicación o entidad proporcionada por el cliente) utilizando una estructura de aplicación	No recomendada

B.6.5 Índice exclusivo

Tabla 14. Índices exclusivos

Acción	¿No recomendada?
Crear un nuevo índice exclusivo en una entidad de aplicación	No recomendada
Crear un índice exclusivo nuevo en una entidad proporcionada por el cliente utilizando una estructura de aplicación	No recomendada
Crear un índice exclusivo nuevo en una entidad proporcionada por el cliente utilizando una estructura proporcionada por el cliente	Permitida

B.7 Otros mecanismos

B.7.1 Exclusiones

Tabla 15. Exclusiones

Acción	¿No recomendada?
Utilizar Exclusiones para intentar excluir clases de una compilación de servidor	No recomendada: a partir de la versión 6.0.3 las clases de aplicaciones no se vuelven a compilar cada vez

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos. Es posible que IBM no ofrezca los productos, servicios o características que se describen en este documento en otros países. Solicite información al representante local de IBM acerca de los productos y servicios disponibles actualmente en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implica que sólo pueda utilizarse ese producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ningún derecho de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM. IBM podría tener patentes o solicitudes de patentes pendientes relacionadas con el tema principal que se describe en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

EE.UU.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe sus consultas, por escrito, a:

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokio 103-8510, Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde las disposiciones en él expuestas sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O IDONEIDAD PARA UN PROPÓSITO DETERMINADO. Algunos países no permiten la renuncia a garantías explícitas o implícitas en determinadas transacciones, por lo que puede que esta declaración no sea aplicable en su caso.

La información de este documento puede incluir imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede reservarse el derecho de realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Cualquier referencia incluida en esta información a sitios web que no sean de IBM sólo se proporciona para su comodidad y en ningún modo constituye una aprobación de dichos sitios web. El material de esos sitios web no forma parte del material de este producto de IBM y la utilización de esos sitios web se realizará bajo su total responsabilidad.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente. Los titulares de licencias de este programa que deseen obtener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y el uso mutuo de información que se haya intercambiado, deben ponerse en contacto con:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una cuota.

IBM proporciona el programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible para el mismo bajo los términos del Acuerdo de cliente de IBM, el Acuerdo internacional de licencias de programas de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos aquí se determinaron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar considerablemente. Algunas mediciones podrían haberse realizado en sistemas en desarrollo y, por lo tanto, no existe ningún tipo de garantía de que dichas mediciones sean las mismas en los sistemas con disponibilidad general. Además, es posible que algunas mediciones se hayan calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a sus entornos específicos.

La información relacionada con productos que no son de IBM se ha obtenido de los proveedores de dichos productos, de sus anuncios publicados o de otras fuentes de disponibilidad pública.

IBM no ha probado estos productos y no puede confirmar la precisión de rendimiento, compatibilidad ni otras afirmaciones relacionadas con productos que no son de IBM. Las preguntas relativas a las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Las afirmaciones relativas a las intenciones futuras de IBM están sujetas a cambio o retirada sin previo aviso, y sólo representan objetivos

Todos los precios de IBM que se muestran son precios de distribuidor recomendados por IBM, corresponden al momento actual y están sujetos a cambios sin aviso previo. Los precios de los distribuidores pueden variar.

Esta información se ofrece con fines de planificación únicamente. La información incluida en este documento puede cambiar antes de que los productos descritos estén disponibles.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos de la manera más completa posible, los ejemplos incluyen los nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con nombres y direcciones utilizados por empresas comerciales reales son mera coincidencia.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir los programas de ejemplo de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni implicar la fiabilidad, capacidad de servicio o función de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no es responsable de ningún daño resultante de la utilización de los programas de ejemplo por parte del usuario.

Todas las copias o fragmentos de las copias de estos programas de ejemplo o cualquier trabajo que de ellos se derive, deberán incluir un aviso de copyright como el que se indica a continuación:

© (el nombre de la empresa) (año). Algunas partes de este código proceden de los programas de ejemplo de IBM Corp.

© Copyright IBM Corp. _escriba el año o los años_. Reservados todos los derechos.

Si visualiza esta información en una copia software, es posible que no aparezcan las fotografías ni las ilustraciones en color.

Marcas registradas

IBM, el logotipo de IBM e [ibm.com](http://www.ibm.com) son marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Encontrará una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information" en <http://www.ibm.com/legal/us/en/copytrade.shtml>.

Otros nombres pueden ser marcas registradas de sus respectivos propietarios. Otros nombres de empresas, productos o servicios pueden ser marcas registradas o de servicio de terceros.



Impreso en España