

Versión 6.0.1



Visión general técnica del producto



Visión general técnica del producto

Nota

Antes de utilizar esta información y el producto al que hace referencia, lea la información que se incluye al final de esta publicación.

Contenido

Capítulo 1. Integración empresarial . . . 1

Integración entre unidades de negocio	2
Integración entre empresas	3
WebSphere Integration Developer	3
Estándares	5
Cometido del desarrollador de integración	5

Capítulo 2. Arquitectura de componentes de servicio 7

Componentes de servicio	8
Objetos de datos de servicio	10
Calificadores de servicio	11
Módulos	12
Importaciones y exportaciones	14
Tipos de enlace de importación y exportación de servicios	14
Seleccionar enlaces adecuados	16
Tipos de implementación de servicio	17
Objetos Java	17

Proceso BPEL	18
Máquinas de estado	19
Reglas de negocio	20
Selectores	21
Tarea manual	22
Correlación de interfaces	23
Flujo de mediación	24
Referencias autónomas	25
Información relacionada	25

Capítulo 3. Aprender acerca de las herramientas 27

Visión general de la vista Bienvenida	27
Guías de aprendizaje de la vista Bienvenida	28
Ejemplos de la vista Bienvenida	28
Centro de información	29

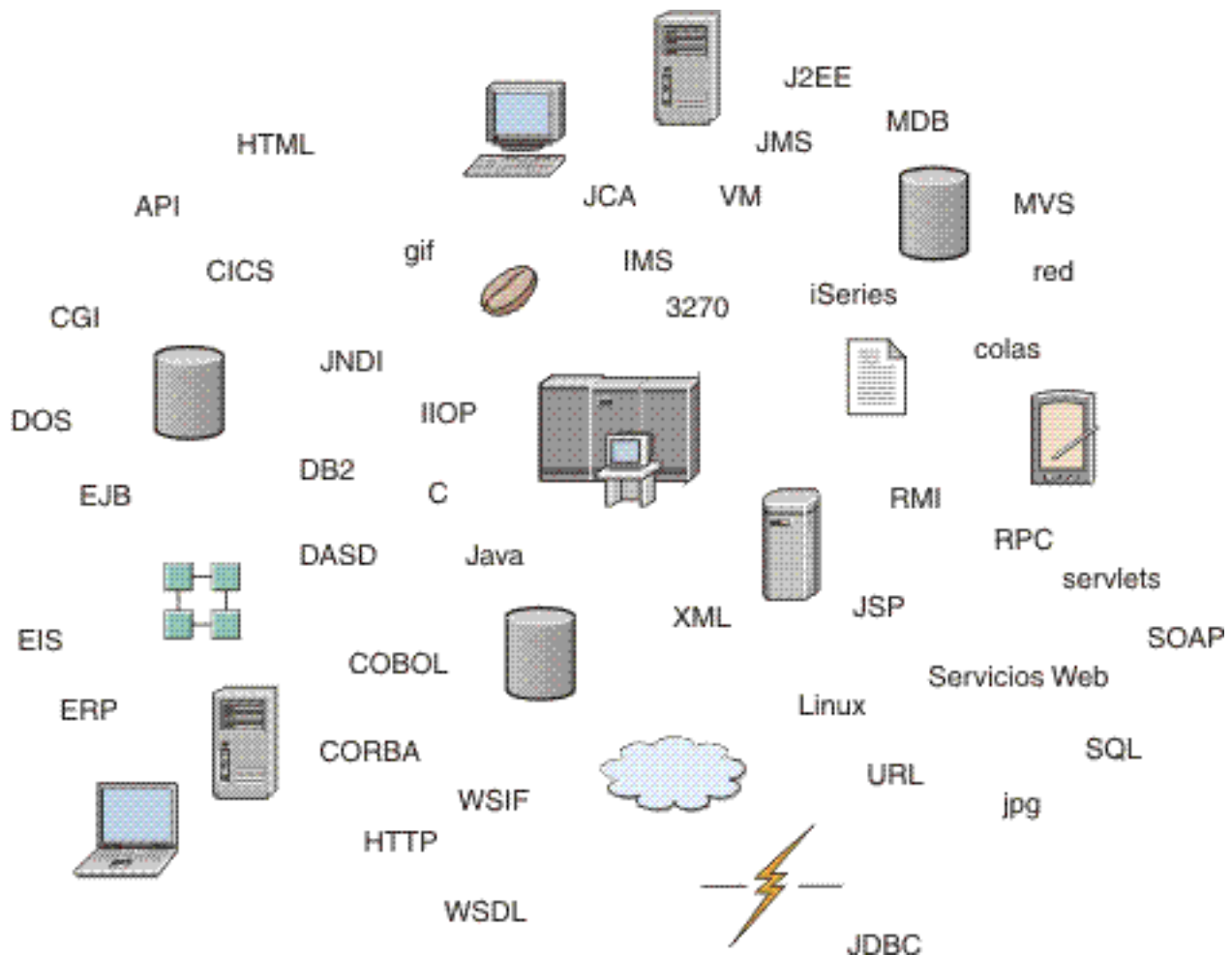
Avisos 31

Capítulo 1. Integración empresarial

Integración empresarial 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.

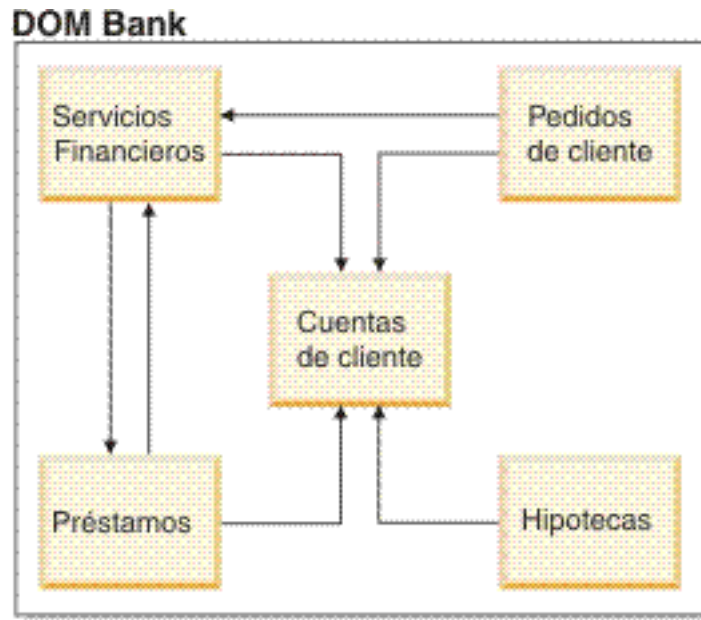
Examinemos las dos fuerzas que llevan el problema al punto crítico: la integración entre unidades de negocio dentro de una empresa y la integración entre empresas. A continuación observaremos cómo WebSphere Integration Developer trata estos temas y en particular el significado del compromiso con estándares 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.



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.

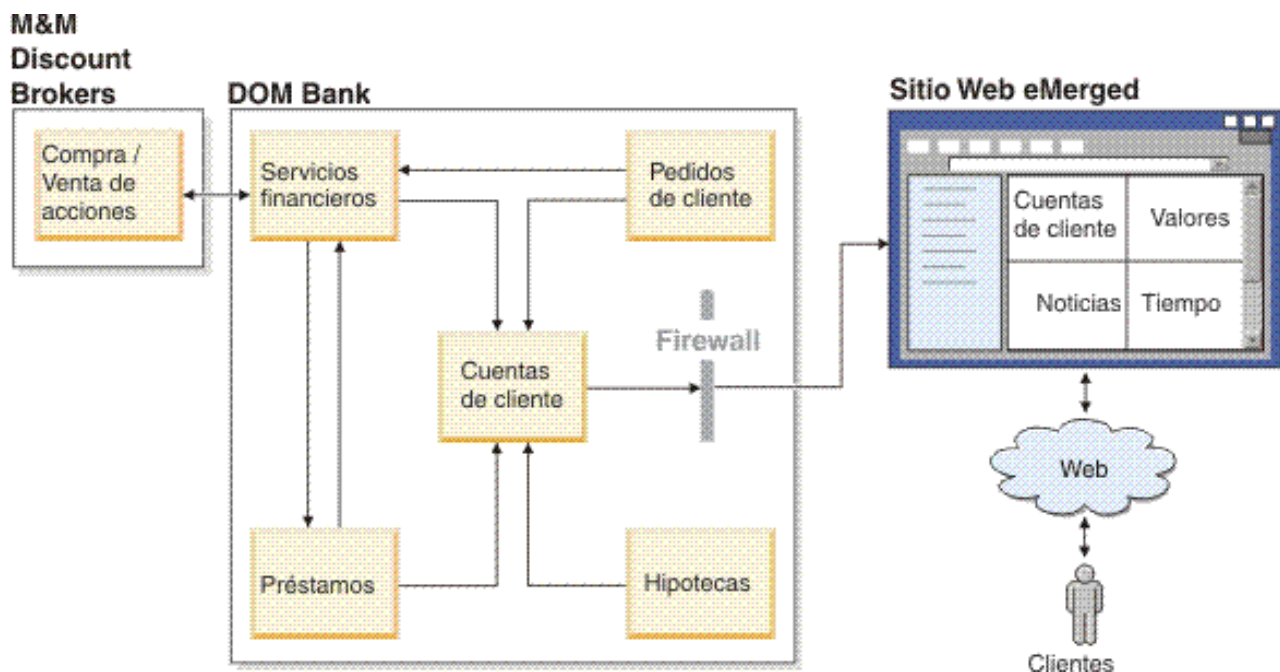
Sin embargo, las distintas empresas tienen historias también diferentes. Sus aplicaciones están codificadas en lenguajes diferentes en plataformas distintas que utilizan protocolos de comunicación diversos. Asimismo, existen mayores riesgos de seguridad al trabajar con empresas distintas. Independientemente 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.

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.

Cometido del desarrollador de integración

El desarrollador de integración es el usuario principal de WebSphere Integration Developer. Esta persona puede construir una aplicación integrada compleja utilizando las herramientas visuales sin tener un amplio conocimiento de la implementación subyacente.

WebSphere Integration Developer presenta aplicaciones y procesos de negocios como *componentes*. La implementación de los componentes permanece oculta y los componentes interaccionan a través de interfaces. Como resultado, los desarrolladores de integración no necesitan tener un amplio conocimiento de la implementación subyacente de los componentes para crear una aplicación integrada que los utilice. Sin embargo, los desarrolladores de integración tendrán un conocimiento técnico amplio del campo de integración ya que necesitan conocer algo de los sistemas EIS, los procesos de negocio y las aplicaciones codificadas en Java o en otros idiomas. Por ejemplo, un arquitecto tiene un conocimiento amplio del funcionamiento de un sistema sin saber detalladamente qué hace cada componente. Igual que un arquitecto, un desarrollador de integración puede ser la persona de una organización que diseñe la aplicación general y que tenga otros que codifiquen la implementación de los componentes específicos.

Capítulo 2. Arquitectura de componentes de servicio

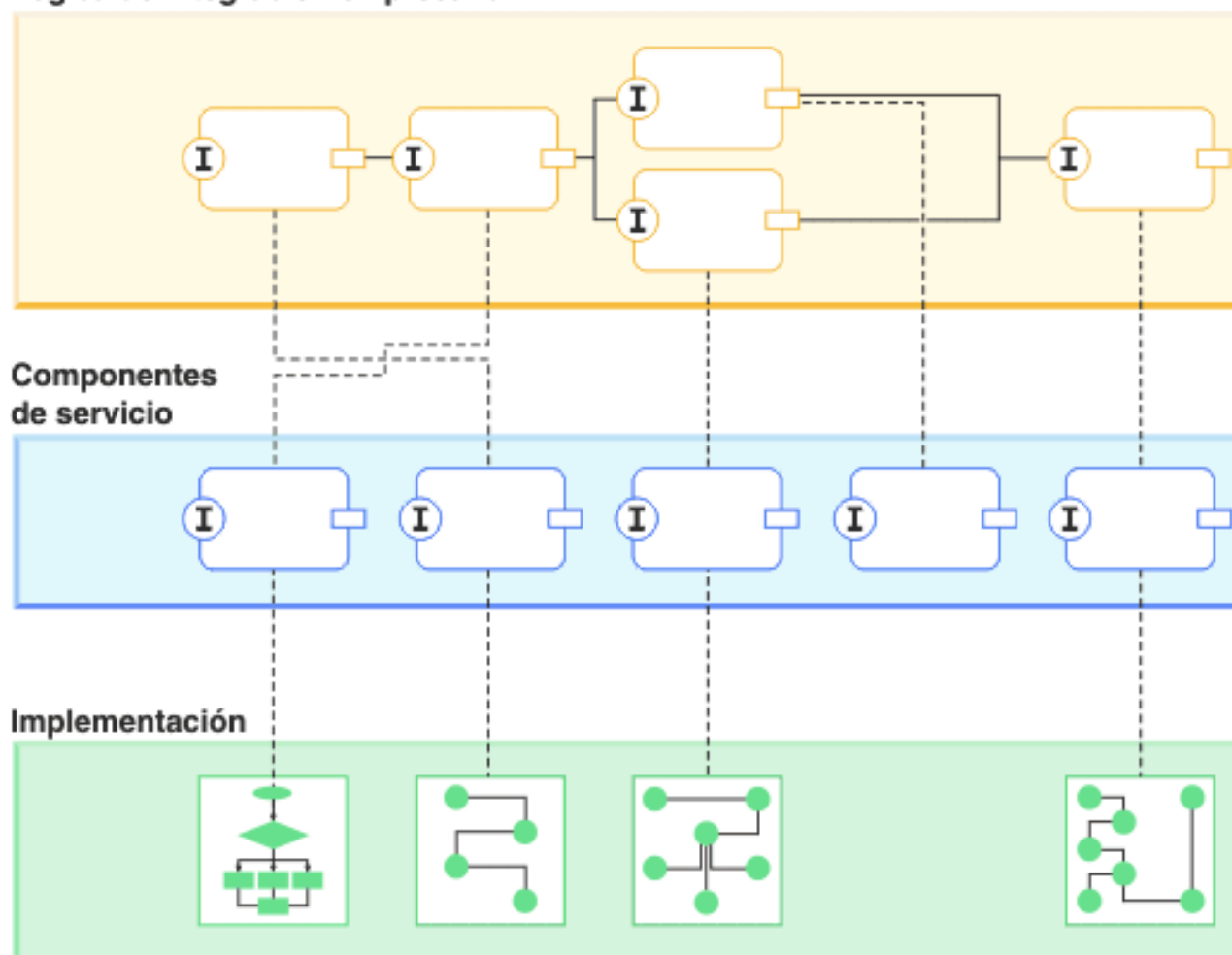
La Arquitectura de componentes de servicio, basada en la arquitectura orientada a servicios, estándar de la industria, presenta todos los procesos de negocio: servicios Web, activos de servicio de EIS, flujos de trabajo, bases de datos, etc. de forma orientada al servicio. En esta sección, se examinan a alto nivel los servicios y los objetos de datos de servicio creados por esta arquitectura, que conjuntamente expresan la lógica de negocio y hacen referencia a los datos de negocio.

El objetivo de la Arquitectura de componentes de servicio consiste en separar la lógica de integración empresarial 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 empresarial, 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 de bajo nivel. 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 el producto, 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

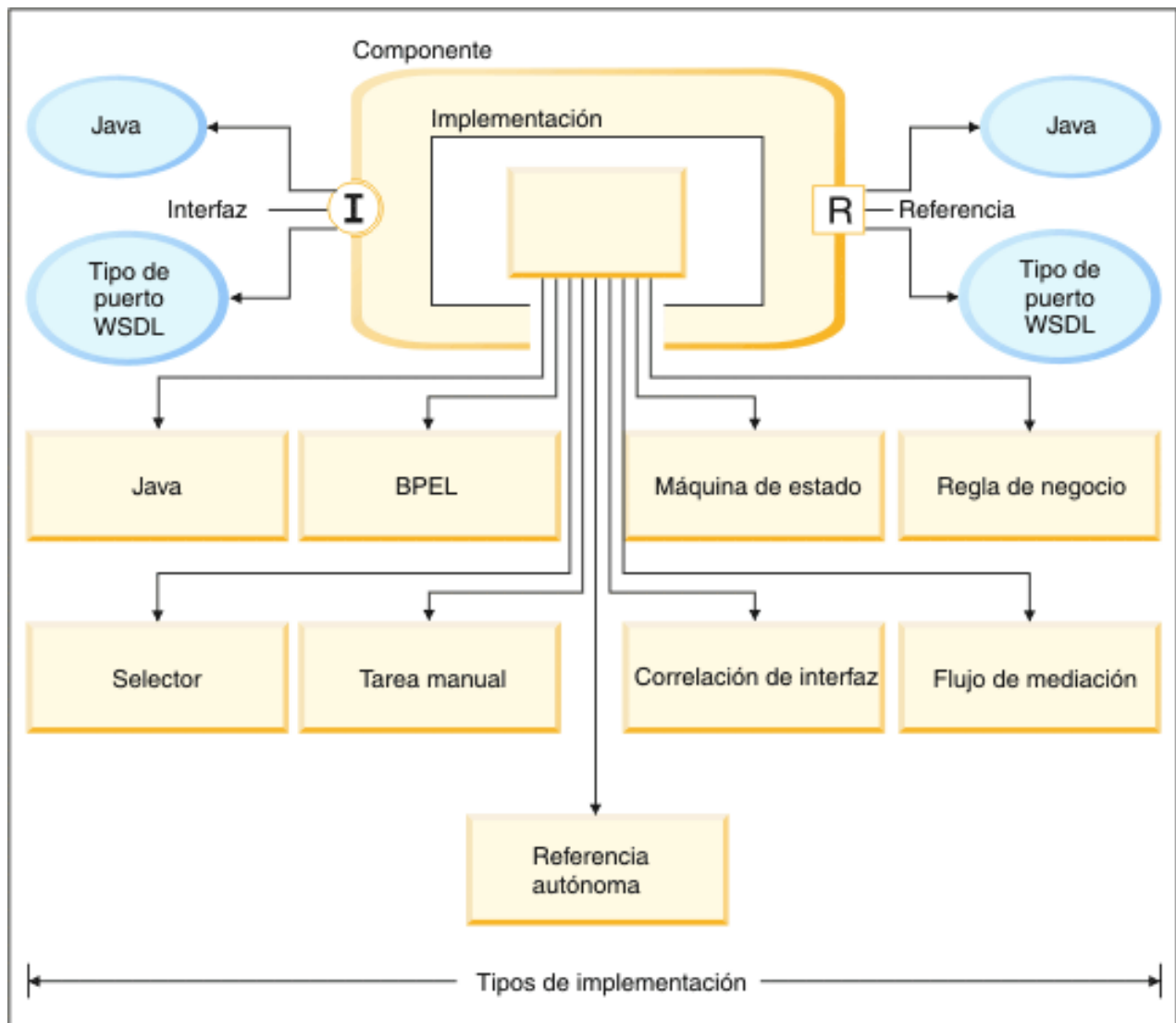


Componentes de servicio

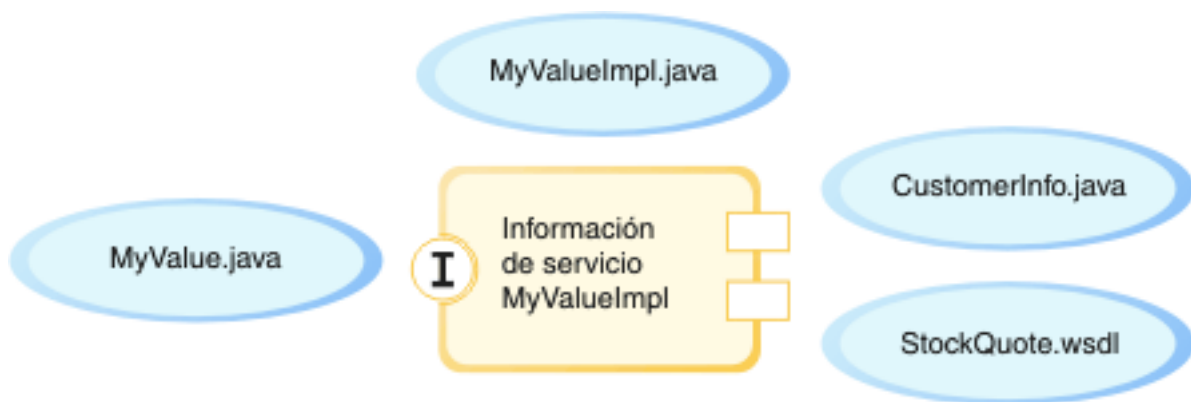
Un componente de servicio configura una implementación de servicio. Un componente de servicio se presenta en un diagrama de bloque estándar.

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 14). 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.



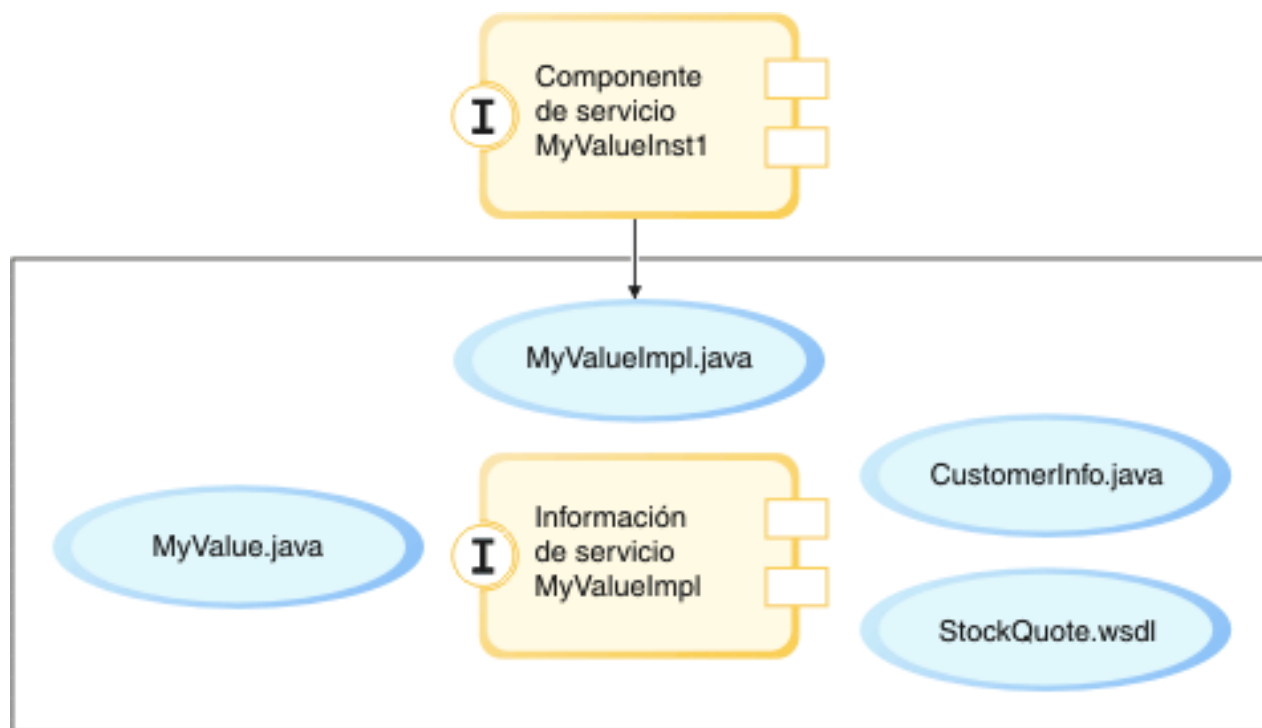
En la imagen siguiente, tenemos un componente en el centro. La implementación correspondiente MyValueImpl, está en Java como su interfaz. Tiene dos referencias: otra interfaz Java y una interfaz WSDL.



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.



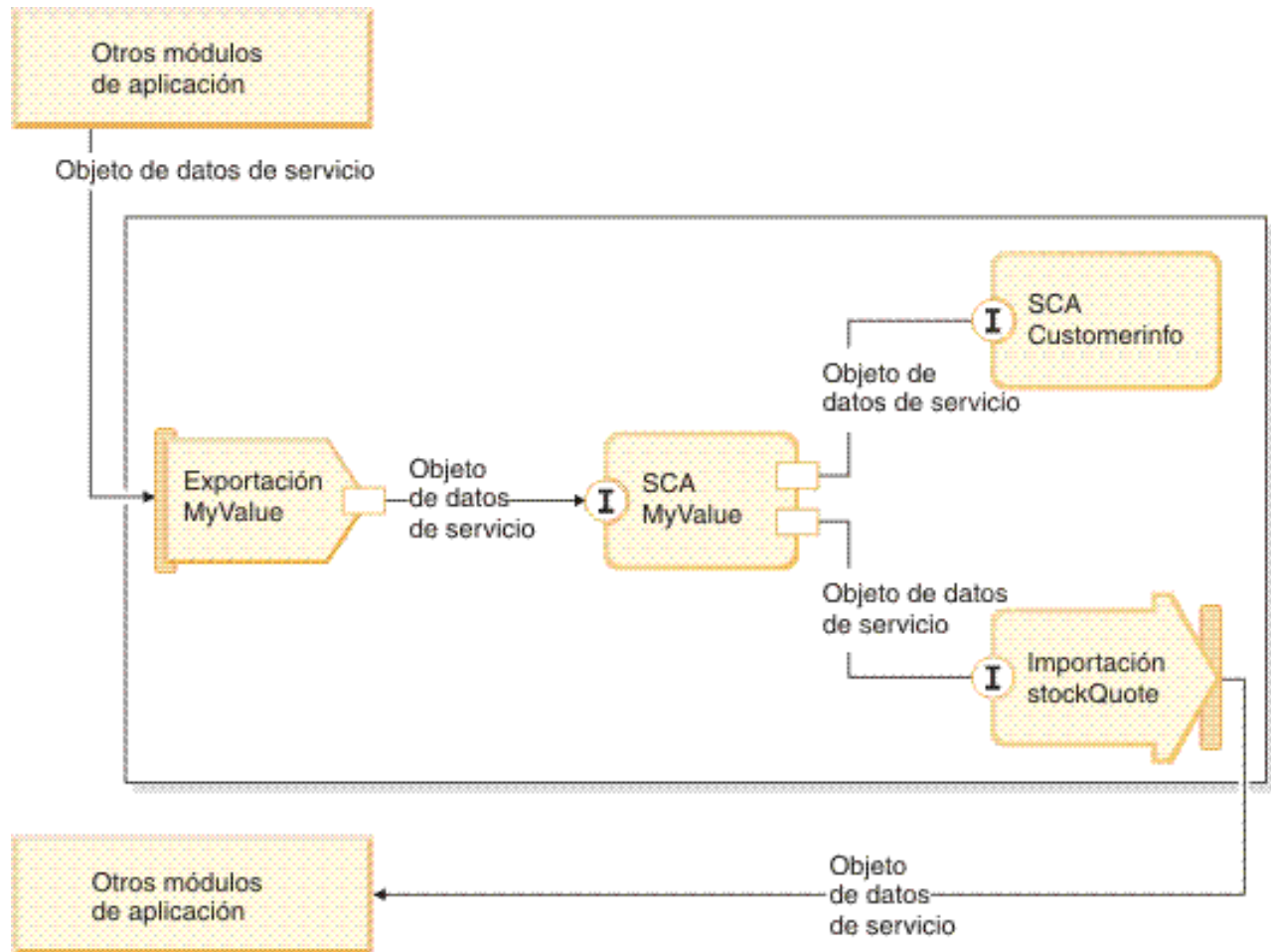
Objetos de datos de servicio

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.



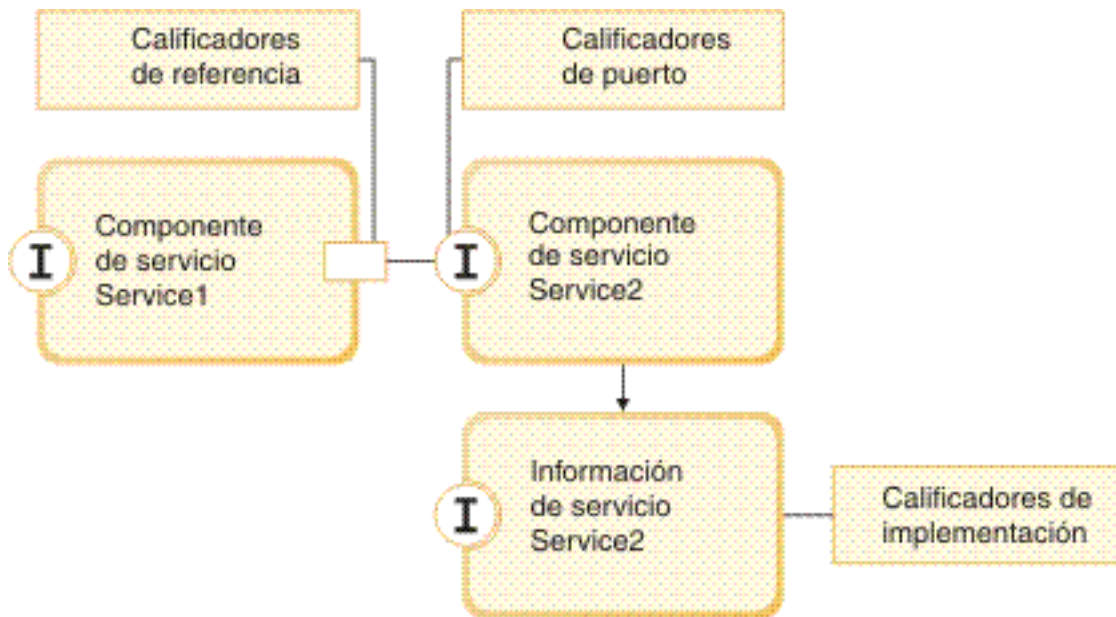
Calificadores de servicio

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.

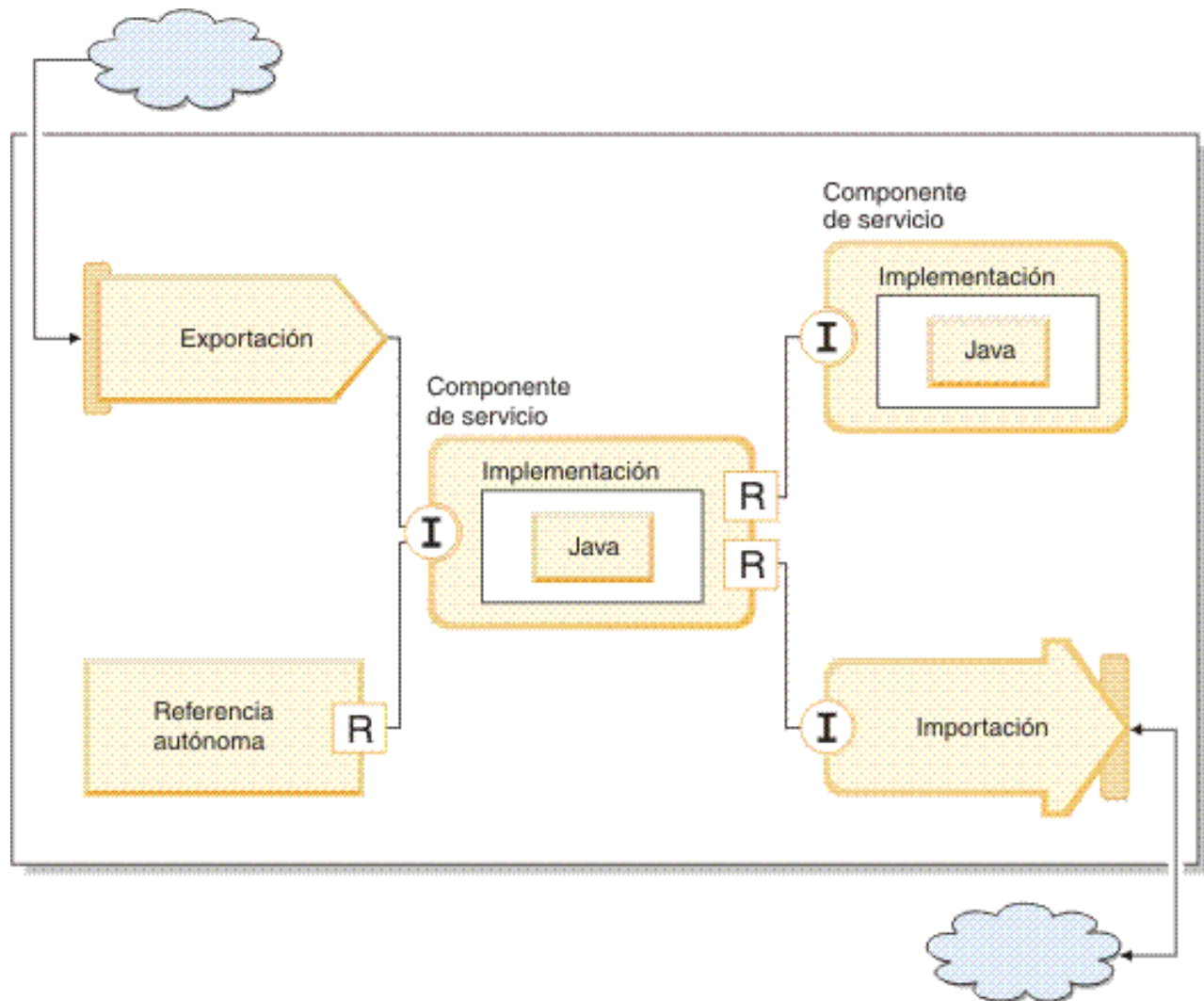
Un módulo es un compuesto de componentes de servicio, importaciones y exportaciones. Los componentes de servicio, las importaciones y las exportaciones residen en los mismos proyecto y carpeta raíz que también contienen la conexión que enlaza los componentes y los enlaces necesarios para las importaciones 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 empresarial) 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, un componente de flujo de mediación y varios componentes Java o ninguno que aumentan el componente de flujo de mediación.

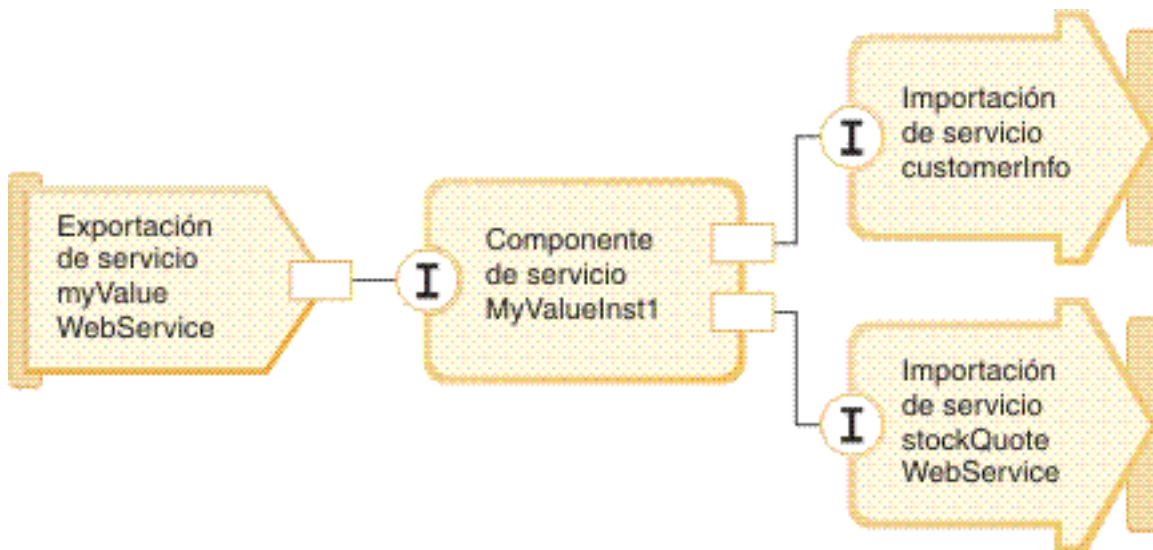
¿Por qué hay dos tipos de módulo? El primer tipo de módulo está principalmente diseñado para los procesos de negocio. Un módulo de mediación es como una pasarela a servicios externos existentes que es común en las arquitecturas de Enterprise Service Bus. A estos servicios externos o exportaciones en 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.

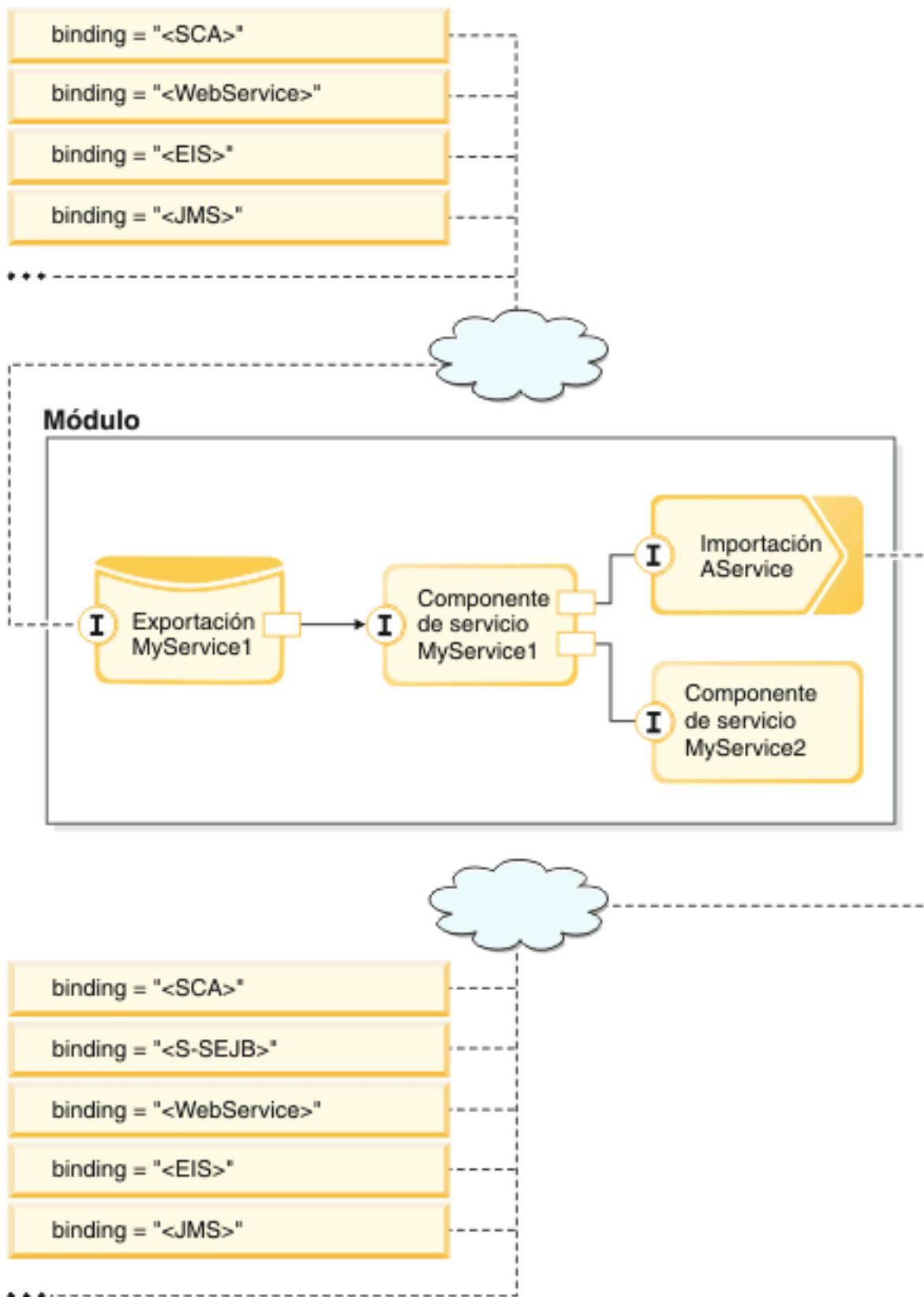
Tipos de enlace de importación y exportación de servicios

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 permite proporcionar un servicio a clientes externos como un servicio Web. El asistente de descubrimiento de servicio de empresa crea, importa y exporta representando un servicio en un sistema EIS. Los enlaces creados son de tipo EIS o JMS (Java Message Service.) Un enlace EIS proporciona comunicación síncrona con el servicio en el sistema EIS. Un enlace JMS se utilizará normalmente en interacciones con grandes sistemas EIS en los que 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 modules; es decir, no hay servicios externos
- El rendimiento es importante
- Los módulos están estrechamente cohesionados.

Elija un enlace *Servicios 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
- El protocolo de un servicio externo al que está accediendo o un servicio que desea proporcionar es SOAP/HTTP o JMS/HTTP.

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 poco cohesionados
- 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 poco cohesionado
- 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 "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

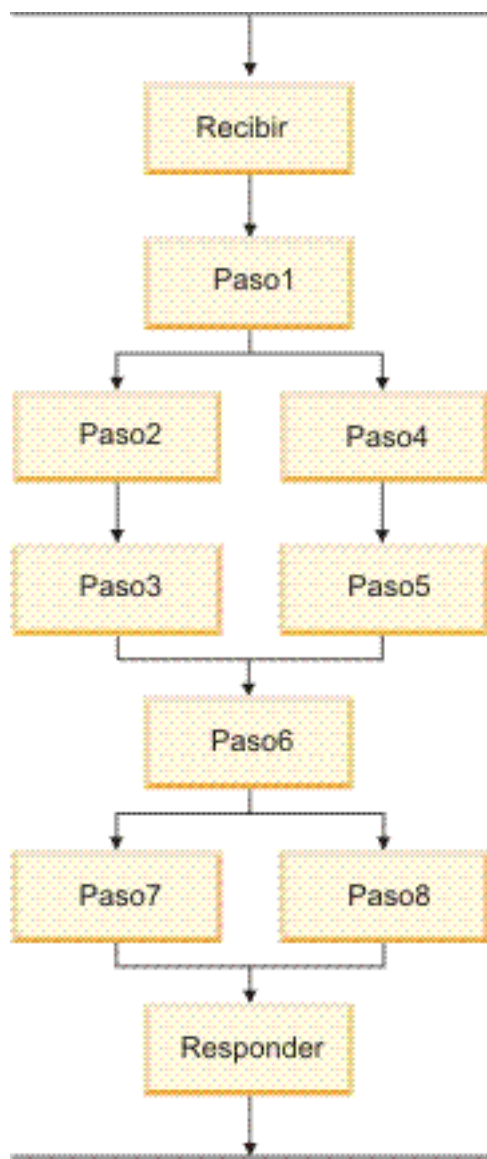
Un componente de *Proceso BPEL* implementa un proceso de negocio.

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.

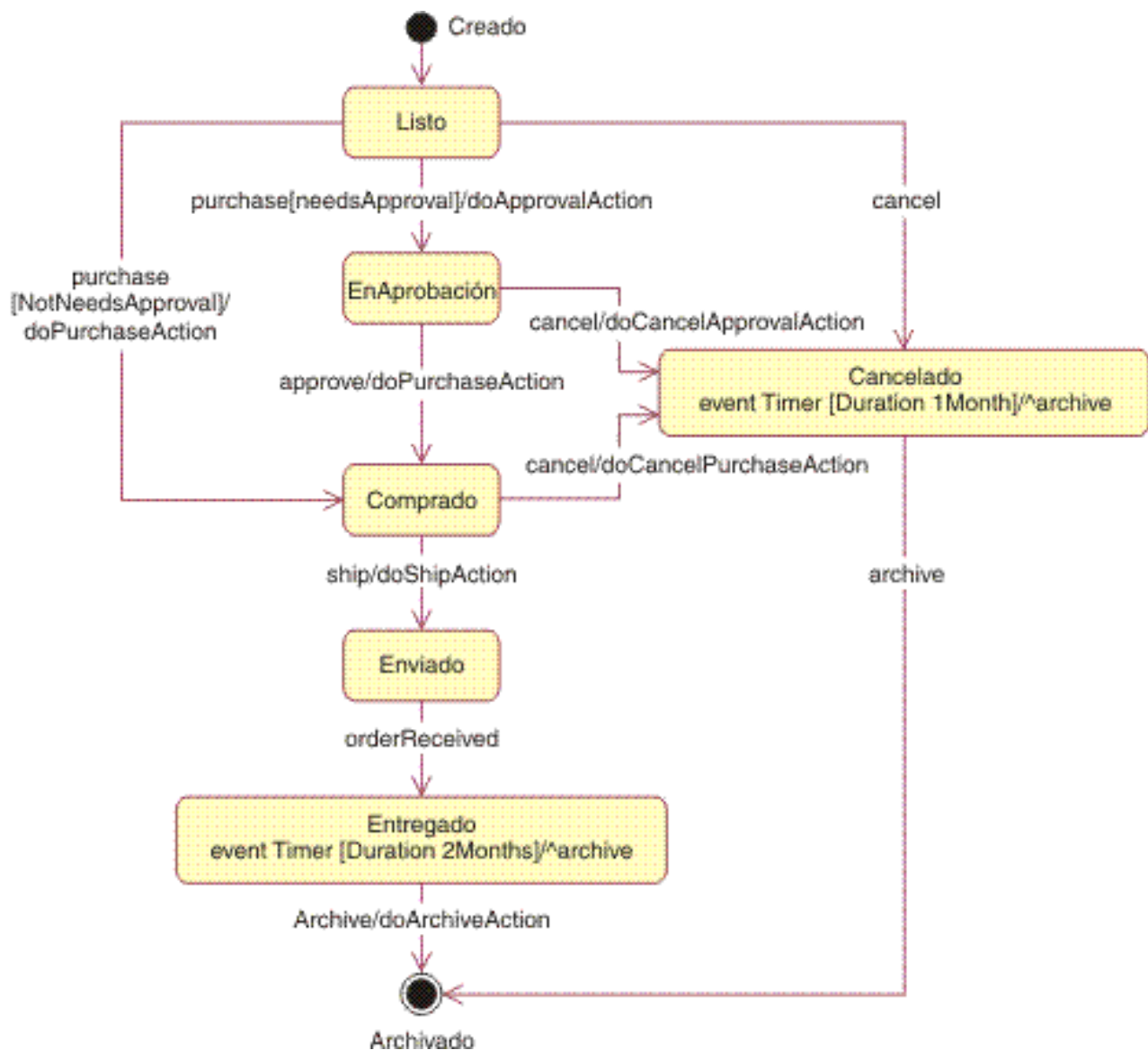


Un proceso BPEL no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

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 expendedor 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.

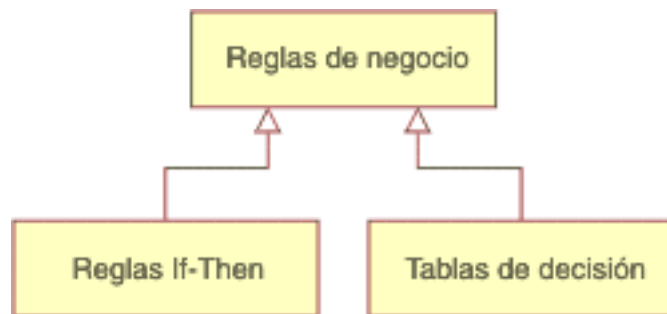
Reglas de negocio

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 un descuento de ventas postvacacional del 20%.* Sin embargo, si las ventas siguen siendo bajas, podría modificar el descuento en cualquier momento por un 40%.

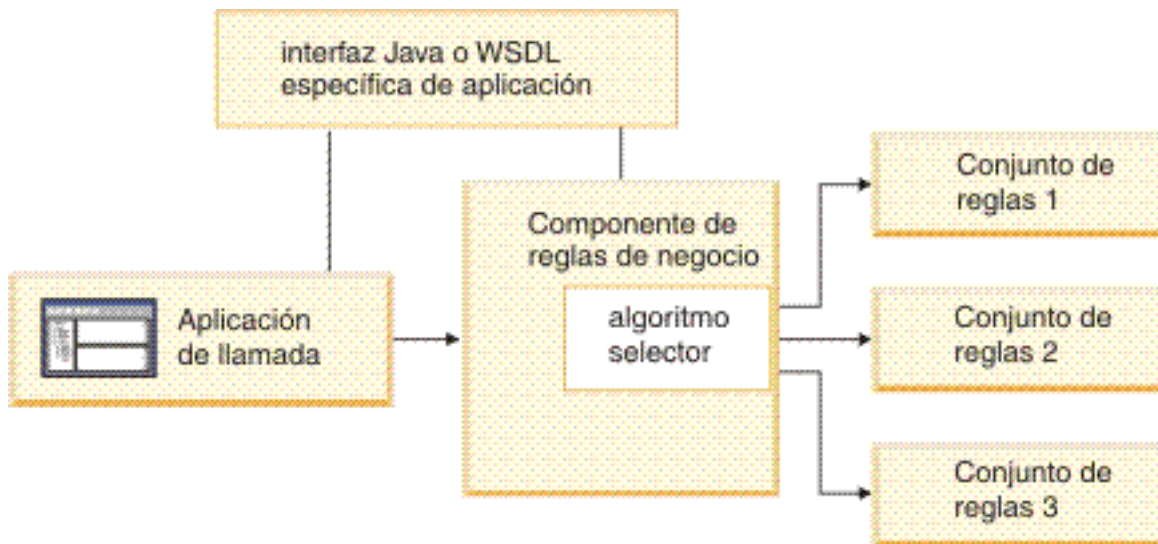


Las reglas de negocio no pueden utilizarse en un módulo de mediación. Solo pueden desplegarse en un servidor WebSphere Process Server.

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.

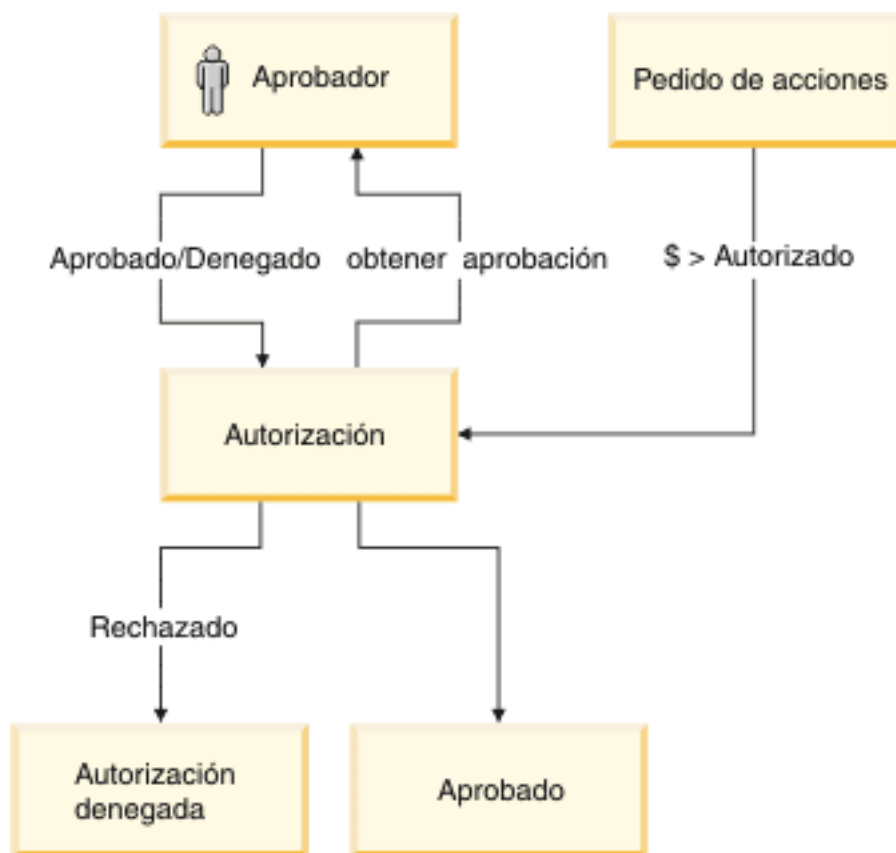


Un selector no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

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 las 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.

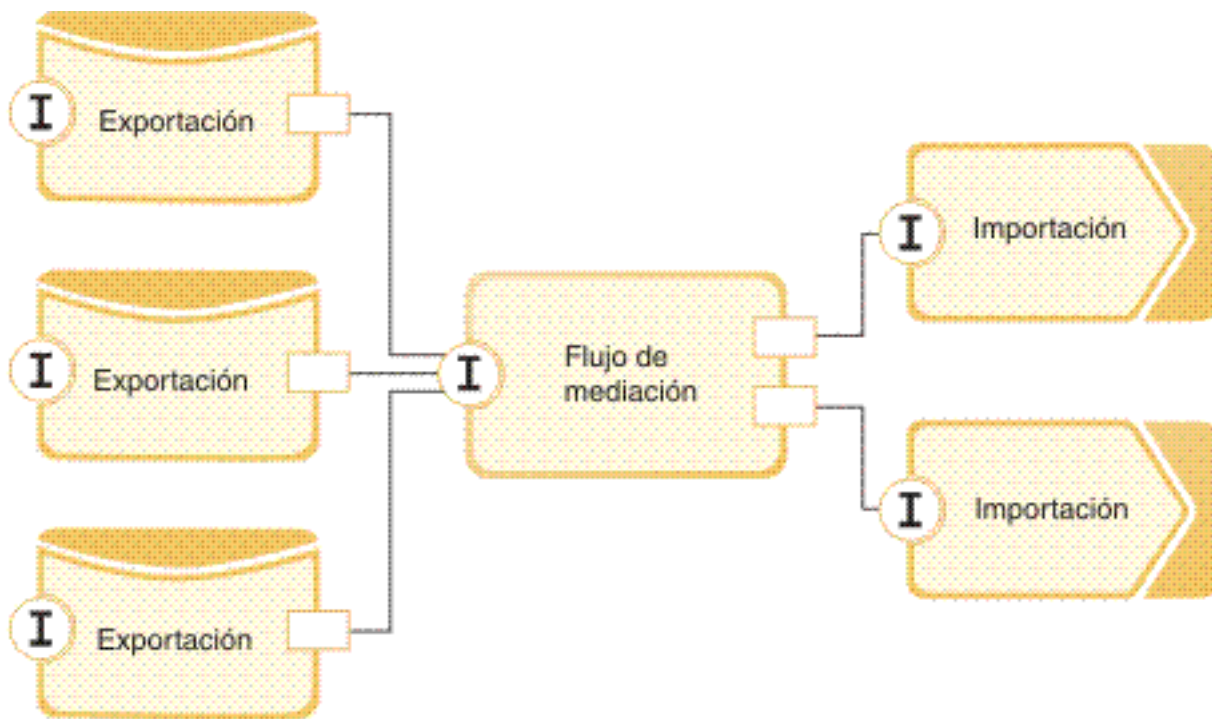
Una correlación de interfaces no puede utilizarse en un módulo de mediación. Solo puede desplegarse en un servidor WebSphere Process Server.

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, hay un módulo de mediación para un componente de flujo de mediación. Puede contener hasta un componente de flujo de mediación más cero o varios componentes Java que aumentan el componente de flujo de mediación. 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.

Información relacionada

En esta sección se tratan algunos temas relacionados con la arquitectura.

Estos temas proporcionan información adicional asociada con la arquitectura del producto.

- “Desarrollar los servicios a utilizar con los servicios .NET”
- “Soporte bidireccional” en la página 26

Desarrollar los servicios a utilizar con los servicios .NET

Si pretende desarrollar servicios que vaya a utilizar con los servicios .NET, deberá hacer algunas consideraciones especiales. Hay dos aspectos a considerar: incluir los archivos WSDL desarrollados en un entorno de desarrollo .NET en el entorno de desarrollo de WebSphere Integration Developer y exportar los archivos WSDL de WebSphere Integration Developer para que pueda utilizarlos con los servicios .NET. La diferencia clave entre los dos entornos de desarrollo es que el entorno .NET utiliza esquemas incorporados y WebSphere Integration Developer no. Los esquemas incorporados son una manera de incluir el esquema en un archivo WSDL en lugar de especificar que se importe como un archivo aparte. En WebSphere Integration Developer se proporciona información para ayudarle en ambos casos.

Esquemas incorporados, en la sección Crear interfaces, muestra como importar archivos WSDL que contienen esquemas incorporados como tienen los archivos WSDL de .NET.

Crear proxys para trabajar con servicios .NET, en la sección Ensamblar el servicio de negocio, muestra cómo publicar los servicios externamente para trabajar con servicios .NET. Este tema en concreto se centra en la creación de proxys que funcionarán con servicios .NET.

Soporte bidireccional

WebSphere Integration Developer funciona en un entorno multilingüe. Esto significa que puede visualizar y manejar datos representados en distintos idiomas. Algunos idiomas con scripts bidireccionales (por ejemplo árabe o hebreo) están incluidos en este soporte. 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, algunos puntos técnicos específicos de la utilización del soporte y las limitaciones.

Capítulo 3. Aprender acerca de las herramientas

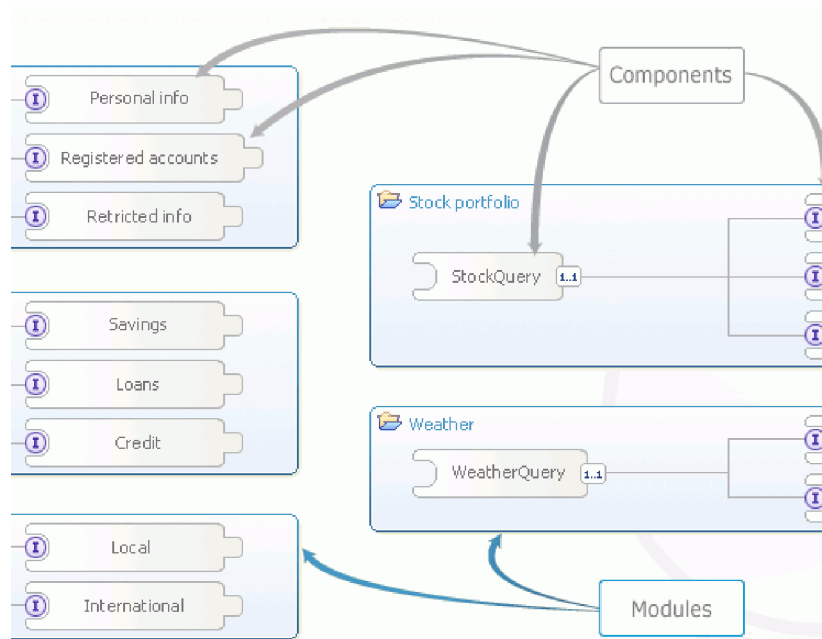
WebSphere Integration Developer utiliza herramientas visuales para crear aplicaciones integradas basándose en la arquitectura orientada a servicios.

La sección de visión general de la vista Bienvenida proporciona la explicación de nivel más alto acerca de las herramientas de WebSphere Integration Developer. En este caso, el nivel más alto significa la presentación más abstracta de las herramientas. Los gráficos explican que hacen las herramientas. Lea la sección de visión general si no conoce el producto y desea saber rápidamente qué es lo que hacen las herramientas. La sección de guías de aprendizaje que también puede encontrar en la vista Bienvenida, permite ver el funcionamiento de las herramientas. La sección de ejemplos permite empezar a utilizar las herramientas pero en modalidad a prueba de fallos. Finalmente, cuando empiece a desarrollar aplicaciones con WebSphere Integration Developer, utilice el centro de información para buscar información en profundidad sobre cada herramienta, incluyendo información de conceptos, de tareas que puede realizar y de consulta para cada herramienta.

Visión general de la vista Bienvenida

La sección de visión general de la vista Bienvenida en la vista a más alto nivel sobre las herramientas y características del producto. Dentro de la visión general, los gráficos se utilizan para transmitir lo que las herramientas clave hacen en el producto.

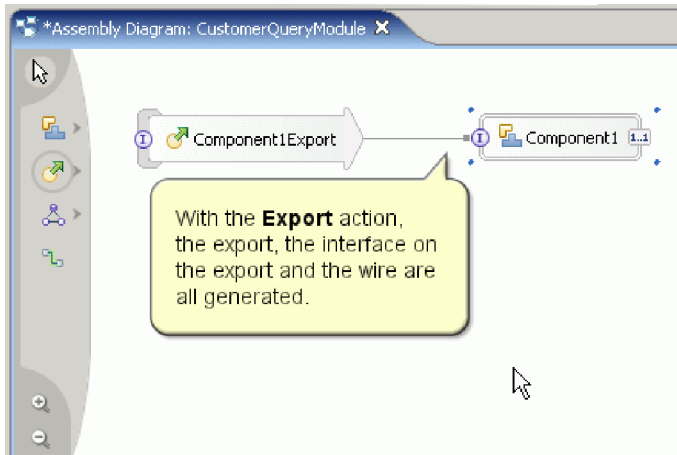
La sección de visión general de la vista Bienvenida es una forma de obtener información rápidamente acerca del producto antes de empezar a trabajar con él. La mayoría de usuarios que no conozcan el producto deberían empezar aquí. Al seleccionar un icono en la visión general se lanza un gráfico sobre una herramienta o un tema. Por ejemplo, la relación de componentes y módulos se muestra en el diagrama siguiente. También puede lanzar un visita guiada audiovisual del producto.



Guías de aprendizaje de la vista Bienvenida

Las guías de aprendizaje permiten ver las herramientas en acción en modalidad a prueba de fallos. Puede seleccionar una guía de aprendizaje y observar como realiza algunas tareas como lo haría un usuario.

Una vez conozca el producto a nivel conceptual, puede jugar con las herramientas lanzando una guía de aprendizaje. Cada guía de aprendizaje es una especie de película acerca de una herramienta determinada. Además de observar la guía de aprendizaje, puede leer una descripción de lo que está viendo en el texto encerrado en bocadillos. Las guías de aprendizaje de WebSphere Integration Developer se encuentran en la sección Aprender mirando de la galería de la vista Bienvenida.



Ejemplos de la vista Bienvenida

Los ejemplos proporcionan una exposición práctica de la utilización de WebSphere Integration Developer para desarrollar soluciones de servicio de negocio en un entorno a prueba de fallos.

Los ejemplos se encuentran en la Galería de ejemplos de la vista Bienvenida. Al seleccionar un ejemplo se dan varias opciones. Puede construirlo usted mismo siguiendo las instrucciones paso a paso o puede obtenerlo prefabricado.

El ámbito de los ejemplos es variado. Un ejemplo de tecnología se centra en una herramienta que realiza una tarea específica. Un ejemplo de aplicación utiliza varias herramientas para conseguir un resultado más complejo. Un caso práctico es un ejemplo extenso que construye una gran aplicación con muchas herramientas. Cada ejemplo indica el tiempo que se tarda en construirlo.

Business state machine (simple)

This sample demonstrates how a business state machine can be used to moderate a sales order transaction. Specifically, the state machine emulates an on-line brokerage that manages the selling of a share.

The following tools are used in this application:

- Business state machine editor
- Human task editor

To import the ready-made sample, click the **Import** link below and click **Finish** in the opened wizard. See **Running instructions** to run the imported code.

or

If you want to build the sample for yourself, click **Step-by-step instructions**.

Ready-made sample

[Import](#) [Running instructions](#)

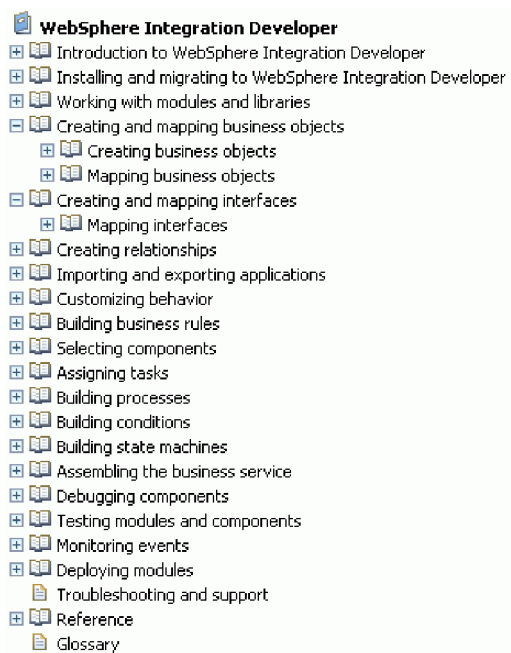
Build it yourself

[Step-by-step instructions](#)

Centro de información

El centro de información proporciona toda la información del producto. Además de guías de aprendizaje y ejemplos, aquí encontrará información sobre conceptos, tareas e información de consulta para cada herramienta de WebSphere Integration Developer.

Navegando por el centro de información puede acceder rápidamente a la información deseada. También puede realizar búsquedas en todo el centro de información. Seleccione entre los resultados de la búsqueda para ver varios temas que contengan información en profundidad sobre el tema, así como enlaces a información relacionada.



Avisos

La documentación XDoclet incluida en este producto de IBM se utiliza con permiso y está cubierta por la declaración de atribución de copyright siguiente: Copyright (c) 2000-2004, XDoclet Team. Reservados todos los derechos.

Partes basadas en *Design Patterns: Elements of Reusable Object-Oriented Software*, por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, Copyright (c) 1995 por Addison-Wesley Publishing Company, Inc. Reservados todos los derechos.

Derechos restringidos de los usuarios del gobierno de los EE.UU.: utilización, duplicación o revelación restringidas por el contrato GSA ADP Schedule Contract con IBM Corp.

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 esta documentación 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 servicios 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 esta documentación. 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 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, 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 O CONDICIONES IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O ADECUACIÓN A UN PROPÓSITO DETERMINADO. Algunas legislaciones no contemplan la declaración de limitación de responsabilidad, ni implícita ni explícita, 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 hecha en esta información a sitios Web que no sean de IBM se proporciona únicamente para su comodidad y no debe considerarse en modo alguno como promoción 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 éste) y (ii) utilizar mutuamente la información que se ha intercambiado, deben ponerse en contacto con:

*Intellectual Property Dept. for Rational Software
IBM Corporation
20 Maguire Road
Lexington, Massachusetts 02421-3112
EE.UU.*

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 esta documentación 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.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlas de la forma más completa posible, los ejemplos pueden incluir 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 ilustra 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. Usted puede copiar, modificar y distribuir los programas de ejemplo de cualquier forma,

sin tener que pagar a IBM, con el fin de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con las interfaces de programación de aplicaciones (API) de IBM.

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:

(C) (el nombre de su empresa) (el año). Algunas partes de este código se derivan de programas de ejemplo de IBM Corp. (C) Copyright IBM Corp. 2000, 2006. 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 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

Consulte el sitio <http://www.ibm.com/legal/copytrade.shtml>.



Impreso en España

SC11-3120-03

