

Versión 6 Release 2



Visión general técnica del producto

Versión 6 Release 2



Visión general técnica del producto

Nota

Antes de utilizar esta información y el producto al que hace referencia, lea la información que se incluye en “Avisos” en la página 35.

Contenido

Capítulo 1. Integración de negocio . . . 1

Integración entre unidades de negocio 2

Integración entre empresas 3

Capítulo 2. Familia de gestión de los procesos de negocio de WebSphere . . 5

WebSphere Integration Developer 7

Estándares 9

Roles de usuario en WebSphere Integration

Developer 9

Capítulo 3. Arquitectura orientada a servicios 11

Capítulo 4. Arquitectura de componentes de servicio 15

Componentes de servicio 15

Objetos de datos de servicio 17

Calificadores de servicio 18

Módulos 19

Importaciones y exportaciones 21

Tipos de enlace de importación y exportación de servicios 21

Seleccionar enlaces adecuados 23

Tipos de implementación de servicio 25

Objetos Java 25

Proceso BPEL 25

Máquinas de estado 27

Reglas de negocio 28

Selectores 28

Tarea manual 29

Correlación de interfaces 30

Flujo de mediación 31

Referencias autónomas 32

Capítulo 5. Soporte bidireccional . . . 33

Avisos 35

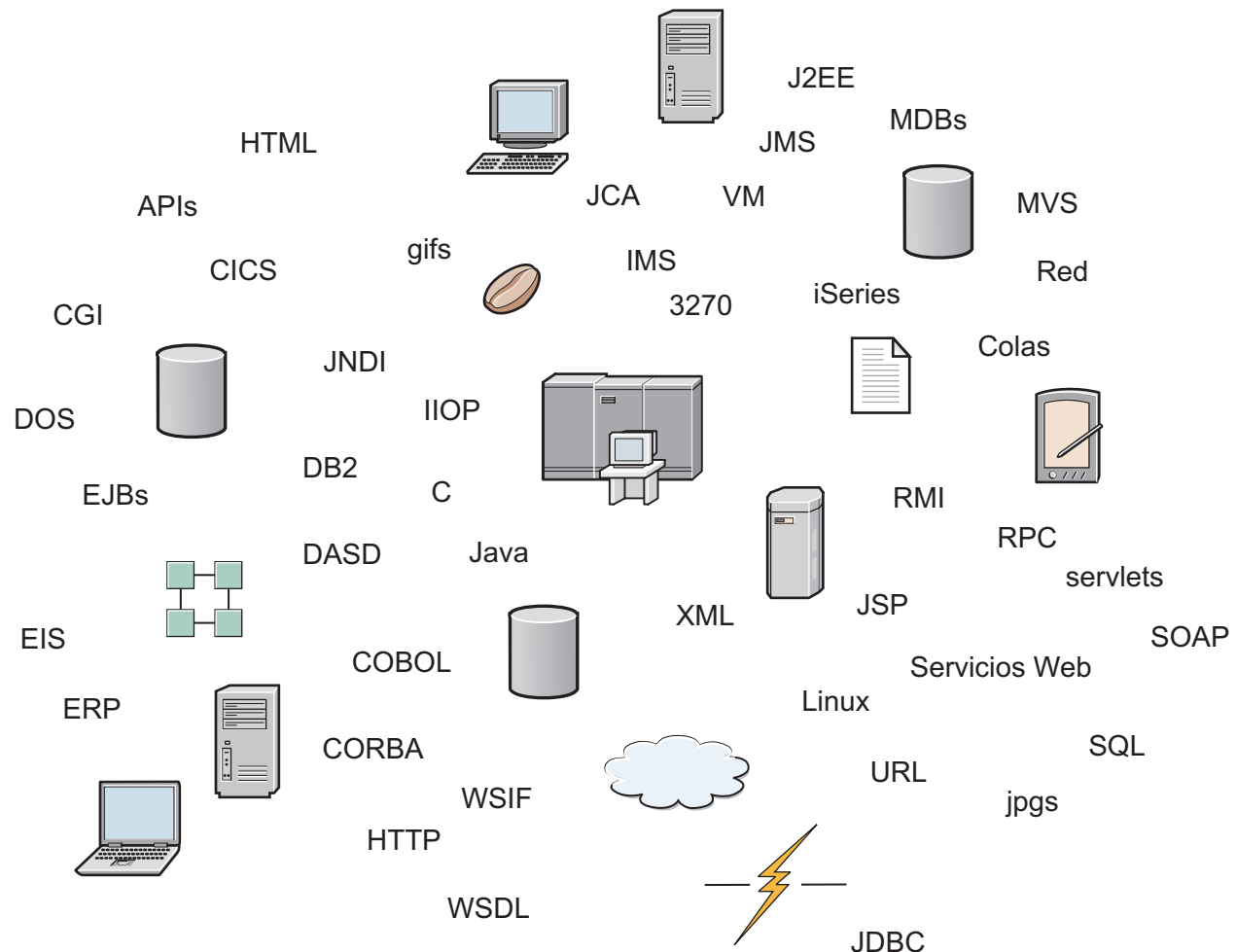
Términos de uso 39

Capítulo 1. Integración de negocio

Integración de negocio significa integración de aplicaciones, datos y procesos de una empresa o un conjunto de empresas. Se examina el desafío de esta tarea y la forma en que WebSphere Integration Developer lo lleva a cabo.

Se le ha pedido que construya un portal destinado a los clientes para sus aplicaciones de negocio. Debe proporcionar acceso a docenas de importantes aplicaciones de negocio y datos relacionados repartidos por todas las unidades de negocio. También se le ha solicitado que añada las aplicaciones de los socios de negocio al portal. La integración también significa desarrollar procesos, ya que habrá alguna lógica en la secuencia de estas aplicaciones ensambladas. Tiene 20 unidades de negocio y una docena de socios comerciales. El portal debe estar disponible en la Web las 24 horas del día. Se le han asignado seis desarrolladores, con usted incluido y tiene cuatro meses para ponerlo en marcha.

La mayor parte de las personas a las que se les planteara este reto examinarían la tecnología que han acumulado en las últimas décadas y verían un collage como el que muestra la ilustración siguiente:



Es una situación abrumadora, pero no imposible. El problema más difícil al que se enfrenta además del conjunto inconexo de hardware y software son las limitaciones de tiempo y recursos. Debe encontrar

potentes herramientas para reunir rápidamente las aplicaciones y los datos esparcidos por las empresas involucradas. La codificación manual no es una opción.

¿Es la única persona que se enfrenta a esta situación? No. Se trata de un problema general que lleva muchos años en proceso. Según la encuesta de directores generales de sistemas de información *CIO Survey* de diciembre 2001, la integración de aplicaciones es en todos los casos una de las tres prioridades máximas en materia tecnológica. Según la edición de *The Business Integrator Journal* de invierno de 2001, dos de cada tres desarrolladores de una encuesta reciente utilizan ahora software de integración para desarrollar soluciones basadas en la Web. De promedio, cada desarrollador integraba tres sistemas distintos.

Examinemos las dos fuerzas que llevan el problema al punto crítico: la integración entre unidades de negocio dentro de una empresa y la integración entre empresas. A continuación observaremos cómo WebSphere Integration Developer trata estos temas y en particular el significado del compromiso con estándares de la industria. Finalmente, trataremos sobre el especialista de integración, que es el tipo de persona que utilizará las herramientas de WebSphere Integration Developer para resolver los problemas expuestos anteriormente.

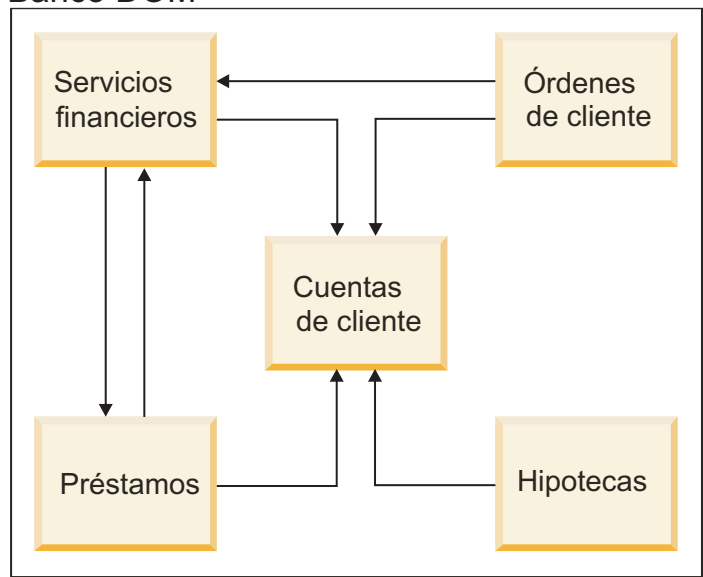
Integración entre unidades de negocio

Hoy en día las unidades de negocio colaboran, con lo que se crea la necesidad de una integración estrecha de sus aplicaciones.

Unidades de negocio que antes eran autónomas se integran ahora debido a que la tecnología les permite conectarse y a que razones de eficacia les llevan a trabajar en colaboración más estrecha para minimizar los gastos y maximizar el rendimiento. Además, un objetivo empresarial común une a las unidades de negocio. Tanto una unidad de marketing como una de investigación y desarrollo quieren producir un producto rentable. Mediante la integración del conocimiento del mercado con la información de desarrollo de productos, aumentan las probabilidades de producir ese producto exitoso. La colaboración entre unidades de negocio también permite a las empresas sacar el máximo provecho de las aplicaciones de negocio existentes al permitir su reutilización en distintos contextos de negocio.

La integración entre unidades de negocio es más sencilla que entre empresas ya que existen menos riesgos de seguridad y la gestión de las interacciones entre las unidades no puede resultar tan difícil. Es probable que las unidades de negocio utilicen los mismos protocolos, sistemas operativos y lenguajes de sistemas. Dicho de otro modo, es un entorno relativamente homogéneo. Sin embargo, la clave consiste en tener las herramientas adecuadas para integrar rápidamente las aplicaciones. En el diagrama siguiente, el banco DOM tiene varias unidades de negocio que necesitan compartir la información entre sí. Hace algunos años, el banco DOM podía llevar a cabo la gestión imprimiendo copias de información de una unidad de negocio a otra que tenía sus propios sistemas y aplicaciones. Ahora, el banco DOM debe crear aplicaciones integradas que engloben sus unidades de negocio si desea mantenerse al lado de la competencia.

Banco DOM



Integración entre empresas

Las fuerzas que controlan la integración de aplicaciones entre unidades de negocio también se aplican entre empresas, ya que las asociaciones o las adquisiciones requieren datos y procesos compartidos.

La tecnología permite a las empresas enlazarse en áreas de beneficio mutuo. Por ejemplo, un fabricante automovilístico puede configurar un proceso integrado con un proveedor de neumáticos de modo que, cuando las existencias de neumáticos sean escasas, se informe automáticamente de ello al proveedor. La integración entre las empresas viene motivada por la necesidad económica. El hecho de tener unos lazos más estrechos entre las empresas significa menos retardos y menos gastos para conseguir que las tareas se lleven a cabo. Estos procesos automatizados significan que la gente tarda menos tiempo en procesar transacciones entre empresas y que es posible reducir significativamente los costes de desplazamiento y el tiempo de reuniones personales. Los costes de administración disminuyen igualmente, así como el tiempo de respuesta entre notificaciones, entregas y facturas.

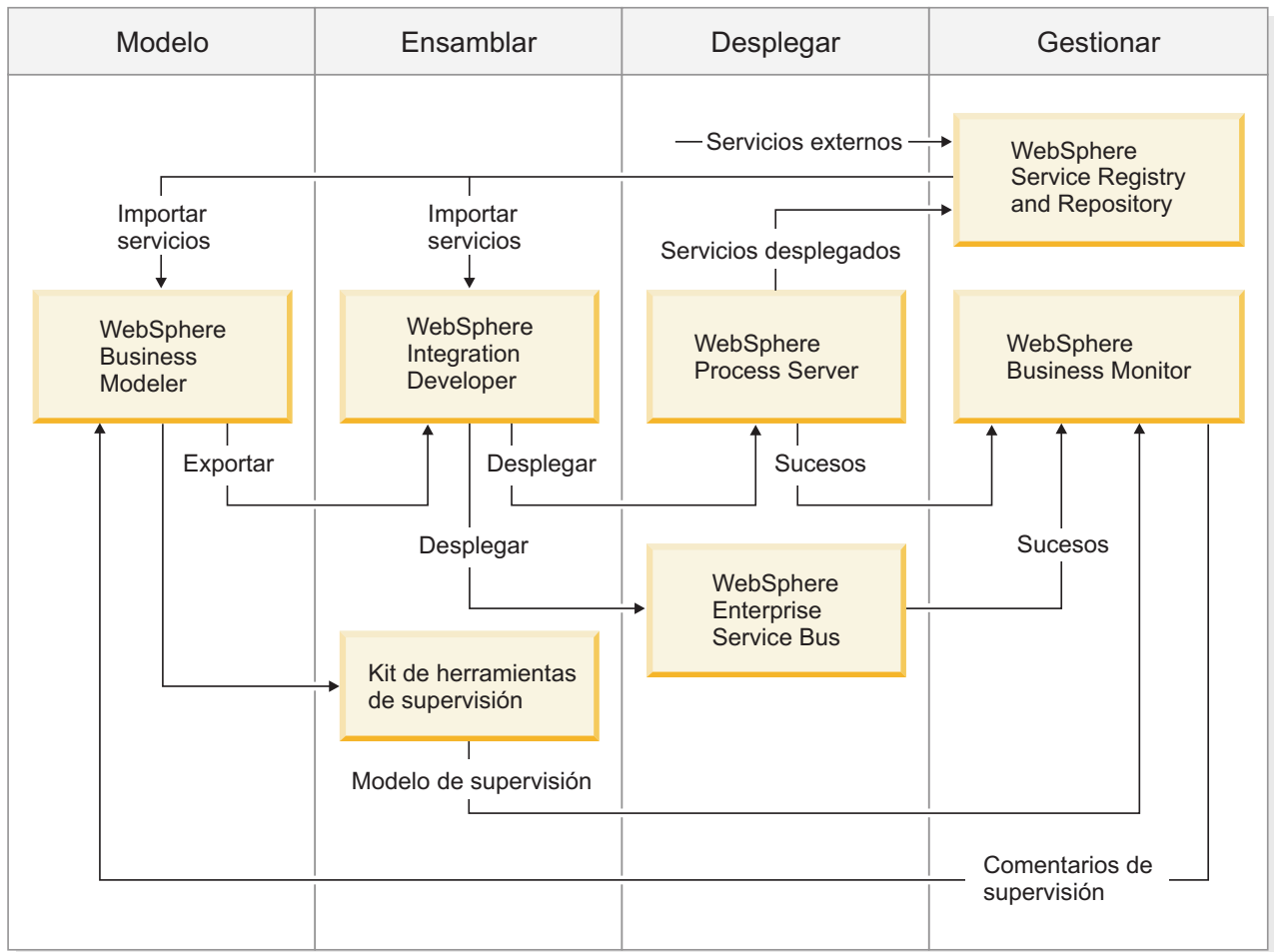
Sin embargo, las distintas empresas tienen historias también diferentes. Sus aplicaciones están codificadas en lenguajes diferentes en plataformas distintas que utilizan protocolos de comunicación diversos. Asimismo, existen mayores riesgos de seguridad al trabajar con empresas distintas. Independientemente de las ventajas e incluso la necesidad de la integración entre empresas, los costes en tiempo de desarrollo pueden ser importantes si no se dispone de las herramientas adecuadas.

Capítulo 2. Familia de gestión de los procesos de negocio de WebSphere

WebSphere Integration Developer es una familia de productos relacionados con la gestión de procesos de negocio.

Un proceso de negocio es un conjunto de actividades relacionadas que producen una salida que ninguna actividad podría producir de forma individual. El concepto de proceso de negocio ha existido durante más de cien años. Adam Smith fue uno de los primeros en definir un proceso de negocio cuando estudiaba cómo la fabricación de un producto se podría dividir entre un conjunto de trabajadores. Hoy en día, el software ha entrado en el entorno de gestión de los procesos de negocio. Con software como la familia de productos de gestión de proceso de negocio de WebSphere, se puede modelar un proceso de negocio. Así, el modelo se puede automatizar mediante la integración de aplicaciones en el proceso de negocio, en vez de realizar trabajo manual. En tiempo de ejecución, un proceso de negocio creado con software también se puede supervisar cuando esté en uso, pudiendo así mejorar la eficacia del modelo de proceso de negocio.

En el diagrama siguiente, verá varios productos de WebSphere relacionados que están agrupados por etapas que definen el enfoque de IBM a la arquitectura orientada a servicio. Veamos brevemente cada una de ellas, para ver cómo la familia de gestión de procesos de negocio de WebSphere trabaja junta para crear y gestionar de forma efectiva los procesos de negocio.



WebSphere Business Modeler, en la fase de modelado, es un punto de inicio para la gestión de procesos de negocio. Este producto puede modelar, simular y analizar un proceso de negocio. También puede especificar medidas de negocio a alto nivel para el proceso, si quiere supervisar las actividades del proceso de negocio en tiempo de ejecución. Por lo general, WebSphere Business Modeler se utilizaría para modelar su proceso de negocio actual, y también procesos de negocio futuros, adaptándose a sus necesidades. Los analistas de negocio suelen ser principalmente los usuarios de WebSphere Business Modeler. WebSphere Service Registry and Repository, en la fase de gestión, es un registro que almacena y gestiona servicios que WebSphere Business Modeler puede importar y utilizar cuando desarrolla sus procesos de negocio.

WebSphere Integration Developer, en la fase de ensamblaje, toma el modelo de proceso de negocio exportado de WebSphere Business Modeler y añade implementación y automatización al mismo. Este producto también genera su propio modelo de proceso de negocio.

Los desarrolladores de integración utilizan WebSphere Integration Developer para ensamblar las soluciones de negocio utilizando adaptadores, procesos de negocio, componentes de código y flujos de mediación. Los adaptadores proporcionan una forma de conexión a otros sistemas, como bases de datos, sistemas de planificación de recursos empresariales como PeopleSoft y registros. Los flujos de mediación transforman, direccionan y mejoran los datos. WebSphere Service Registry and Repository puede proporcionar servicios a WebSphere Integration Developer que son necesarios para crear una aplicación de integración. Las herramientas visuales en WebSphere Integration Developer implican que los desarrolladores precisan de conocimientos y habilidades mínimas de las implementaciones subyacentes. En el producto también se incluyen herramientas de prueba, depuración y despliegue.

Con el kit de herramientas de supervisión, puede añadir la implementación de medidas de negocio según se especifiquen inicialmente en WebSphere Business Modeler que, posteriormente, se pueden enviar a WebSphere Business Monitor para supervisar los procesos de negocio en tiempo de ejecución. WebSphere Integration Developer es el entorno de desarrollo principal de IBM para la creación de aplicaciones de integración. Estas aplicaciones están basadas en la arquitectura de componente de servicios, que se tratará más adelante en esta visión general del producto.

Las aplicaciones creadas en WebSphere Integration Developer se pueden desplegar en dos servidores en la fase de despliegue. WebSphere Process Server puede ejecutar cualquier aplicación creada en WebSphere Integration Developer, incluyendo aplicaciones que contienen procesos de negocio y utilizan adaptadores. Puede pensar en este servidor como en el servidor primario para aplicaciones desarrolladas en WebSphere Integration Developer. Los componentes del flujo de mediación no se despliegan en este servidor.

WebSphere Enterprise Service Bus es el servidor para los componentes de flujo de mediación. También tiene soporte para adaptadores. Puede pensar en este servidor como que funciona a nivel de infraestructura. Utiliza flujos de mediación para transformar, direccionar y mejorar los datos entre y componentes de arquitectura de componente de servicio independiente.

Ambos servidores tienen soporte para supervisión de aplicaciones en tiempo real utilizando WebSphere Business Monitor.

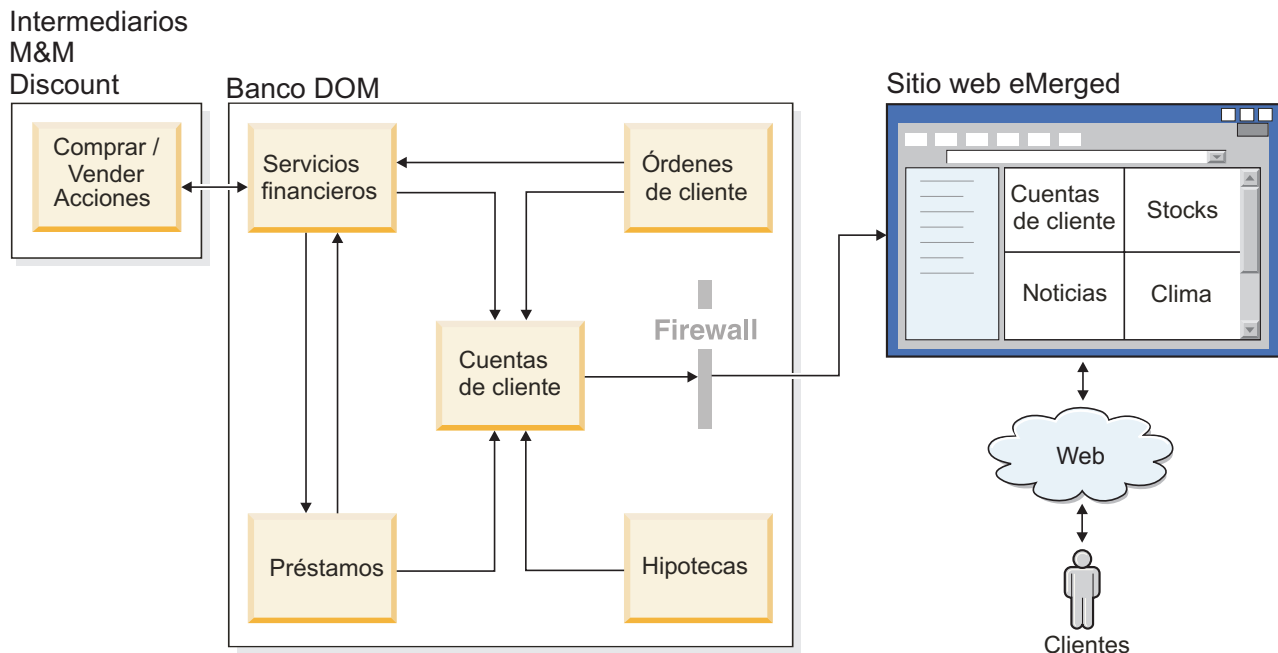
En la fase de gestión, además de WebSphere Service Registry and Repository mencionado anteriormente, WebSphere Business Monitor presenta una visualización en tiempo real de los procesos de negocio definidos anteriormente. Una vista de panel de instrumentos con indicadores que genera avisos y notificaciones que le permiten saber en todo momento el rendimiento del proceso de negocio. Los cuellos de botella, procesos ineficaces y errores en la asignación de recursos en un proceso de negocio en ejecución se pueden detectar (y corregir) rápidamente. Un panel de instrumentos de flujo de trabajo proporciona detalles sobre los procesos de negocio indicando el rendimiento de las unidades de negocio y si se cumplirán los plazos. Un panel de instrumentos de negocio proporciona informe de estadísticas, comparando el rendimiento real de la empresa con los objetivos de negocio marcados.

Los datos recopilados en tiempo real por el supervisor se pueden utilizar para revisar el proceso de negocio según está definido inicialmente en WebSphere Business Modeler. De esta forma, el ciclo completo desde el diseño de procesos de negocio hasta la información de estado sobre los mismos en base a los eventos en tiempo de ejecución, permite una continua optimización del tiempo de ejecución.

WebSphere Integration Developer

WebSphere Integration Developer es la respuesta a los desafíos de integración que una organización afronta diariamente. Se ha diseñado como un entorno de desarrollo de integración completo para aquellos que construyen aplicaciones integradas. Para simplificar y acelerar el desarrollo de aplicaciones integradas, este entorno proporciona una capa de abstracción que separa los componentes presentados visualmente con los que trabaja desde la implementación subyacente.

Las aplicaciones integradas no son simples. Pueden llamar a aplicaciones en sistemas de información de empresa (EIS), implicar procesos de negocio entre departamentos o empresas e invocar aplicaciones escritas local o remotamente en distintos lenguajes y que se ejecutan en distintos sistemas operativos. Por ejemplo, eMerged Corporation se creó de la fusión entre el banco DOM y los intermediarios M&M Discount. La fusión implicó todo lo anterior: las aplicaciones en sistemas EIS, los procesos de negocio y las aplicaciones dentro de cada empresa antigua debían compartirse entre corporaciones y presentarse sin costuras al nuevo conjunto de clientes. En cualquier caso, eMerged cumplió la tarea y, tal como se muestra en el diagrama siguiente, los clientes de los dos negocios pueden acceder a toda su información financiera en línea.



eMerged utilizó las herramientas de WebSphere Integration Developer para construir las aplicaciones integradas para sí y para sus clientes. Estas herramientas presentan aplicaciones, incluyendo aplicaciones que existen remotamente en sistemas EIS y procesos de negocio como componentes. Los componentes se crean y se ensamblan en otras aplicaciones integradas (es decir, aplicaciones creadas a partir de un conjunto de componentes) mediante de editores visuales. Los editores visuales presentan una capa de abstracción entre los componentes y sus implementaciones. Un desarrollador que utilice las herramientas puede crear una aplicación integrada sin el conocimiento detallado de la implementación subyacente de cada componente.

Para construir una aplicación integrada, las herramientas permiten seguir un método de diseño descendente, según el cuál no existe la implementación para uno o varios componentes y se añade posteriormente o ascendente, según el cuál los componentes ya están implementados y el desarrollador los ensambla arrastrándolos y soltándolos en un editor visual y crea después un flujo lógico entre ellos uniéndolos con líneas. Un entorno de depuración y prueba significa realizar una prueba completa antes de desplegar las aplicaciones en un servidor de producción. Establecer los puntos de supervisión permite ver cómo se utiliza una aplicación en tiempo real para ajustarla para un rendimiento óptimo.

Las herramientas de WebSphere Integration Developer están basadas en una arquitectura orientada a servicios. Los componentes son servicios y una aplicación integrada que incluye muchos componentes es un servicio. Los servicios creados se ajustan a los estándares principales de la industria. Los procesos de negocio, que también se convierten en componentes se crean de forma parecida con herramientas visuales de fácil utilización que se ajustan al lenguaje estándar de la industria BPEL (Business Process Execution Language.) WebSphere Integration Developer está disponible en plataformas Windows y Linux.

A continuación se proporcionan algunas ventajas de las herramientas de WebSphere Integration Developer:

- Son de fácil aprendizaje
- Se pueden aplicar a situaciones de integración compleja
- Puede generar aplicaciones que se ajusten rápidamente a estándares de la industria

Estándares

Las aplicaciones creadas por WebSphere Integration Developer se ajustan a los estándares de la industria asociados a la arquitectura orientada a servicios.

Nadie desea crear aplicaciones ligadas a código propietario que pueda no tener soporte dentro de pocos años o que implique costosos pagos por licencia. La integración basada en estándares es por tanto un aspecto fundamental de WebSphere Integration Developer. Para la conectividad, se utilizan estándares de J2EE Connector Architecture. Para la mensajería asíncrona, utilizada a menudo en grandes aplicaciones que requieren la entrega de datos garantizada, se utiliza el estándar JMS (Java Message Service). WebSphere Integration Developer puede integrar fácilmente servicios Web basados en el protocolo SOAP (Simple Object Access Protocol). Para describir un servicio, se utiliza el estándar bien establecido WSDL (Web Services Description Language). Para definir un proceso de negocio, se emplea el estándar BPEL (Business Process Execution Language).

Estas interfaces y componentes basados en estándares conforman una arquitectura abierta y conectable. Sin embargo, no se excluyen los elementos propietarios; a ellos se accede a través de la utilización de interfaces estandarizadas. Esto significa que las aplicaciones creadas en WebSphere Integration Developer pueden interactuar con aplicaciones .NET, por ejemplo. En la sección de arquitectura, se proporciona un enlace a la Arquitectura de componentes de servicio que proporciona una lista exhaustiva de los distintos estándares soportados.

Roles de usuario en WebSphere Integration Developer

El arquitecto de integración, el desarrollador de integración y el integrador son los tipos principales de usuarios de WebSphere Integration Developer.

El arquitecto de integración trabaja a nivel de arquitectura para desarrollar aplicaciones de arquitectura orientada a servicio grandes y complejas. Estas aplicaciones constan de servicios poco cohesionados que están basados en estándares del Lenguaje de descripción de servicios web (WSDL) y del Lenguaje de ejecución de procesos de negocio (BPEL). Estos servicios basados en estándares interoperan con servicios de otros proveedores, de forma que se pueden importar desde otro proveedor o desde WebSphere Service Registry and Repository (Repositorio y registro de servicios), repositorio en el que los servicios se pueden compartir en toda la empresa. Los servicios también separan la implementación de la representación ya que los servicios son independientes de la ubicación y son a menudo componentes reutilizables de otras aplicaciones orientadas a servicio, lo que significa que el arquitecto de integración puede crear rápidamente su aplicación utilizando servicios existentes o incluso servicios que aún no se han implementado.

El arquitecto de integración tiene un amplio conocimiento de todos los aspectos de la aplicación que está construyendo. Está muy familiarizado con la gestión del proceso de negocio y la arquitectura orientada a servicio, además de su experiencia en su empresa y en la industria. Trabaja a nivel de ensamblaje, construyendo el flujo global del proceso de negocio. Además del orden de los servicios, toma decisiones clave sobre los tipos de servicio a reutilizar desde cualquier lugar y los servicios que podrían implicar la invocación de aplicaciones largas y complejas en los sistemas de información de empresa (Enterprise information systems - EIS). Podría conocer cada servicio en detalle y crear estos servicios él mismo, o hacer que otros creen los servicios, como el desarrollador de integración.

El desarrollador de integración trabaja dentro de la infraestructura creada por el arquitecto de integración. Por lo general, el arquitecto de integración sería propietario de uno o más módulos utilizados en la arquitectura global de la aplicación de arquitectura orientada a servicio. El desarrollador de integración tiene un amplio conocimiento de sus módulos, incluyendo su implementación en un lenguaje de programación específico. El desarrollador de integración puede refinar el diseño creado por el arquitecto de integración, buscar servicios que puede reutilizar para rellenar el diseño del arquitecto de integración o para construir sus módulos desde cero. Una aplicación grande creada por el arquitecto de integración podría tener a varios desarrolladores de integración trabajando en ella. El desarrollador de integración está familiarizado con todos los editores y asistentes de WebSphere Integration Developer.

También está familiarizado con las interfaces de servicio en WSDL y los objetos de negocio que se utilizan para pasar datos entre sus componentes y los componentes de otros desarrolladores de integración.

El integrador trabaja principalmente en el área de la configuración de enlaces, concretamente los enlaces de importación y exportación que implican adaptadores. Se encarga de los detalles de bajo nivel de la interfaces de correlación y las definiciones de datos, así como de los aspectos de seguridad en el acceso a los sistemas EIS de fondo. Tiene conocimientos considerables sobre la integración de sistemas EIS externos con aplicaciones orientadas a servicios. Además, el integrador sabe los detalles técnicos de los propios sistemas EIS externos, como CICS, IMS, PeopleSoft y bases de datos relacional. Genera las correlaciones entre los formatos de los sistemas de fondo, como estructuras de datos de COBOL y la representación de la arquitectura orientada a servicio en XML. El integrador trabaja principalmente con el desarrollador de integración, que añade servicios que acceden a los sistemas de fondo.

Capítulo 3. Arquitectura orientada a servicios

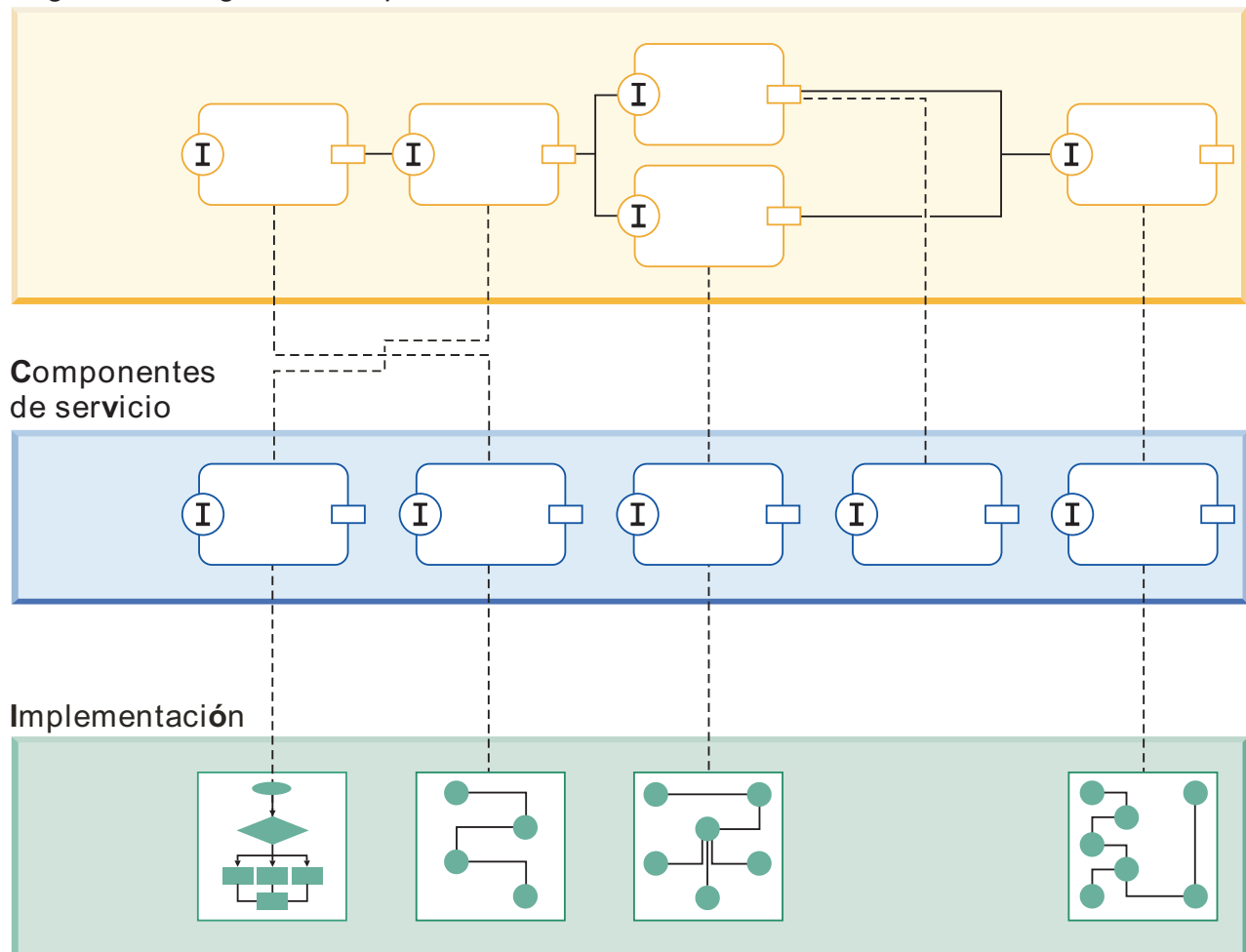
La arquitectura orientada a servicios, a la que se suele denominar simplemente SOA, es una estándar de la industria definido de forma somera, que presenta todos los procesos de negocio - Servicios web, activos de servicio de Sistemas de información de empresa (EIS), flujos de trabajo, bases de datos, ... - en una forma orientada a servicios. Esto significa que las dependencias entre los servicios se minimizan y que la implementación de dichos servicios queda oculta.

El objetivo de la arquitectura de componentes de servicio consiste en separar la lógica de integración de negocio de la implementación de forma que un desarrollador pueda centrarse en ensamblar una aplicación integrada en lugar de en los detalles de la implementación. Para conseguir este objetivo, se crean los componentes de servicio que contienen la implementación de servicios individuales necesarios para los procesos de negocio. El resultado es una arquitectura de tres capas, lógica de integración de negocio, componentes de servicio e implementación, tal como se muestra en el diagrama siguiente.



Puesto que los componentes de servicio contienen la implementación, el desarrollador de integración puede ensamblarlos gráficamente sin el conocimiento de detalles de implementación. Los componentes de servicio también proporcionan la opción de permitir que el desarrollador de integración o alguien que trabaje para él, añada posteriormente la implementación. Tal como verá en WebSphere Integration Developer, los componentes se ensamblan visualmente. En otras palabras, no se muestra el código que hay dentro de los componentes. En el nivel de lógica de negocio que se muestra en el diagrama siguiente, los componentes se ensamblan independientemente de la implementación. Entonces, la arquitectura orientada a servicios, le permite centrarse en la resolución de los problemas de negocio utilizando y reutilizando componentes en lugar de distraer su atención hacia la tecnología que implementa los servicios que está utilizando.

Lógica de integración empresarial



¿Cuales son las ventajas clave de la arquitectura orientada a servicios?

Con el aumento del ritmo y la naturaleza impredecible de los negocios actuales, una arquitectura orientada a servicio proporciona la agilidad necesaria para reacción y capitalización, en las cambiantes situaciones de los negocios. También ofrece un software sostenible, a lo largo del tiempo. Logra estos objetivos gracias a los puntos siguientes:

- Consolidación de lógica empresarial y datos de negocio. Los componentes utilizados por varios grupos de una empresa o incluso compartidos por un conjunto de empresas pueden ser utilizados s por cualquiera, siempre que los componentes cumplan estándares de la industria, como el Lenguaje descriptivo de servicios web (Web Services Descriptive Language - WSDL) y el Lenguaje de ejecución de procesos de negocio (Business Process Execution Language - BPEL), que son independientes de la plataforma y del proveedor. Los datos se representan de forma consistente, permitiendo que los mismos se compartan entre los componentes que componen una aplicación de arquitectura orientada a servicio.
- Aprovechamiento de aplicaciones y sistemas existentes. Al colocar las aplicaciones y sistemas dentro de código WSDL, pasan a estar disponibles de forma universal para cualquier desarrollador de aplicaciones de la empresa que esté actualmente desarrollando una aplicación.

¿Cuáles son los problemas de diseño fundamentales de una arquitectura orientada a servicio?

- Los componentes están poco cohesionados. Un componente que accede a otro, no precisa conocer las estructuras de datos, las llamadas a otros componentes, la gestión de transacciones ni muchos otros detalles del otro componente.
- Los componentes son configurables. Observar una arquitectura orientada a servicio, como la del diagrama anterior, es parecido a observar un diagrama de configuración. Los componentes se pueden añadir, suprimir y configurar de distintas formas, creando aplicaciones nuevas.
- Los componentes son interoperables. Cualquier componente puede interoperar con otro componente, incluyendo los componentes creados por entornos de desarrollo de otros proveedores.
- Los componentes son independientes de la ubicación. De la misma forma que un componente no sabe ni le hacen falta los detalles de implementación de otro componente, tampoco le es de relevancia su ubicación.

Juntos, estos principios de diseño crean una arquitectura flexible capaz de adaptarse y mejorar rápidamente ajustándose a las condiciones cambiantes del negocio.

Capítulo 4. Arquitectura de componentes de servicio

Service Component Architecture que habilita una arquitectura orientada a servicios está dispuesto por muchas empresas, incluyendo IBM. Service Component Architecture (SCA) es un modelo de programación independiente de plataforma y proveedor que proporciona un medio simple y coherente para expresar los datos de negocio y la lógica de negocio como servicios SOA, independientemente de los detalles de implementación técnica. En esta sección, examinamos servicios SCA y objetos de datos de servicio tal como se realizan en WebSphere Integration Developer.

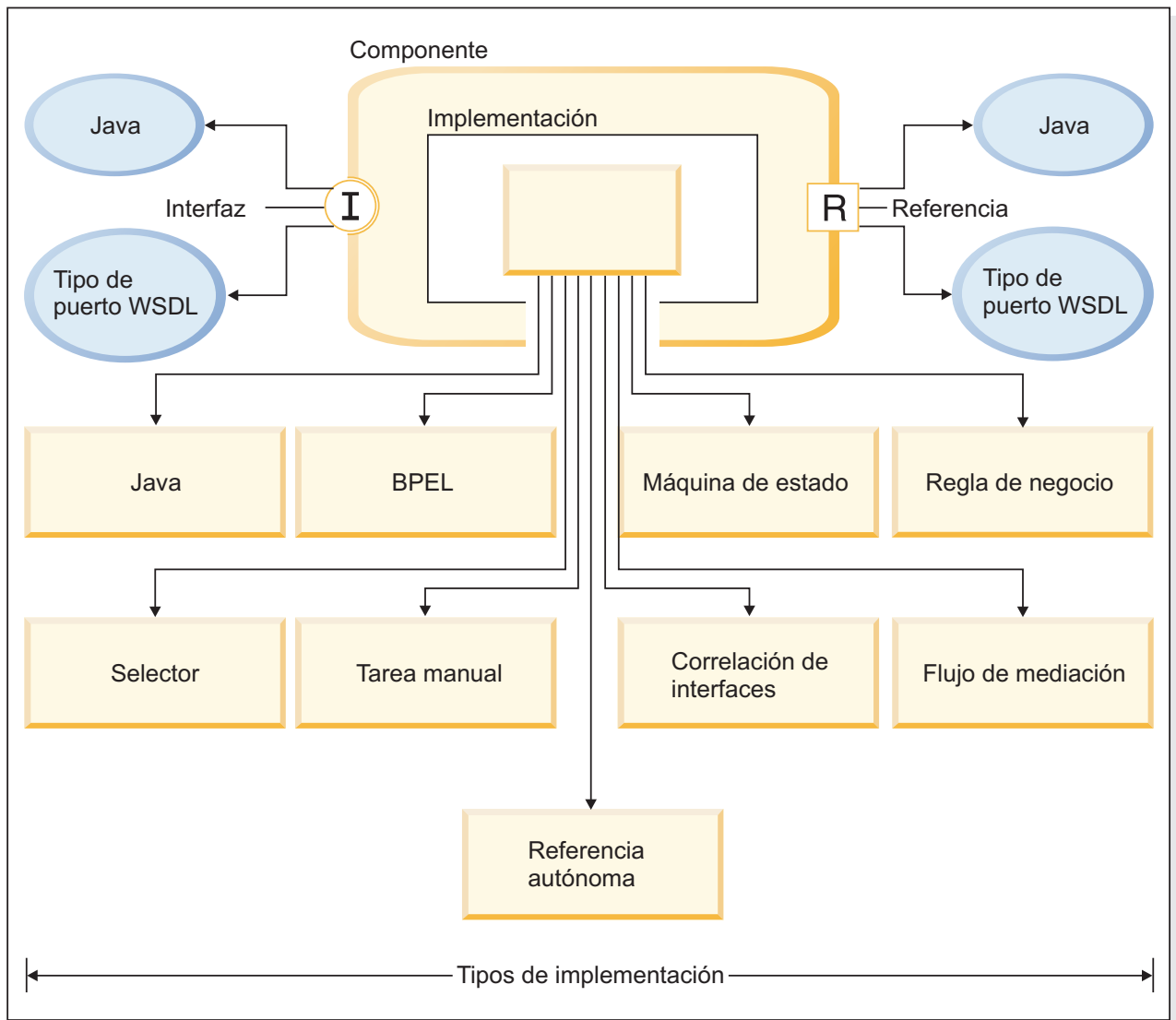
En la Especificación de la arquitectura de componentes de servicio hay más detalles y las últimas actualizaciones de esta arquitectura tan extendida.

Componentes de servicio

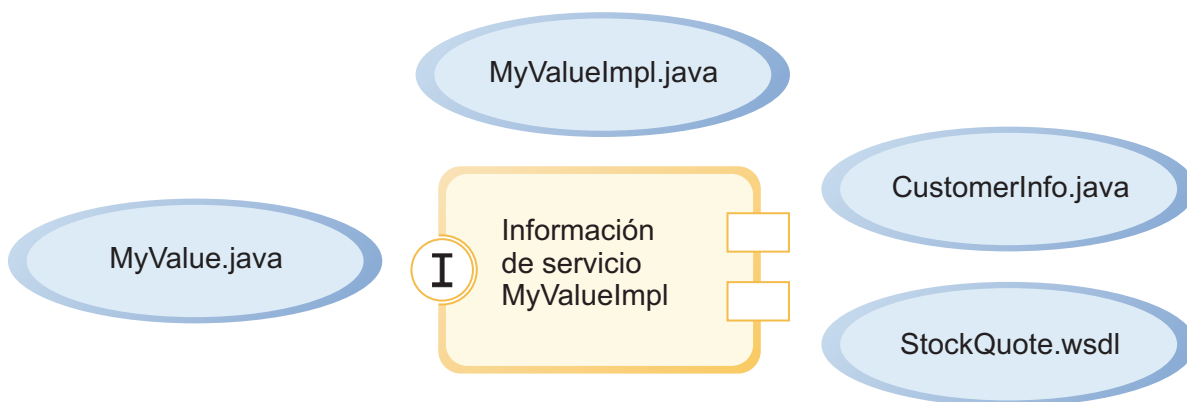
Un componente de servicio configura una implementación de servicio. Un componente de servicio se presenta en un diagrama de bloque estándar.

Un componente consta de una implementación que se oculta cuando se utilizan las herramientas de WebSphere Integration Developer, una o varias interfaces que definen las entradas, salidas y faltas correspondientes y varias referencias o ninguna. Una referencia identifica la interfaz de otro servicio o componente que este componente necesita o consume. Una interfaz puede definirse en dos lenguajes: un tipo de puerto WSDL o Java. Una interfaz soporta estilos de interacción síncronos y asíncronos. La implementación de un componente puede estar en varios lenguajes.

El tipo de interfaz recomendada es WSDL; las guías de aprendizaje y los ejemplos utilizan de forma coherente el tipo de interfaz WSDL. Una interfaz Java, sin embargo, está soportada y se utiliza sobre todo cuando se importa un EJB de sesión sin estado (esto se trata posteriormente en la sección “Importaciones y exportaciones” en la página 21). Incluso si desarrolla un componente Java descendente, es decir, define un componente y añade la implementación Java más adelante, debería poder utilizar una interfaz WSDL. No puede mezclar componentes basados en una interfaz WSDL con componentes basados en una interfaz Java.



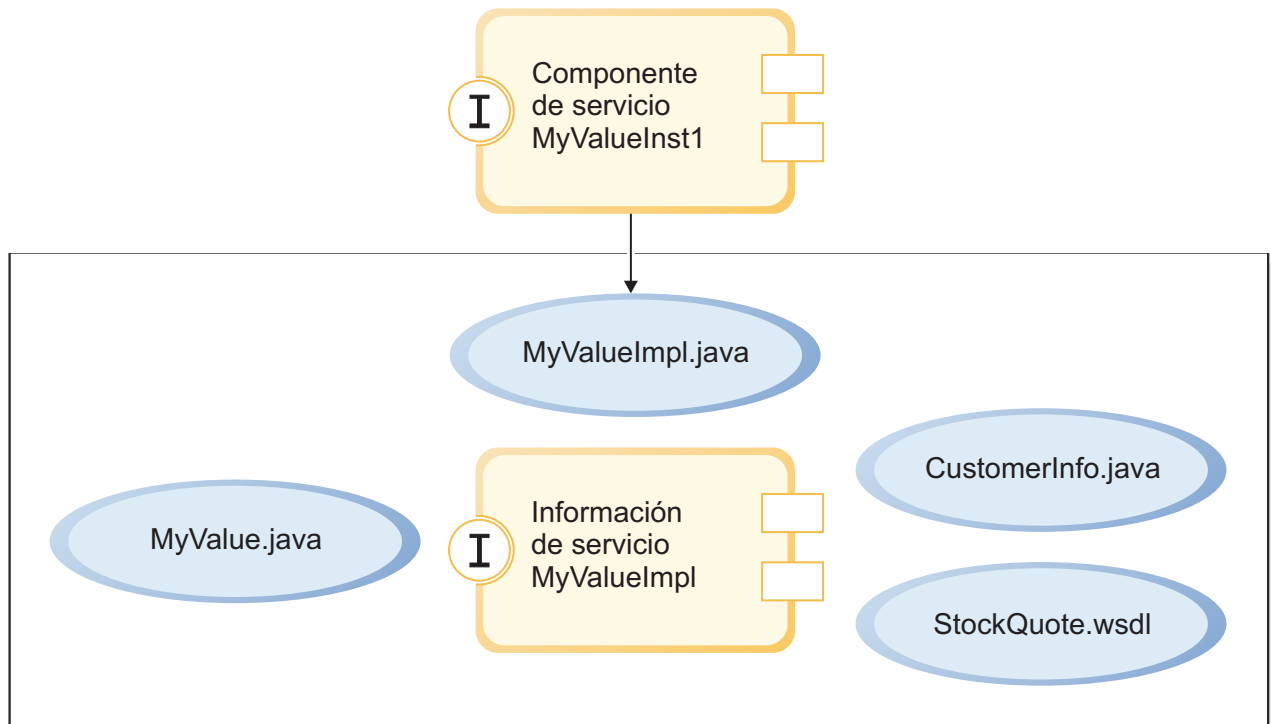
En la imagen siguiente, tenemos un componente en el centro. La implementación correspondiente MyValueImpl, está en Java como su interfaz. Tiene dos referencias: otra interfaz Java y una interfaz WSDL.



Al trabajar con este componente, tal como se muestra a continuación, solo se ve el mismo componente. Una referencia a este componente por parte de otro componente se indicará visualmente mediante una

línea hacia la interfaz correspondiente. Una referencia a este componente se indica mediante una línea trazada desde el punto de referencia a la interfaz de otro componente. Una referencia representa un servicio que este componente consume. Al denominar una referencia y solo especificar la interfaz correspondiente, el autor de la implementación del componente puede dejar el enlace de esa referencia con un servicio real para un momento posterior. En ese momento posterior, el especialista de integración lo hará así conectando la referencia a la interfaz de otro componente o importación. Este acoplamiento suelto que permite diferir los enlaces y reutilizar las implementaciones, es una de las razones clave para utilizar la Arquitectura de componentes de servicio de WebSphere Integration Developer.

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



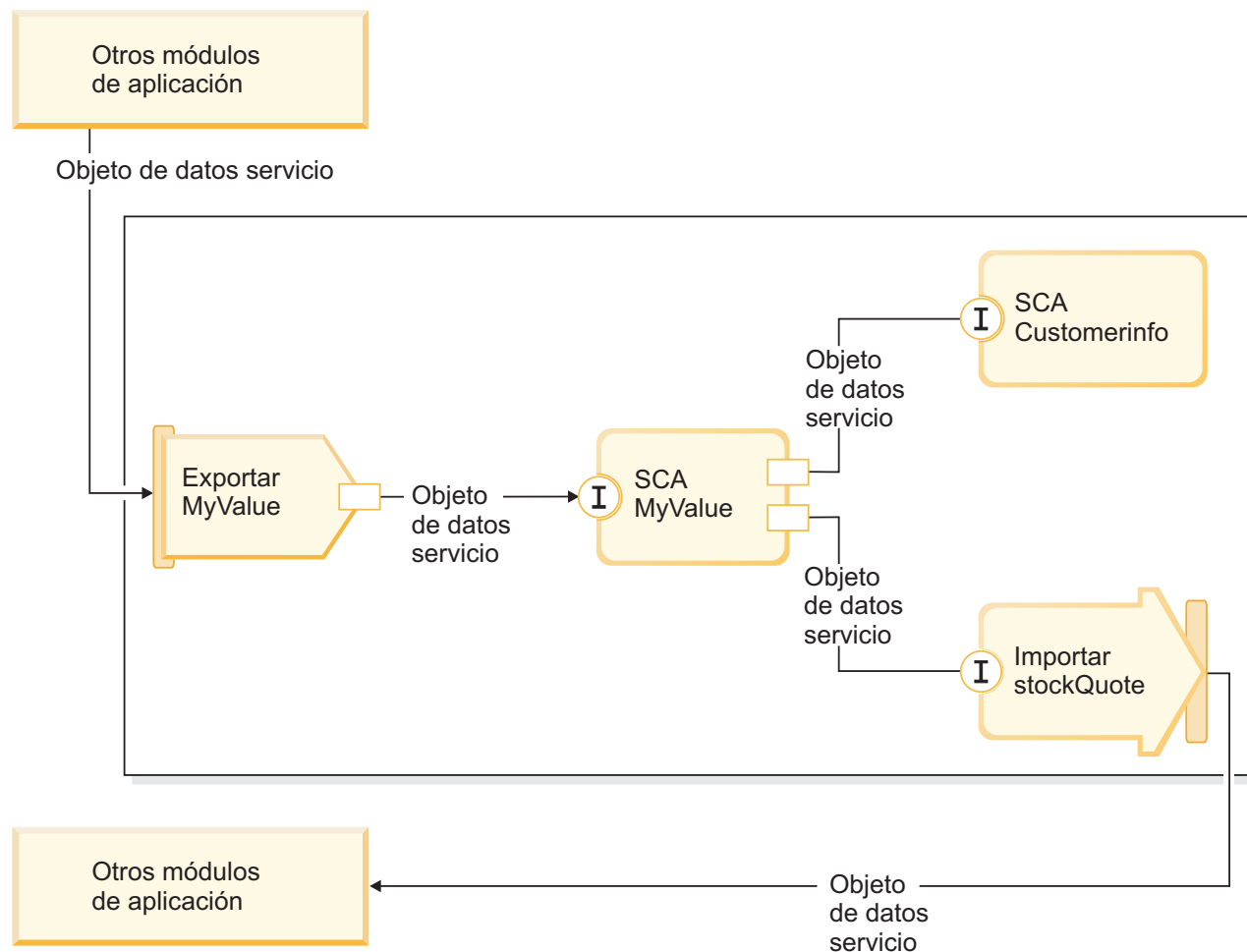
Objetos de datos de servicio

Los objetos de datos de servicio complementan la Arquitectura de componentes de servicio. La Arquitectura de componentes de servicio define los servicios como los componentes y la conectividad entre ellos. Los *Objetos de datos de servicio* definen los datos que fluyen entre componentes.

Cada componente pasa información como entrada y salida. Cuando se invoca un servicio, los objetos de datos se pasan como un documento XML con codificación de literales de documento al utilizar un tipo de puerto WSDL o como un objeto Java al utilizar una interfaz Java. Los objetos de datos son la forma preferida para datos y metadatos en los servicios de Arquitectura de componentes de servicio. Al igual que los componentes, los objetos de datos de servicio separan el objeto de datos de la implementación correspondiente. Por ejemplo, un componente interactúa con las ordenes de compra mientras que la orden de compra puede utilizar JDBC, EJB, etc. para realizar las actualizaciones en los datos. Los objetos de datos de servicio 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. Están definidos por una JSR (Petición de especificación Java de objetos de datos de servicio).

En el diagrama siguiente, los objetos de datos de servicio se pasan de un servicio externo a una exportación, de una exportación a un componente, de un componente a un componente, de un

componente a una importación y de una importación a un servicio. Las importaciones y exportaciones se tratan en la sección de importaciones y exportaciones.



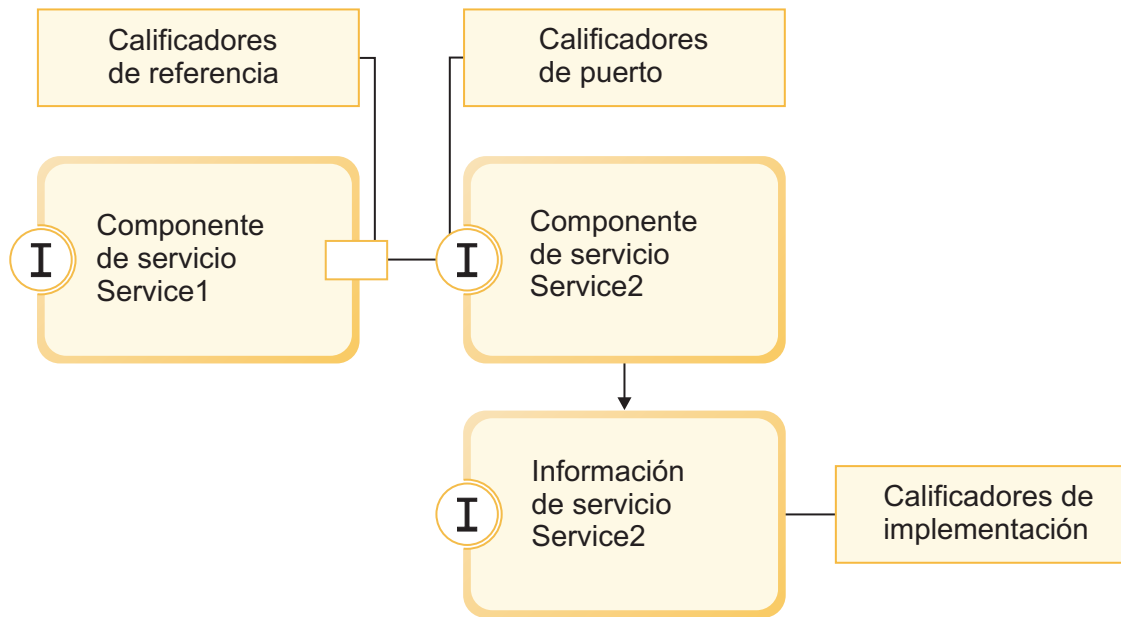
Calificadores de servicio

Una aplicación comunica sus necesidades de calidad de servicio (QoS) al entorno de tiempo de ejecución especificando *calificadores de servicio*. Estos gobiernan la interacción entre un cliente de servicio y un servicio destino.

Pueden especificarse calificadores en referencias a componentes de servicio, interfaces e implementaciones. Puesto que la declaración de los valores de QoS es externa a una implementación, puede cambiar estos valores sin cambiar la implementación o establecerlos de manera diferente cuando se utilizan varias instancias de la misma implementación en distintos contextos.

Estas 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 el permiso
- Fiabilidad asíncrona: reglas para la entrega de mensajes asíncrona



Módulos

Un *módulo* es una unidad de despliegue que determina qué artefactos se empaquetan en un archivo EAR. Los componentes de un módulo se colocan para el rendimiento y pueden pasar sus datos por referencia. Un módulo se puede establecer como mecanismo de ámbito, es decir: establece un límite organizacional para los artefactos.

Un módulo es un compuesto de componentes de servicio, importaciones y exportaciones. Los componentes de servicio, las importaciones y las exportaciones residen en los mismos proyecto y carpeta raíz que también contienen la conexión que enlaza los componentes y los enlaces necesarios para las importaciones y exportaciones. Un módulo también puede contener las implementaciones y las interfaces a las que hacen referencia los componentes, las importaciones y las exportaciones correspondientes o estos pueden colocarse en otros proyectos, como por ejemplo un proyecto de biblioteca.

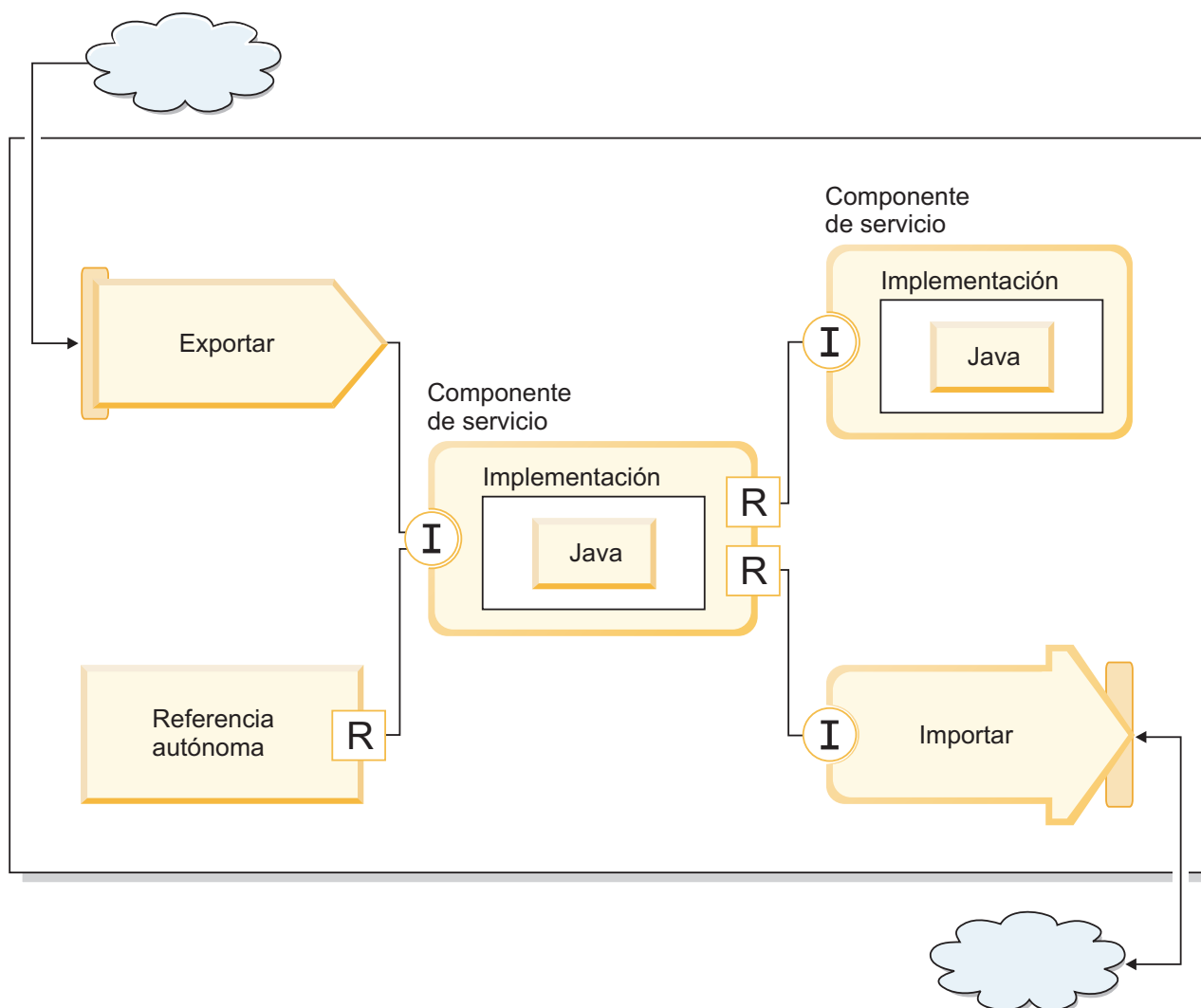
Hay dos tipos de módulos. En primer lugar, un módulo llamado *módulo* (a veces llamado módulo de integración de negocio) que contiene una opción de muchos tipos de componente y que a menudo se utiliza para soportar un proceso de negocio. En segundo lugar, un módulo llamado *módulo de mediación* que contiene hasta un componente, uno o varios componentes de flujo de mediación, además de varios componentes Java o ninguno que aumentan el componente de flujo de mediación.

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

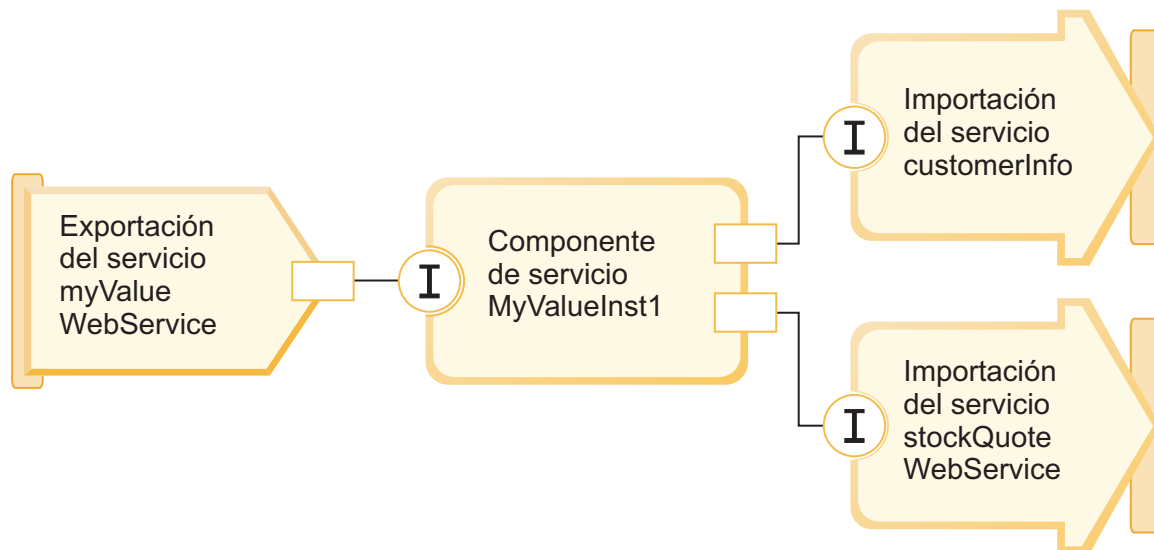
¿Por qué hay dos tipos de módulo? El primer tipo de módulo está principalmente diseñado para los procesos de negocio. Un módulo de mediación es como una pasarela a servicios externos existentes que es común en las arquitecturas de Enterprise Service Bus. A estos servicios externos o exportaciones en un módulo de mediación acceden las importaciones o los proveedores de servicios. Al desacoplar los peticionarios de servicio de cliente de los proveedores de servicio mediante un flujo de mediación, las aplicaciones ganan flexibilidad y elasticidad, un objetivo de la arquitectura orientada a servicios. Por ejemplo, el flujo de mediación puede anotar mensajes entrantes, direccionar mensajes a un servicio específico determinado en tiempo de ejecución o transformar datos para adecuarlos a otro servicio. Estas funciones pueden añadirse y cambiarse con el tiempo sin modificar los servicios peticionario o suministrador.

Un módulo resulta en la prueba y despliegue de una aplicación de servicio en WebSphere Process Server. Un módulo de mediación resulta en la prueba y despliegue de una aplicación de servicio en el servidor WebSphere Process Server o WebSphere Enterprise Service Bus. Ambos tipos de módulos soportan importaciones y exportaciones.

A menudo es necesario compartir entre módulos las implementaciones, las interfaces, los objetos de negocio, las correlaciones de objetos de negocio, los cometidos, las relaciones y otros artefactos que necesitan compartirse entre módulos. Una *biblioteca* es un proyecto utilizado para almacenar estos recursos compartidos.



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



Los artefactos de módulo y módulo de mediación incluyen lo siguiente:

- Definición de módulo: define el módulo.
- Componentes de servicio: definiciones de los servicios del módulo. El nombre de un 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 útil para un usuario.
- Importaciones: definiciones de importaciones que son llamadas a servicios externos a este módulo. Las importaciones tienen enlaces que se tratan en la sección Importaciones y exportaciones.
- Exportaciones: definiciones de exportaciones que se utilizan para exponer componentes a llamantes externos a este módulo. Las exportaciones tienen enlaces que se tratan en la sección Importaciones y exportaciones.
- Referencias: referencias de un componente a otro en el módulo.
- Referencias autónomas: hace referencia a aplicaciones que no están definidas como componentes de Arquitectura de componentes de servicio (por ejemplo JavaServer Pages), que habilitan estas aplicaciones para interactuar con los componentes de Arquitectura de componentes de servicio. Solo pueden haber un artefacto de referencia autónoma por módulo.
- Otros artefactos: estos artefactos incluyen archivos WSDL, clases Java, archivos XSD, procesos BPEL, etc.

Importaciones y exportaciones

Las importaciones y las exportaciones definen los puntos de acceso o las interfaces externas de un módulo. Las *importaciones* identifican servicios fuera de un módulo para que puedan llamarse desde el módulo. Las *exportaciones* permiten a los componentes proporcionar los servicios a clientes externos. La información de enlace es necesaria para una importación o una exportación. Hay varios enlaces disponibles y se proporciona algún consejo sobre qué tipo puede ser adecuado para la aplicación.

Tipos de enlace de importación y exportación de servicios

Las importaciones y exportaciones requieren información de enlace que especifica el significado de transportar los datos desde los módulos. Un *enlace de importación* describe la forma específica en que un servicio externo está enlazado a un componente de importación. Un *enlace de exportación* describe la forma específica en que los servicios de un módulo se ponen a disposición de los clientes.

El enlace SCA o enlace predeterminado permite al servicio comunicarse con otros servicios en otros módulos. Una importación con un enlace SCA permite acceder a un servicio en otro módulo. Una

exportación con un enlace SCA permite ofrecer un servicio a otros módulos. Un enlace de importación de servicio Web permite enlazar un servicio Web externo a una importación.

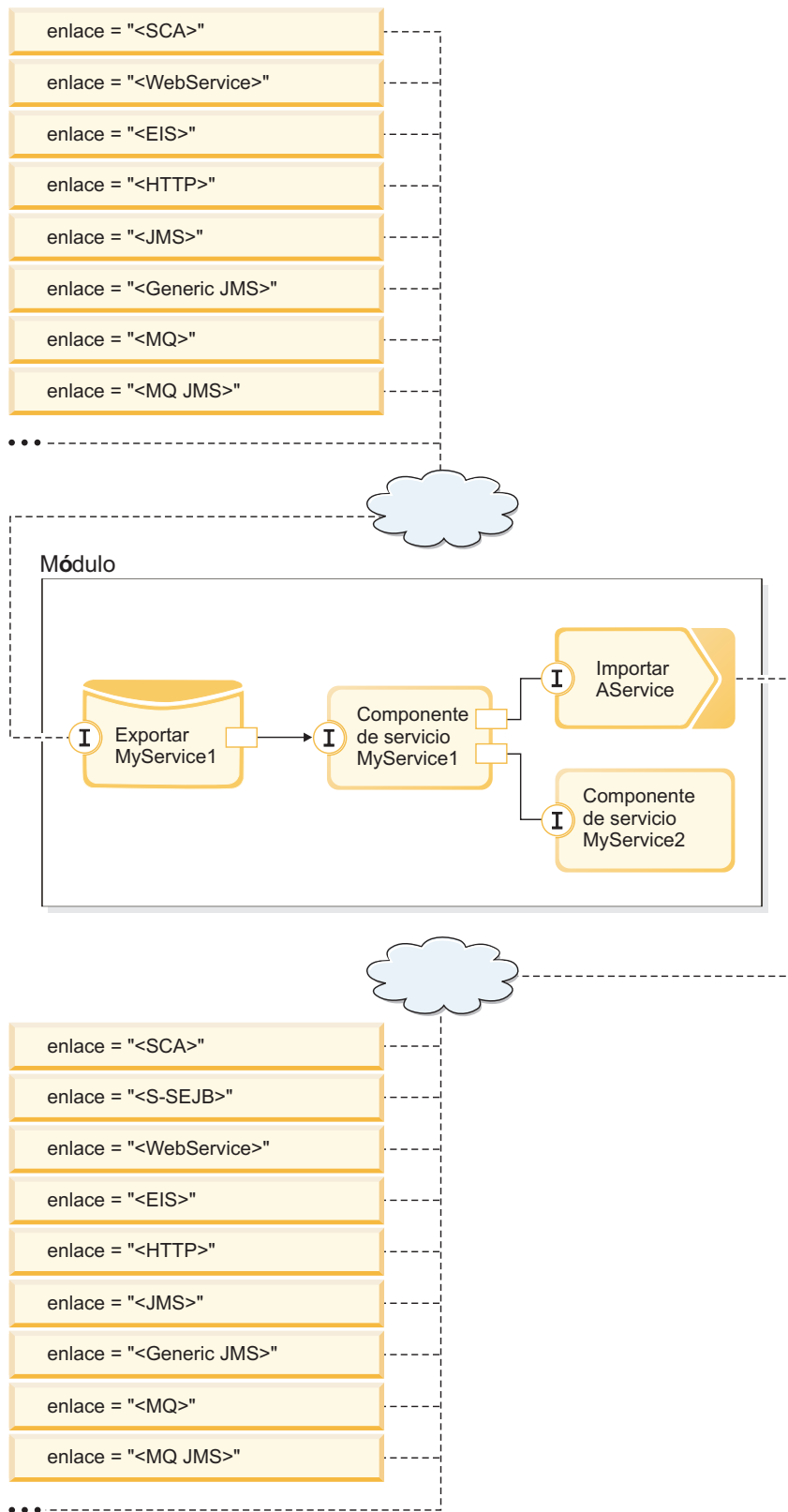
Un enlace de exportación de servicio Web proporciona un servicio a clientes externos como un servicio Web. El enlace de servicio Web puede utilizar un protocolo de transporte que sea SOAP/HTTP (SOAP por HTTP) o SOAP/JMS (SOAP por JMS).

Observe que WebSphere Integration Developer soporta varios tipos de enlaces de datos JMS: JMS, MQ JMS y JMS genérico. El único tipo de enlace de datos JMS soportado con el enlace de servicios Web es JMS. Por lo tanto, el único protocolo de transporte soportado es SOAP/JMS. Los protocolos no soportados son: SOAP/MQ JMS y SOAP/JMS genérico. El protocolo de transporte SOAP/MQ (nativo) tampoco está soportado.

El enlace del Protocolo de transferencia de Hipertexto (HTTP), a diferencia del enlace de servicio web anterior, presupone que el enlace funciona con aplicaciones HTTP nativas, y muestra un modelo más familiar a estos usuarios.

El asistente de servicio externo crea, importa y exporta representando un servicio en un sistema EIS. Los enlaces creados son de tipo EIS. Un enlace EIS proporciona comunicación síncrona con el servicio en el sistema EIS. Los enlaces de Java Message Service (JMS), JMS genérico, WebSphere MQSeries (MQ) y WebSphere MQSeries JMS (MQ JMS) se utilizan para interacciones con sistemas de mensajería, donde la comunicación asíncrona a través de colas de mensajes es de vital importancia para la fiabilidad. Una importación (aunque no una exportación) también puede tener un enlace EJB de sesión sin estado.

El editor de ensamblajes lista los enlaces soportados y simplifica la creación de los mismos cuando desea crear una importación o una exportación. Una vista de propiedades en el editor de ensamblaje muestra la información de enlace de cualquier importación o exportación.



Seleccionar enlaces adecuados

En esta sección se discute la idoneidad de un enlace determinado para las necesidades de su aplicación.

Los enlaces disponibles en WebSphere Integration Developer proporcionan varias opciones. Esta lista ayuda a identificar cuándo un tipo de enlace puede ser más adecuado para las necesidades de su aplicación que otro tipo de enlace.

Elija un enlace *SCA* cuando se den los factores siguientes:

- Todos los servicios están contenidos en módulos de WebSphere Integration Developer; es decir, no hay servicios externos
- El rendimiento es importante
- Los módulos están estrechamente cohesionados.

Elija un enlace *Servicio Web* cuando se den los factores siguientes:

- Necesita acceder a un servicio externo a través de Internet o proporcionar un servicio a través de Internet
- Los servicios están poco cohesionados
- Se prefiere la comunicación síncrona; es decir, una petición procedente de un servicio puede esperar una respuesta procedente de otro.
- El protocolo de un servicio externo al que está accediendo o un servicio que desea proporcionar es SOAP/HTTP o SOAP/JMS.

Elija un enlace *HTTP* cuando se den los factores siguientes:

- Necesita acceder a un servicio externo por Internet o proporcionar un servicio por Internet y está trabajando con otros servicios Web basados en el modelo HTTP; es decir, está utilizando operaciones de interfaz HTTP de todas conocidas, como las de GET, PUT, DELETE, etcétera.
- Los servicios están débilmente acoplados
- Se prefiere la comunicación síncrona; es decir, una petición procedente de un servicio puede esperar una respuesta procedente de otro.

Elija un enlace *EIS* cuando se den los factores siguientes:

- Necesita acceder al servicio en un sistema EIS utilizando un adaptador de recursos
- El rendimiento es más importante que la fiabilidad; es decir, la transmisión de datos síncrona es preferible a la asíncrona.

Elija un enlace *JMS* cuando se den los factores siguientes:

- Necesita acceder a un sistema de mensajería
- Los servicios están débilmente acoplados
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.
- Nota: hay varios tipos de enlaces JMS. Si está utilizando el enlace de servicios Web, solo está soportado en protocolo de transporte SOAP/JMS. Vea: “Tipos de enlace de importación y exportación de servicios” en la página 21.

Elija un enlace *JMS genérico* cuando se den estos factores:

- Necesita acceder a un sistema de mensajería de distribuidor no IBM
- Los servicios están débilmente acoplados
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.
- Nota: hay varios tipos de enlaces JMS. Si está utilizando el enlace de servicios Web, solo está soportado en protocolo de transporte SOAP/JMS. Vea: “Tipos de enlace de importación y exportación de servicios” en la página 21.

Elija un enlace *MQ* cuando se den los factores siguientes:

- Necesita acceder a un sistema de mensajería WebSphere MQ y tiene que usar las funciones MQ nativas
- Los servicios están débilmente acoplados
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.

Elija un enlace *MQ JMS* cuando se den estos factores:

- Necesita acceder a un sistema de mensajería WebSphere MQ, pero puede hacerlo dentro de un contexto JMS; es decir, el subconjunto de funciones JMS es lo suficientemente bueno para su aplicación
- Los servicios están débilmente acoplados
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.

Elija un enlace *EJB de sesión sin estado* cuando se den los factores siguientes:

- El enlace es para un servicio importado que es un EJB
- El servicio importado está débilmente acoplado
- El estado del EJB no es importante.

Tipos de implementación de servicio

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

Las implementaciones estándar de los servicios se describen en esta sección. Estas implementaciones aparecerán en los servicios del editor de ensamblaje y/o en procesos BPEL.

Objetos Java

A una implementación de un componente en Java se la llama objeto Java.

Una implementación común es un componente escrito en Java. Esta implementación a veces se llama POJO o "plain old Java object" (antiguo objeto Java simple). Generalmente, esta implementación tendrá un tipo de interfaz WSDL aunque esta implementación también puede tener una interfaz Java. Si hay varias interfaces especificadas, no puede mezclar interfaces WSDL con interfaces Java. Sin embargo 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 Bienvenida muestra cómo hacerlo.

Al trabajar con un objeto Java, el código permanece oculto dentro del contexto de los editores.

Un objeto Java puede utilizarse en un módulo de mediación. Puede desplegarse en un servidor WebSphere Process Server o un servidor WebSphere Enterprise Service Bus.

Proceso BPEL

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

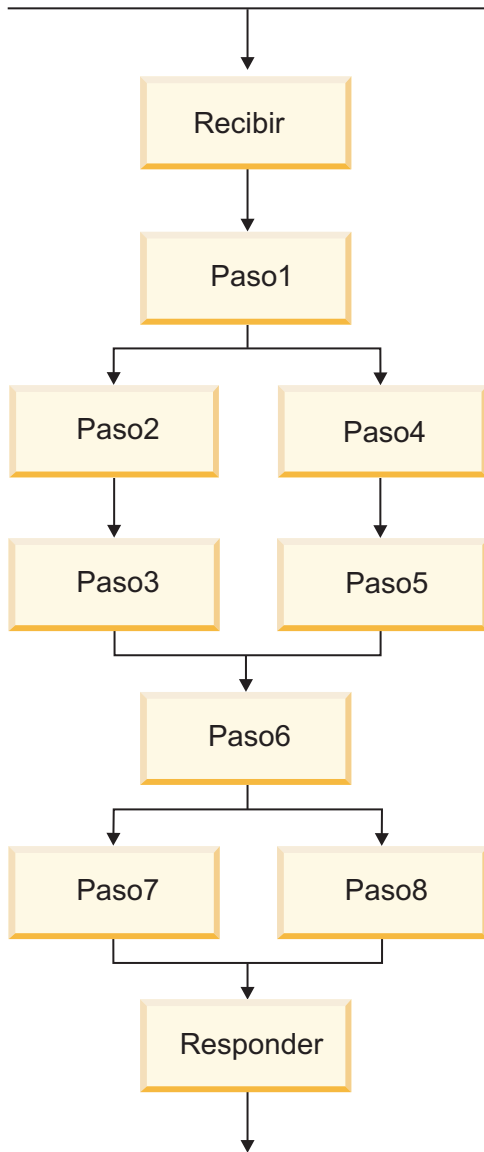
El lenguaje de implementación correspondiente es el estándar de la industria Business Process Execution Language for Web Services (BPEL4WS) y las extensiones de IBM correspondientes. Un proceso BPEL implementa un servicio de posible larga ejecución a través de la utilización de más servicios elementales. Un proceso BPEL creado en el editor de procesos puede hacer lo siguiente:

- Describir la orquestación de otros servicios que utilizan los gráficos de flujo de control
- Utilizar variables para guardar el estado de proceso
- Utilizar un manejo de errores sofisticado mediante el manejo de faltas
- Soportar eventos asíncronos

- Correlacionar peticiones de entrada con la instancia adecuada de un proceso determinado utilizando conjuntos de correlación para marcar los datos de negocio dentro de la petición que identifican a 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 personas en un proceso con el soporte de *tarea manual*. Por ejemplo, esta extensión puede añadir a un proceso la necesidad de que una persona apruebe un préstamo.

El editor de procesos utiliza representaciones visuales de las construcciones de BPEL para construir de forma rápida y sencilla el proceso de negocio.

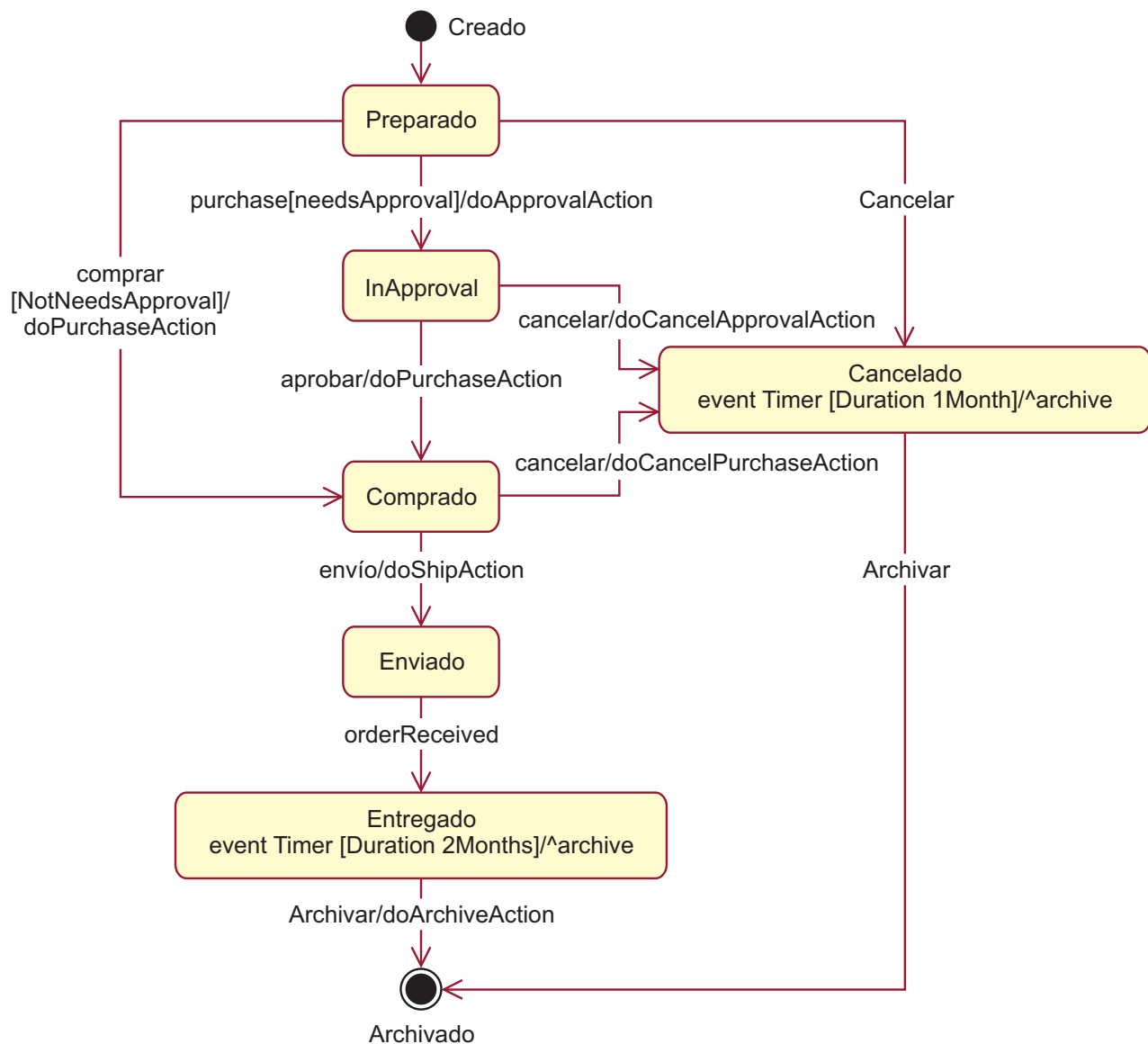


Un proceso BPEL no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

Máquinas de estado

Una máquina de estado es una forma alternativa de crear un proceso de negocio. Una máquina de estado está indicada para procesos relacionados con el cambio de estado en lugar de con un flujo de control. Un estado define lo que un artefacto puede hacer en un momento determinado. Una *máquina de estado* es una implementación de este conjunto de estados.

Las máquinas de estado son una forma común de mostrar un conjunto de estados interrelacionados en un proceso. Un expendededor de bebidas es una máquina de estado conocida. El usuario introduce monedas en la máquina y si todo va bien, conjuntamente con la bebida se obtiene el cambio exacto ya que la máquina determina las monedas a devolver teniendo en cuenta las monedas insertadas. En el diagrama siguiente, se muestra una máquina de estado típica creada por el editor de máquinas de estado. En la máquina de estado, se compra un elemento y se envía a un cliente.



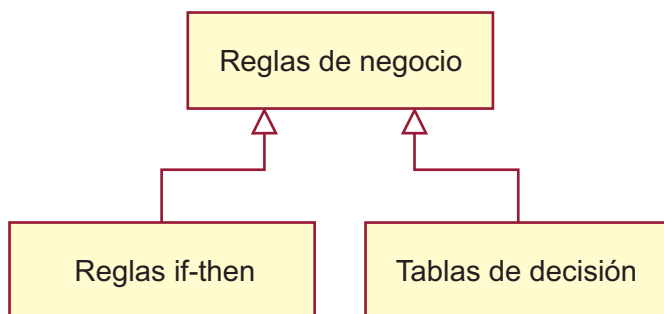
Una máquina de estado no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

Reglas de negocio

Las reglas de negocio complementan los procesos de negocio y las máquinas de estado. Si hay una condición con una variable, por ejemplo, una *regla de negocio* puede cambiar el valor en esa variable en tiempo de ejecución. Las reglas de negocio, creadas mediante un lenguaje de programación visual, toman decisiones basándose en el contexto. La decisión puede ser simple o compleja. Las reglas de negocio no son de procedimiento y las reglas pueden cambiarse independientemente de la aplicación.

Las reglas de negocio determinan la salida de un proceso basado en un contexto. Las reglas de negocio se utilizan en las situaciones de negocio cotidianas para tomar una decisión dado un conjunto de circunstancias específico. Esta decisión puede necesitar muchas reglas para cubrir todas las circunstancias. Las reglas de negocio dentro de un proceso de negocio permiten a las aplicaciones responder rápidamente a las condiciones cambiantes del negocio. Por ejemplo, en una empresa de seguros, una regla de negocio para aprobar el seguro de un coche de un solicitante podría ser: *Si el solicitante es hombre mayor de 25 años y la categoría del coche es deportivo y ha estado asegurado los últimos 5 años, aprobar la solicitud del seguro con una cuota de 100 \$ mensuales.*

WebSphere Integration Developer ofrece varios métodos para la creación de reglas de negocio. Puede crear reglas if-then o tablas de decisión, las cuales dan forma al resultado del proceso. Tenga en cuenta que estas reglas son independientes del proceso en sí mismo, lo que significa que puede cambiar las reglas en cualquier momento sin tener que rehacer el proceso. Por ejemplo, dependiendo de dónde esté ubicado el negocio, puede tener una regla que diga: *Si la fecha está entre el 26 de diciembre y el 1 de enero, ofrecer una descuento de ventas postvacacional del 20%.* Sin embargo, si las ventas siguen siendo bajas, podría modificar el descuento en cualquier momento por un 40%.

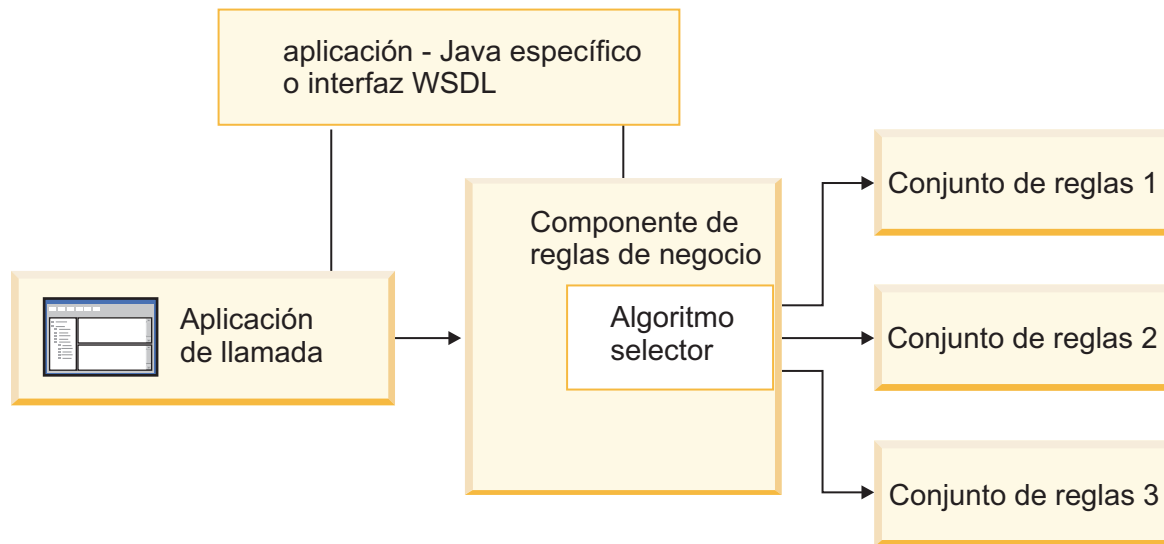


Las reglas de negocio no pueden utilizarse en un módulo de mediación. Solo pueden desplegarse en un servidor WebSphere Process Server.

Selectores

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

El direccionamiento a un componente está basado en las fechas. Por ejemplo, este es un direccionamiento basado en una fecha: *Dos semanas antes de iniciar el colegio, ofrecer un precio especial de vuelta a clase en nuestras mercancía escolares.* Los negocios pueden tener muchos direccionamientos basados en fechas. Un selector toma una decisión para seleccionar un direccionamiento sobre otro en tiempo de ejecución, basándose en una fecha. Por ejemplo, en época inmediatamente anterior al inicio del colegio, se llamaría a la oferta de vuelta al colegio indicada anteriormente. Sin embargo, en época cercana al final del curso escolar, puede haber una oferta para preparar a los niños para el verano.

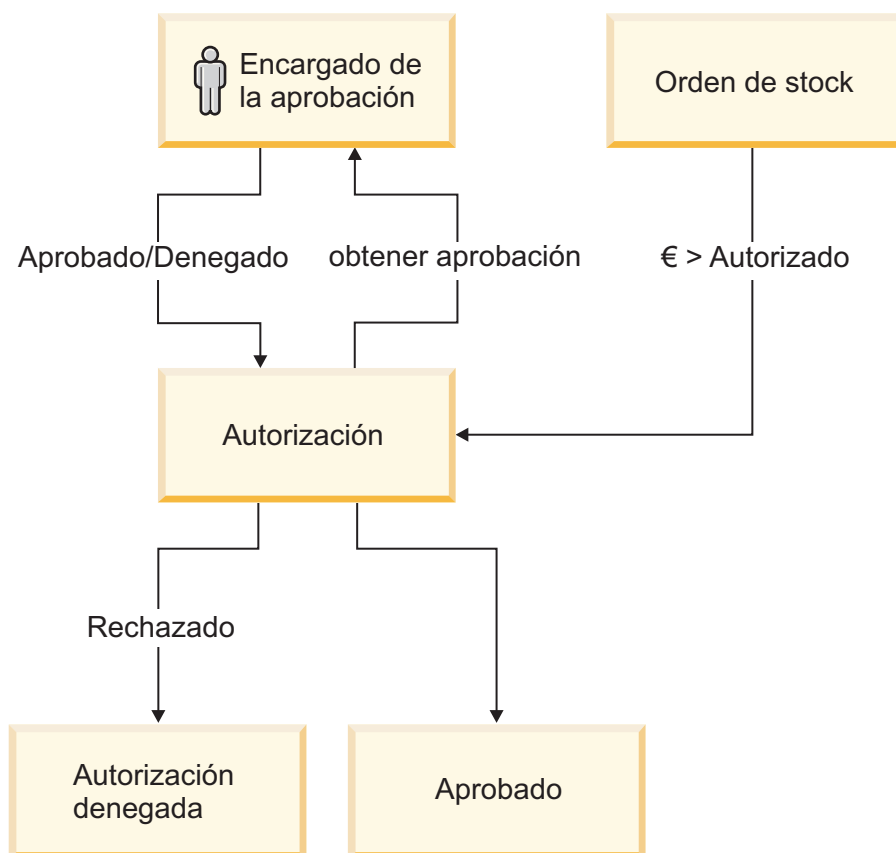


Un selector no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

Tarea manual

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

A veces, es necesario que las personas intervengan en un proceso de negocio. Por ejemplo, un cliente desea comprar un elemento que está por encima de su límite de crédito. Una tarea manual permite intervenir y alterar temporalmente una regla de negocio que impide al cliente hacer la compra. Una tarea manual puede tener atributos, como por ejemplo establecer el propietario de la tarea y proporcionar un proceso de escalada en el caso de que la persona especificada no esté disponible. El componente de tarea manual reconoce el hecho de que muchos procesos necesitan la intervención manual para tareas como la revisión, la investigación y la aprobación.



Una tarea manual no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

Correlación de interfaces

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

Las diferencias entre las interfaces de componentes que deben interactuar entre sí, son comunes. Estas diferencias surgen porque en WebSphere Integration Developer a menudo se ensamblan componentes creados para distintas aplicaciones. La reutilización para crear una aplicación nueva es uno de los puntos fuertes de WebSphere Integration Developer ya que de otro modo debería volver a codificar componentes parecidos. Sin embargo, es habitual tener que hacer algunos ajustes.

Por ejemplo, dos componentes pueden tener métodos que realizan básicamente la misma acción pero que tienen nombres distintos como por ejemplo `getCredit` y `getCreditRating`. También pueden tener nombres de operación distintos y las operaciones pueden tener tipos de parámetro distintos. Una correlación de interfaces correlaciona las operaciones y los parámetros de estos métodos para que las diferencias se resuelvan y los dos componentes puedan interactuar. Una correlación de interfaces es como un puente entre las interfaces de dos componentes, lo que permite conectarlos a pesar de las diferencias.

Una correlación de interfaces existe independientemente de los componentes que la utilizan, lo que significa que no es necesario cambiar los componentes.

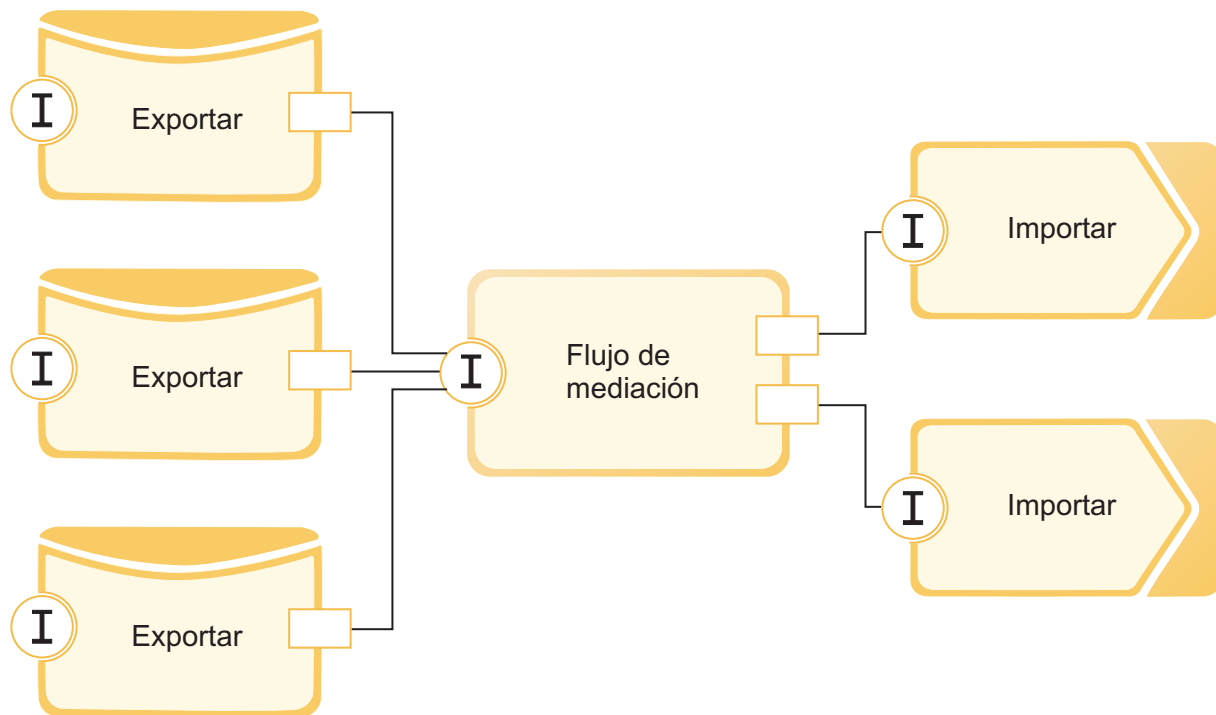
Una correlación de interfaces no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

Flujo de mediación

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

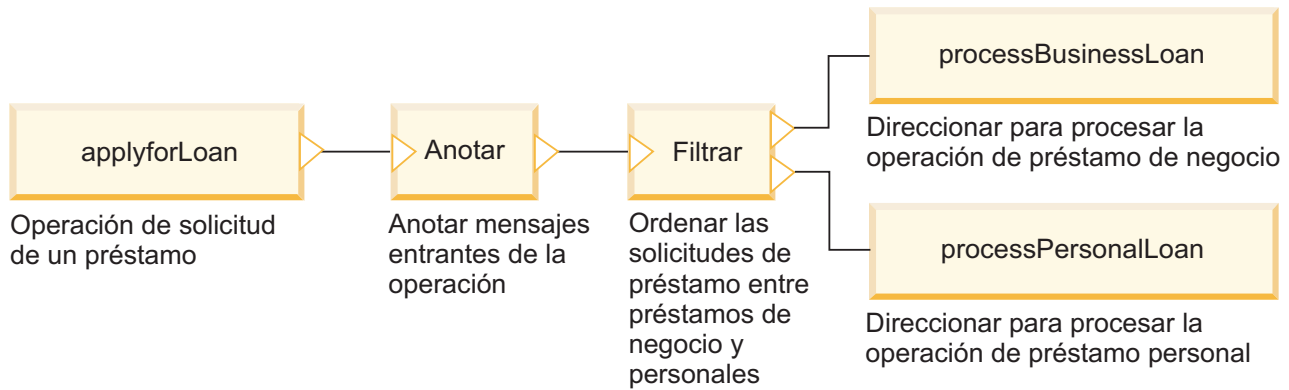
La mediación tiene varias funciones útiles. Por ejemplo, puede utilizar la mediación cuando necesite transformar datos de un servicio a un formato aceptable para otro servicio. La función de anotación le permite anotar mensajes de un servicio antes de enviarlos al servicio siguiente. El direccionamiento permite direccionar datos de un servicio a un servicio adecuado determinado por el flujo de mediación. Una mediación funciona independientemente 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 solicitadores 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 adecuados a dos proveedores de servicio o importaciones.



Un flujo de mediación es una construcción de estilo de flujo creada con el editor de flujos de mediación. Al seleccionar un componente de flujo de mediación en el editor de ensamblajes, se lanza el editor de flujos de mediación. En el editor de flujos de mediación, una operación de un servicio, el solicitador de servicio o exportación se correlaciona con la operación de otro servicio, el proveedor de servicio o importación, conjuntamente con las funciones proporcionadas por el editor de flujos de mediación. Estas funciones se llaman *primitivos de mediación* y se conectan en un flujo de mediación tal como se muestra en el diagrama siguiente. IBM proporciona los primitivos de mediación o también puede crear sus propios primitivos de mediación. Los primitivos de mediación pueden actuar sobre el contenido y el contexto del mensaje, donde el contexto es la información específica de enlace como por ejemplo cabeceras SOAP o JMS o propiedades definidas por el usuario.

En el diagrama siguiente, la operación `applyforLoan` envía un mensaje primero al primitivo de anotación `Log` que registra el mensaje. `Log` envía el mensaje al primitivo `Filter` que, dependiendo del mensaje, direcciona el mensaje a una operación `processBusinessLoan` o a una operación `processPersonalLoan`.



Tal como se indica en la sección Módulos, los flujos de mediación pueden estar en un módulo o en un módulo de mediación. Ambos tipos de módulos pueden contener uno o varios módulos componentes de flujo de mediación además de varios componentes Java o ninguno que aumenten el componente de flujo de mediación. Se puede desplegar un módulo en un servidor WebSphere Process Server. Un módulo de mediación puede desplegarse en un servidor WebSphere Process Server o un servidor WebSphere Enterprise Service Bus.

Referencias autónomas

Las *referencias autónomas* son referencias a aplicaciones que no se definen como componentes de Arquitectura de componentes de servicio (por ejemplo JavaServer Pages o servlets.) Las referencias autónomas permiten que estas aplicaciones interactúen con componentes de Arquitectura de componentes de servicio.

Las referencias autónomas no tienen una interfaz ni una implementación (ya que la implementación está fuera del ámbito del módulo.) Un módulo puede contener un artefacto de referencia autónoma o ninguno. Las referencias autónomas tienen el valor práctico de permitir la utilización de las aplicaciones existentes conjuntamente con los componentes de Arquitectura de componentes de servicio creados en WebSphere Integration Developer.

Las referencias autónomas pueden utilizarse en un módulo de mediación. Pueden desplegarse en un servidor WebSphere Process Server o un servidor WebSphere Enterprise Service Bus.

Capítulo 5. Soporte bidireccional

WebSphere Integration Developer funciona en un entorno multilingüe. Esto significa que puede visualizar y manejar datos representados en distintos idiomas. Esta sección trata del soporte para idiomas bidireccionales escritos de derecha a izquierda.

Soporte bidireccional

Algunos idiomas con scripts bidireccionales (por ejemplo árabe o hebreo) necesitan soporte bidireccional. Estos idiomas se escriben de derecha a izquierda, mientras que los números y los segmentos incorporados de texto latino (o cirílico o griego, etc.) se incorporan en este texto de izquierda a derecha.

En Overview of bidirectional script support in IBM WebSphere Integration Developer se trata el soporte de los idiomas bidireccionales, incluyendo la configuración necesaria y algunos puntos técnicos específicos de la utilización del soporte y las limitaciones.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los EE.UU.

IBM puede no ofrecer los productos, servicios o características tratados en este documento en otros países. Póngase en contacto con el representante local de IBM que le informará sobre los productos y servicios disponibles actualmente en su área. Las referencias hechas a productos, programas o servicio de IBM no pretenden afirmar ni dar a entender que únicamente puedan utilizarse dichos productos, programas o servicios de IBM. Puede utilizarse en su lugar cualquier otro producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes de aprobación que cubran temas descritos en este documento. La posesión de esta documentación no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas sobre las licencias, por escrito, a:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
EE.UU.*

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

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japón*

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país en que dichas disposiciones entren en contradicción con las leyes locales: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, NI EXPLÍCITA NI IMPLÍCITA, INCLUYENDO, PERO NO LIMITÁNDOSE, A LAS GARANTÍAS IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O ADECUACIÓN A UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no se aplique en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información incluida en este documento; estos cambios se incorporarán en nuevas ediciones de la publicación. IBM puede efectuar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Cualquier referencia incluida en esta información a sitios Web que no sean de IBM solo se proporciona para su comodidad y en ningún modo constituye una aprobación de dichos sitios Web. Los materiales de estos sitios web no forman parte de los materiales de IBM para este producto y el uso que se haga de estos sitios web es de la entera responsabilidad del usuario.

IBM puede utilizar o distribuir la información que usted le suministre del modo que IBM considere conveniente sin incurrir por ello en ninguna obligación para con usted.

Los licenciatarios de este programa que deseen obtener información acerca del mismo con el fin de: (i) intercambiar la información entre programas creados independientemente y otros programas (incluyendo este) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

*Intellectual Property Dept. for WebSphere Software
IBM Corporation
3600 Steeles Ave. East
Markham, Ontario
Canadá L3R 9Z7*

Esta información puede estar disponible, sujeta a los términos y condiciones adecuados, incluyendo en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia disponible para el mismo, lo proporciona IBM bajo los términos del Acuerdo de Cliente IBM, el Acuerdo de Licencia de Programa Internacional IBM o cualquier otro acuerdo equivalente entre ambas partes.

Los datos de rendimiento incluidos aquí se determinaron en un entorno controlado. Por lo tanto, los resultados que se obtengan en otros entornos operativos pueden variar significativamente. Tal vez se hayan realizado mediciones en sistemas que estén en fase de desarrollo y no existe ninguna garantía de que esas mediciones vayan a ser iguales en los sistemas disponibles en el mercado. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información concerniente a productos no IBM se ha obtenido de los suministradores de dichos productos, de sus anuncios publicados o de otras fuentes de información pública disponibles. IBM no ha comprobado dichos productos y no puede afirmar la exactitud en cuanto a rendimiento, compatibilidad u otras características relativas a productos no IBM. Las consultas acerca de las posibilidades de los productos no IBM deben dirigirse a las personas que los suministran.

Todas las declaraciones relativas a la dirección o la intención futura de IBM están sujetas a cambios o anulación sin previo aviso y tan solo representan metas y objetivos.

Todos los precios de IBM mostrados son precios de venta al detalle sugeridos por IBM, son actuales y están sujetos a cambio sin previo aviso. Los precios de los distribuidores pueden variar.

Esta información se proporciona solamente a efectos de planificación. La información contenida aquí está sujeta a cambios antes de que los productos descritos estén disponibles.

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

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente, que ilustran las 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 bajo todas las condiciones posibles. Por lo tanto, IBM no puede garantizar ni dar por sentada la fiabilidad, la facilidad de mantenimiento ni el funcionamiento de los programas.

Cada copia o cada parte de los programas de ejemplo o de los trabajos que se deriven de ellos debe incluir un aviso de copyright como se indica a continuación:

© (el nombre de su empresa) (el año). Algunas partes de este código se derivan de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _especifique el año o los años_. Reservados todos los derechos.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Información de interfaces de programación

La información de las interfaces de programación, si se proporciona, está destinada a ayudarle a crear software de aplicaciones mediante este programa.

Las interfaces de programación de uso general le permiten escribir software de aplicaciones que obtengan los servicios de las herramientas de este programa.

Sin embargo, aquí también puede haber información de diagnóstico, modificación y ajuste. La información de diagnóstico, modificación y ajuste que se proporciona está destinada a ayudarle a depurar el software de las aplicaciones.

Aviso: no utilice la información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

Marcas registradas y marcas de servicio

IBM, el logotipo de IBM e `ibm.com` son marcas registradas de International Business Machines Corporation en los Estados Unidos de América y/o en otros países. Si estos y otros términos de marcas registradas de IBM se marcan la primera vez que aparecen en esta información con un símbolo de marca registrada ([®] o [™]), estos símbolos indican marcas registradas por derecho o registradas en los EE.UU. propiedad de IBM en el momento de publicar esta documentación. Tales marcas registradas también pueden ser marcas registradas por derecho o marcas registradas en otros países. Encontrará una lista actual de marcas registradas de IBM en el sitio web “Copyright and trademark information”, en www.ibm.com/legal/copytrade.shtml.

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript son marcas registradas de Adobe Systems Incorporated en los Estados Unidos de América y/o en otros países.

Java y todas las marcas registradas y logotipos basados en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos y/o en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos de América y/o en otros países.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium y Pentium son marcas registradas de Intel o sus empresas subsidiarias en los Estados Unidos de América y en otros países.

UNIX es una marca registrada de The Open Group en los Estados Unidos de América y/o en otros países.

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de otros.

Términos de uso

El permiso para el uso de estas publicaciones se otorga sujeto a los términos y condiciones siguientes.

Uso personal: puede reproducir estas publicaciones para su uso personal, no comercial, y siempre que se respeten todos los avisos de propiedad. No puede distribuir, mostrar o realizar trabajos derivados de estas publicaciones, ni de ninguna parte de ellas, sin el consentimiento expreso de IBM.

Uso comercial: puede reproducir, distribuir y mostrar estas aplicaciones únicamente dentro de la empresa, y siempre que se conserven todos los avisos de propiedad. No podrá hacer trabajos derivados de estas publicaciones, ni reproducir, distribuir o mostrar estas publicaciones (ni parte de las mismas) fuera de la empresa, sin consentimiento expreso de IBM.

Salvo lo aquí permitido de forma expresa, no se conceden otros permisos, licencias o derechos, ni implícitos ni explícitos, para las publicaciones o cualquier información, datos software u otra propiedad intelectual que en ellas se incluya.

IBM se reserva el derecho de cancelar los permisos aquí otorgados siempre que, según su propio criterio, el uso de las publicaciones pueda perjudicar sus intereses (según consideración de IBM) o las instrucciones anteriores no se sigan adecuadamente.

No puede descargar, exportar o re-exportar esta información salvo que cumpla completamente con todas las leyes y normas aplicables, incluyendo las leyes y normativas de exportación de Estados Unidos.

IBM NO GARANTIZA EL CONTENIDO DE ESTAS PUBLICACIONES. ESTAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL", SIN GARANTÍAS DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO PERO SIN LIMITARSE A LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD DE UN OBJETIVO CONCRETO.

© Copyright IBM Corporation 2005, 2008. Reservados todos los derechos.

Hoja de Comentarios

Integration Developer
Versión 6.2
Visión general técnica del producto
Versión 6 Release 2

Por favor, sírvase facilitarnos su opinión sobre esta publicación, tanto a nivel general (organización, contenido, utilidad, facilidad de lectura,...) como a nivel específico (errores u omisiones concretos). Tenga en cuenta que los comentarios que nos envíe deben estar relacionados exclusivamente con la información contenida en este manual y a la forma de presentación de ésta.

Para realizar consultas técnicas o solicitar información acerca de productos y precios, por favor diríjase a su sucursal de IBM, business partner de IBM o concesionario autorizado.

Para preguntas de tipo general, llame a "IBM Responde" (número de teléfono 901 300 000).

Al enviar comentarios a IBM, se garantiza a IBM el derecho no exclusivo de utilizar o distribuir dichos comentarios en la forma que considere apropiada sin incurrir por ello en ninguna obligación con el remitente.

Comentarios:

Gracias por su colaboración.

Envíelos por correo a la dirección indicada en el reverso.

Si desea obtener respuesta de IBM, rellene la información siguiente:

Nombre

Dirección

Compañía

Número de teléfono

Dirección de e-mail

IBM Canada Ltd. Laboratory
Information Development para WebSphere Integration Developer
8200 Warden Avenue
Markham, Ontario
Canadá



Impreso en España