

Versão 6.0.1



Visão Geral Técnica do Produto

Versão 6.0.1



Visão Geral Técnica do Produto

Comunicado

Antes de utilizar estas informações e o produto suportado por elas, leia as informações gerais em Avisos no final deste manual.

Índice

Capítulo 1. Integração de Negócios 1

Integração entre Unidades de Negócios	2
Integração entre Empresas	2
WebSphere Integration Developer	3
Padrões	4
Função do Desenvolvedor de Integração	5

Capítulo 2. Service Component

Architecture 7

Componentes de Serviço	8
Objetos de Dados de Serviço	10
Qualificadores de Serviço	11
Módulos	12
Importações e Exportações	14
Tipos de Ligação da Importação e Exportação de	
Serviços	14
Selecionando Ligações Apropriadas	15
Tipos de Implementação de Serviço	16
Objetos Java	16

Processo BPEL	17
Máquinas de Estado	18
Regras de Negócios.	19
Seletores	20
Tarefa Humana	21
Mapa de Interface	22
Fluxo de Mediação	23
Referências independentes	24
Informações Relacionadas	24

Capítulo 3. Aprendendo sobre as Ferramentas 27

Visão Geral da Visualização Bem-vindo	27
Tutoriais na Visualização Bem-vindo	28
Amostras na Visualização Bem-vindo.	28
Centro de Informações.	29

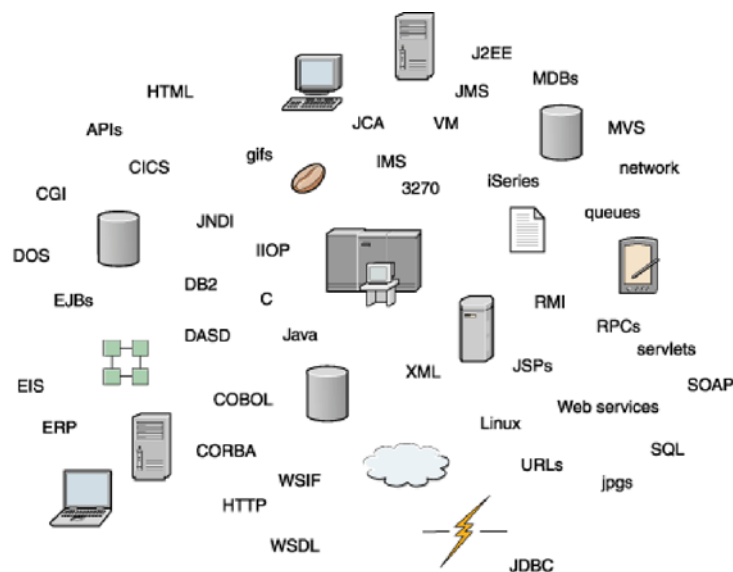
Avisos 31

Capítulo 1. Integração de Negócios

Integração de negócios significa integrar aplicativos, dados e processos em uma empresa ou em um conjunto de empresas. O desafio dessa tarefa e o modo como ela é vista pelo WebSphere Integration Developer são examinados.

Você foi instruído a construir um portal para que seus clientes acessem seus aplicativos de negócios. Ele deve fornecer acesso a dezenas de aplicativos de negócios importantes e dados relacionados espalhados por todas as suas unidades de negócios. Você também foi instruído a incluir os aplicativos de seus parceiros de negócios no portal. Integração também significa desenvolver processos, pois haverá alguma lógica na sequência destes aplicativos montados. Você possui 20 unidades de negócios e uma dúzia de parceiros de negócios. Seu portal deve estar disponível na Web 24 horas por dia. Você tem uma equipe de seis desenvolvedores, incluindo você, e você tem quatro meses para o portal estar em funcionamento e em execução.

A maioria das pessoas com esse desafio olharia para a tecnologia acumulada nas últimas décadas e veria uma colagem como a figura a seguir:



Trata-se de algo assustador, mas não de uma situação impossível. O problema mais difícil que você vai enfrentar além da coleta desorganizada de hardware e software é a sua limitação de tempo e recursos. Você deverá encontrar ferramentas poderosas para reunir rapidamente os aplicativos e dados dispersos nas empresas envolvidas. A codificação manual não é uma opção.

Você é o único que está enfrentando esta situação? Esse é um problema comum que está há muitos anos em formação. De acordo com a pesquisa *CIO Survey*, de dezembro de 2001, a integração de aplicativos é constantemente uma das três maiores prioridades em tecnologia. De acordo com o *The Business Integrator Journal*, inverno de 2001, dois de três desenvolvedores, em uma pesquisa recente, utilizam software de integração para desenvolver soluções baseadas na Web. Em média, cada desenvolvedor estava integrando três sistemas diferentes.

Vamos olhar para duas forças nos negócios que estão levando o problema para uma crise: integração entre unidades de negócios dentro de uma empresa e integração entre empresas. Em seguida, olharemos para o modo como o WebSphere Integration Developer aborda esses problemas e, em particular, o

significado de seu compromisso com os padrões de mercado. Finalmente, conheceremos o especialista em integração, a pessoa que utilizará as ferramentas do WebSphere Integration Developer para solucionar os problemas apresentados anteriormente.

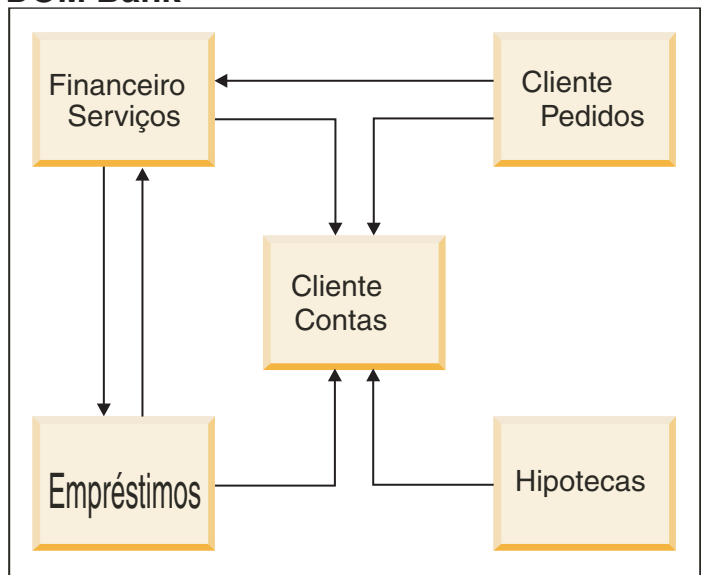
Integração entre Unidades de Negócios

As unidades de negócios freqüentemente colaboram entre si nos dias de hoje, criando uma necessidade de forte integração de seu aplicativos.

Unidades de negócios antes autônomas estão sendo integradas porque a tecnologia permite que elas sejam conectadas e porque a eficiência diz que elas devem estar operando de maneira mais cooperativa para minimizar as despesas e maximizar a saída. Um objetivo corporativo em comum também reúne as unidades de negócios. Uma unidade de marketing e uma unidade de pesquisa-e-desenvolvimento desejam produzir um produto lucrativo. Integrando o conhecimento do mercado com as informações de desenvolvimento do produto, a possibilidade de produzir esse produto bem-sucedido aumenta. A cooperação entre unidades de negócios também permite que as corporações alavanquem seus muitos aplicativos de negócios existentes permitindo sua reutilização em diferentes contextos de negócios.

A integração entre unidades de negócios é mais fácil do que entre empresas porque existem menos riscos de segurança e o gerenciamento das interações entre as unidades não deve ser tão difícil. As unidades de negócios provavelmente estão utilizando os mesmos protocolos, sistemas operacionais e linguagens de computador. Em outras palavras, é um ambiente relativamente homogêneo. No entanto, a chave é ter as ferramentas certas para integrar rapidamente os aplicativos. No diagrama a seguir, o banco DOM possui várias unidades de negócios que precisam compartilhar suas informações umas com as outras. Vários anos atrás, o banco DOM poderia gerenciar imprimindo cópias de informações de uma unidade de negócios e enviando-as para outra unidade, que tinha seus próprios sistemas e aplicativos. Hoje, o banco DOM deve criar aplicativos integrados incluindo suas unidades de negócios se não quiser ficar atrás de seus concorrentes.

DOM Bank



Integração entre Empresas

As forças que guiam a integração de aplicativos entre unidades de negócios também se aplicam entre corporações, visto que as parcerias ou aquisições requerem dados e processos compartilhados.

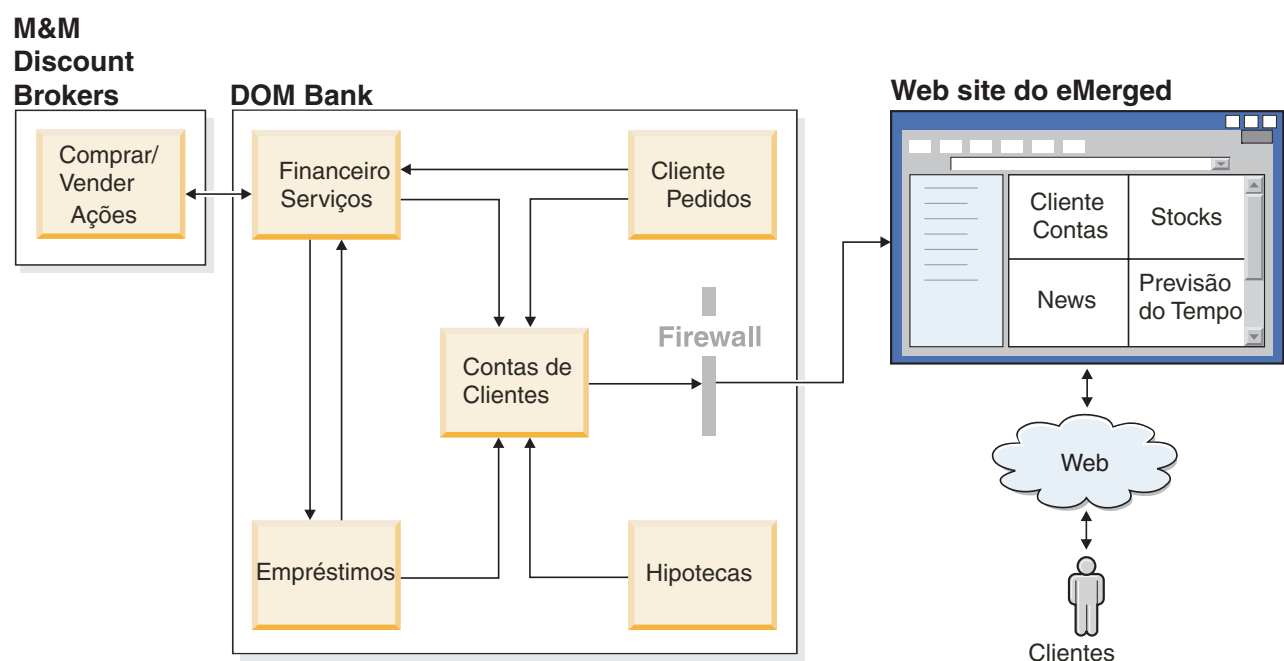
A tecnologia permite que empresas estejam vinculadas mutuamente em áreas benéficas. Por exemplo, um fabricante de automóveis pode configurar um processo integrado com um fornecedor de pneus para que quando o estoque de pneus estiver baixo o fornecedor seja automaticamente notificado. A integração entre empresas está sendo direcionada pela necessidade econômica. Ter relações mais estreitas entre corporações significa menos atraso e menos despesas para efetivar os negócios. Esses processos automatizados significam que as pessoas gastam menos tempo para processar transações entre gastos corporativos e de percurso e as reuniões de comunicação direta podem ser reduzidas de maneira significativa. Da mesma forma, os custos administrativos são reduzidos e o tempo gasto entre notificação, entrega e faturamento é reduzido.

Entretanto, empresas diferentes têm histórias diferentes. Seus aplicativos são codificados em linguagens diferentes em plataformas diferentes utilizando protocolos de comunicação diferentes. Também existem maiores riscos para a segurança ao trabalhar com organizações diferentes. Quaisquer que sejam os benefícios e mesmo a necessidade de integração entre empresas, os custos com o tempo de desenvolvimento podem ser significativos sem as ferramentas certas.

WebSphere Integration Developer

O WebSphere Integration Developer é a resposta para os desafios de integração que as organizações enfrentam diariamente. Ele foi projetado como um ambiente de desenvolvimento de integração completo para aqueles que constroem aplicativos integrados. Para simplificar e acelerar o desenvolvimento dos aplicativos integrados, esse ambiente fornece uma camada de abstração que separa os componentes apresentados visualmente com os quais você trabalha da implementação básica.

Os aplicativos integrados não são simples. Eles podem chamar aplicativos no EIS (Enterprise Information Systems), envolver processos de negócios através de departamentos ou corporações e chamar aplicativos local ou remotamente gravados em uma variedade de idiomas e em execução em diversos sistemas operacionais. Por exemplo, o eMerged Corporation foi criado unindo o banco DOM e os intermediários M&M Discount. A união significou tudo que foi mencionado acima: aplicativos em sistemas EIS, processos de negócios e aplicativos dentro de cada uma das antigas corporações tinham que ser compartilhados entre as corporações e apresentados de maneira direta para o novo conjunto de clientes. No entanto, o eMerged realizou a tarefa e, como mostra o diagrama a seguir, os clientes de ambas as empresas antes separadas podem acessar todas as suas informações financeiras on-line.



O eMerged utilizou as ferramentas do WebSphere Integration Developer para construir os aplicativos integrados para eles mesmos e seus clientes. Essas ferramentas apresentam aplicativos, incluindo aplicativos que existem remotamente em sistemas EIS e processos de negócios como componentes. Os componentes são criados e montados em outros aplicativos integrados (ou seja, aplicativos criados a partir de um conjunto de componentes) através de editores visuais. Os editores visuais apresentam uma camada de abstração entre os componentes e suas implementações. Um desenvolvedor que utiliza as ferramentas pode criar um aplicativo integrado sem conhecimento detalhado da implementação básica de cada componente.

As ferramentas permitem uma abordagem de projeto de cima para baixo para construir um aplicativo integrado, onde a implementação de um ou mais componentes não existe e é incluída posteriormente; ou uma abordagem de baixo para cima, na qual os componentes já estão implementados e o desenvolvedor monta-os arrastando e soltando-os em um editor visual e, em seguida, cria um fluxo lógico entre eles unindo-os através de linhas. Um ambiente de teste e depuração significa um teste completo antes que seus aplicativos sejam implementados em um servidor de produção. A configuração de pontos de monitoramento permite que você veja como um aplicativo é utilizado em tempo real para ajustá-lo para um melhor desempenho.

As ferramentas do WebSphere Integration Developer são baseadas em uma arquitetura orientada a serviços. Componentes são serviços e um aplicativo integrado envolvendo muitos componentes é um serviço. Os serviços criados estão em conformidade com os padrões líderes de mercado. Os processos de negócios, que também se tornam componentes, são criados de maneira semelhante com ferramentas visuais de fácil utilização as quais estão de acordo com o BPEL (Business Process Execution Language) padrão de mercado. O WebSphere Integration Developer está disponível nas plataformas Windows e Linux.

Aqui estão alguns benefícios das ferramentas do WebSphere Integration Developer:

- São fáceis de aprender
- Podem ser aplicadas em situações complexas de integração
- Você pode produzir aplicativos rapidamente de acordo com o padrões de mercado

Padrões

Aplicativos criados pelo WebSphere Integration Developer conforme os padrões de mercado associados com a arquitetura orientada a serviços.

Ninguém quer criar aplicativos ligados ao código proprietário que pode não ser mais suportado em alguns anos ou envolver altas taxas de licenciamento. A integração baseada em padrões é, portanto, um aspecto fundamental do WebSphere Integration Developer. Para a conectividade, são utilizados os padrões do J2EE Connector Architecture. Para o sistema de mensagens assíncrono, muitas vezes utilizado em grandes aplicativos que exigem a entrega de dados com garantia, é utilizado o padrão JMS (Java Message Service). O WebSphere Integration Developer pode integrar facilmente serviços da Web baseados em SOAP (Simple Object Access Protocol). Para descrever um serviço, o padrão WSDL (Web Services Description Language) bem estabelecido é utilizado. Para definir um processo de negócios, o padrão BPEL (Business Process Execution Language) é utilizado.

Essas interfaces e componentes baseados em padrões incluem uma arquitetura ilimitada e que pode ser conectada. Elementos proprietários, no entanto, não são excluídos; eles são acessados através da utilização de interfaces padronizadas. Isso significa que os aplicativos criados no WebSphere Integration Developer podem interagir com aplicativos .NET, por exemplo. Na seção de arquitetura, é fornecido um link para uma Service Component Architecture completa, que fornece uma lista extensiva dos vários padrões suportados.

Função do Desenvolvedor de Integração

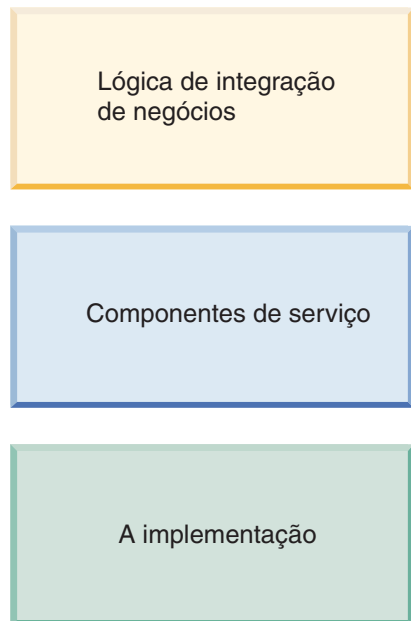
O desenvolvedor de integração é o principal usuário do WebSphere Integration Developer. Essa pessoa, através da utilização de ferramentas visuais, pode construir um aplicativo integrado complexo sem exigir amplo conhecimento da implementação básica.

O WebSphere Integration Developer apresenta aplicativos e processos de negócios como *componentes*. A implementação dos componentes permanece oculta e os componentes interoperam através de interfaces. Como resultado, os desenvolvedores de integração não precisam ter amplo conhecimento da implementação básica dos componentes para criar um aplicativo integrado que os utilize. No entanto, os desenvolvedores de integração provavelmente terão um amplo conhecimento técnico no campo da integração, visto que eles precisam de algum entendimento dos sistemas EIS, processos de negócios e aplicativos codificados em Java ou outras linguagens. Por exemplo, um arquiteto tem amplo conhecimento sobre como um sistema funciona, mas sem saber detalhadamente o que cada componente faz. Como um arquiteto, um desenvolvedor de integração pode ser a pessoa em uma organização que projeta o aplicativo total e, então, tem outros que codificam a implementação dos componentes específicos.

Capítulo 2. Service Component Architecture

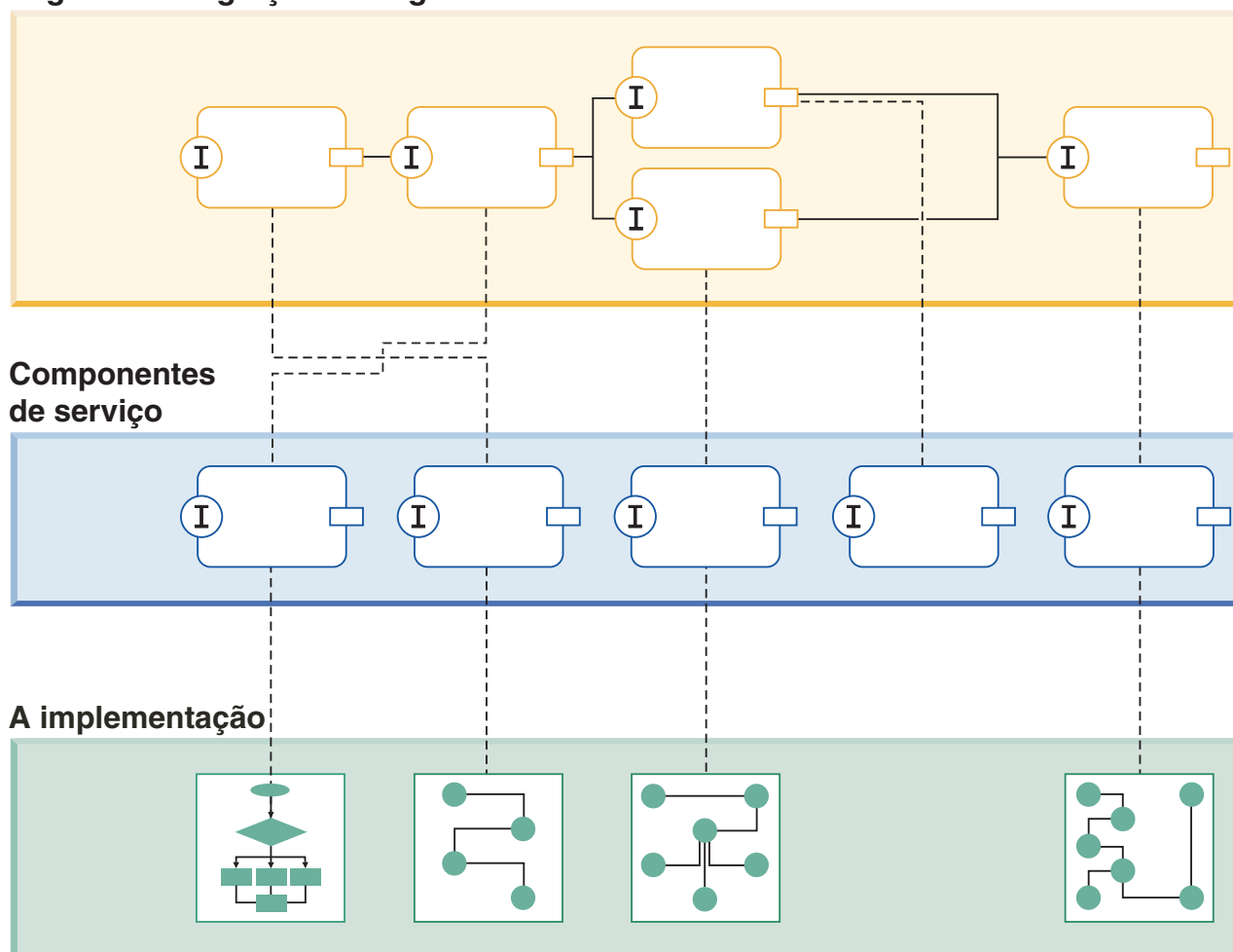
A Service Component Architecture, baseada na arquitetura padrão de mercado orientada a serviço, apresenta todos os processos de negócios - serviços da Web, recursos de serviço EIS (Enterprise Information System), fluxos de trabalho, bancos de dados, entre outros - de uma forma orientada a serviços. Nesta seção, examinaremos em um nível superior os serviços e os objetos de dados de serviço criados por esta arquitetura, que juntos expressam a lógica de negócios e se referem aos dados de negócios.

A meta da Service Component Architecture é separar a lógica de integração de negócios da implementação, de modo que um desenvolvedor de integração possa se concentrar na montagem de um aplicativo integrado em vez de nos detalhes da implementação. Para alcançar esse objetivo, os componentes de serviço que contêm a implementação de serviços individuais requeridos pelos processos de negócios são criados. O resultado é uma arquitetura de três camadas - lógica de integração de negócios, componentes de serviços e implementação - como mostrado no diagrama a seguir.



Visto que os componentes de serviço contêm a implementação, eles podem ser montados graficamente pelo desenvolvedor de integração sem precisar de conhecimento dos detalhes da implementação de nível inferior. Os componentes de serviço oferecem também a opção de permitir que o desenvolvedor de integração ou alguém que trabalhe para o desenvolvedor de integração inclua a implementação posteriormente. Conforme você verá no produto, os componentes são montados visualmente. Em outras palavras, você não é exposto ao código dentro dos componentes. No nível da lógica de negócios, mostrado no diagrama a seguir, os componentes são montados independente de sua implementação. A arquitetura orientada a serviços então permite que você se concentre em solucionar seus problemas de negócios utilizando e reutilizando componentes em vez de desviar sua atenção para a tecnologia que está implementando os serviços que você está utilizando.

Lógica de integração de negócios

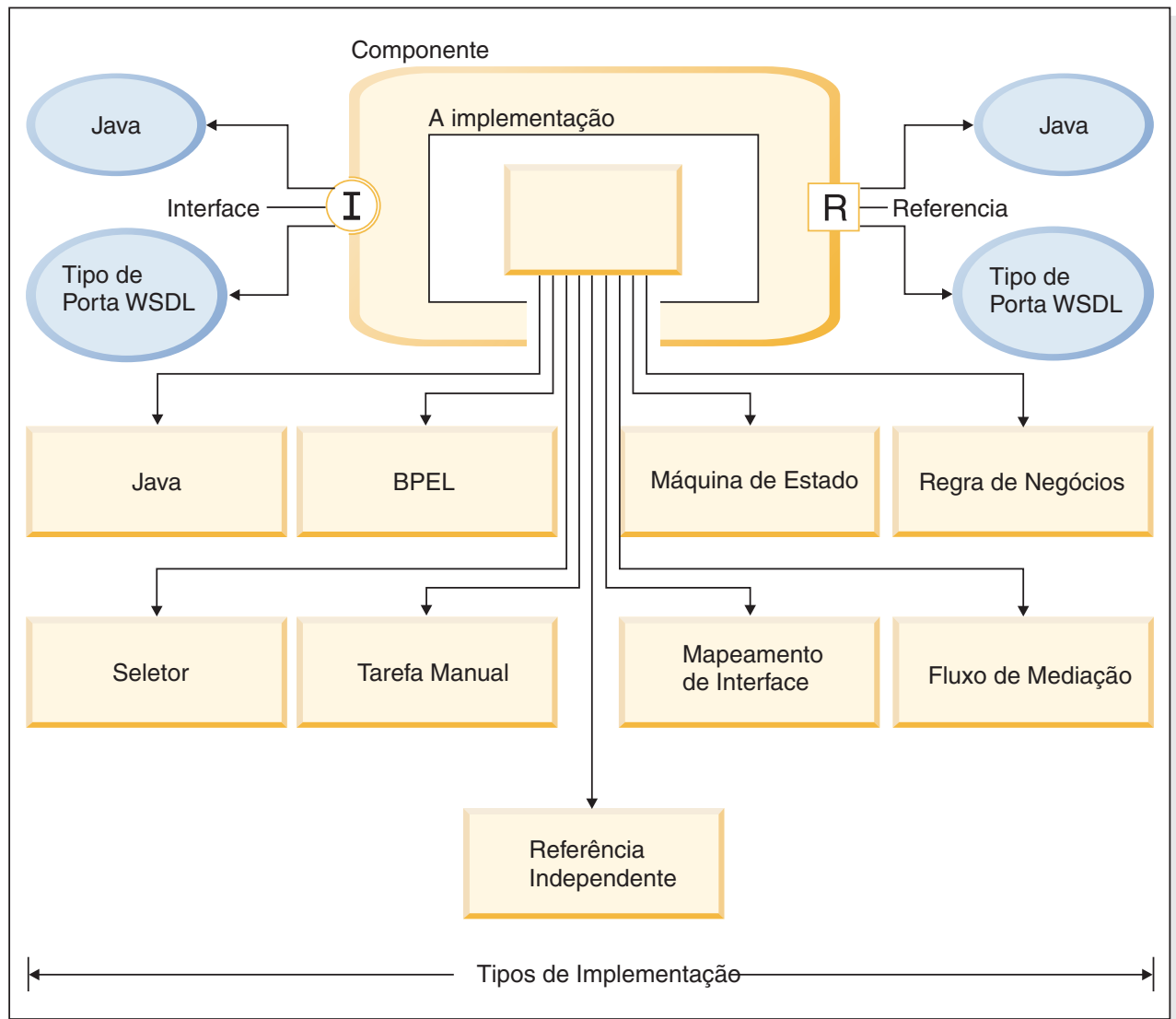


Componentes de Serviço

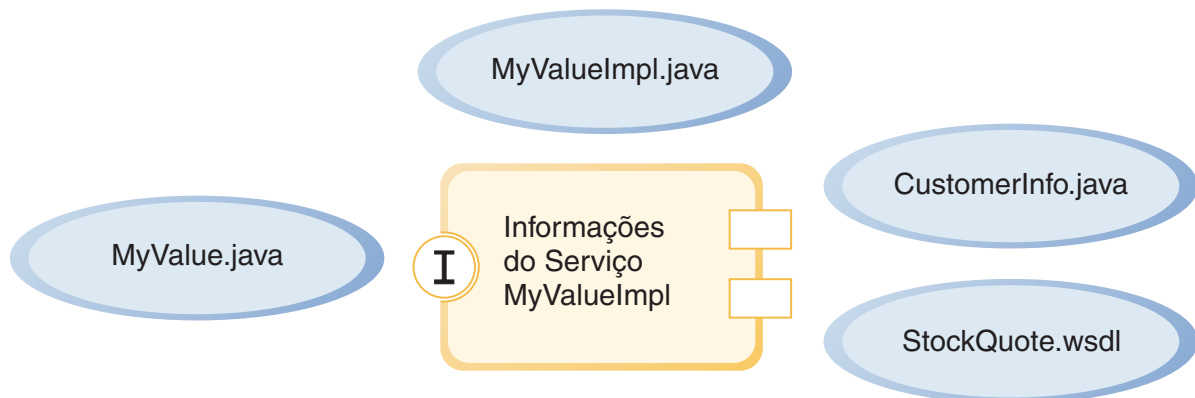
Um componente de serviço configura a implementação de um serviço. Um componente de serviço é apresentado em um diagrama de bloco padrão.

Um componente é composto de uma implementação, que fica oculta ao utilizar as ferramentas do WebSphere Integration Developer, uma ou mais interfaces, que define suas entradas, saídas e falhas, e zero ou mais referências. Uma referência identifica a interface de outro serviço ou componente que este componente requer ou consome. Uma interface pode ser definida em uma das duas linguagens: um tipo de porta WSDL ou Java. Uma interface suporta estilos de interação síncronos e assíncronos. A implementação de um componente pode ser feita em várias linguagens.

Um tipo de interface pode ser WSDL ou Java, mas se houver várias interfaces, você não pode misturar WSDL com Java.



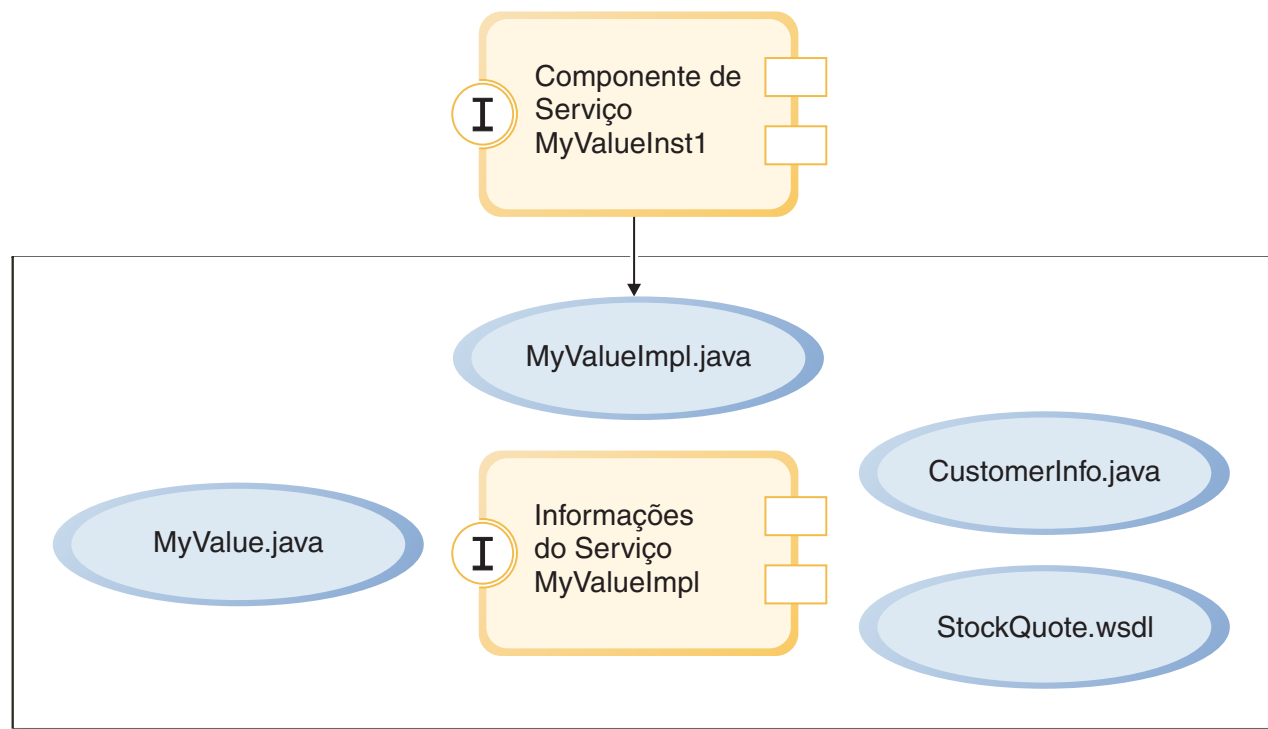
Na figura abaixo, temos um componente ao centro. É uma implementação, MyValueImpl, está em Java como sua interface. Ele possui duas referências: outra interface Java e uma interface WSDL.



Ao trabalhar com este componente, como mostrado abaixo, você vê efetivamente apenas o componente em si. Uma referência a este componente de outro componente seria visualmente revelada por uma linha para sua interface. Um referência deste componente seria revelada por uma linha a partir de seu ponto

de referência para a interface de outro componente. Uma referência representa um serviço que este componente consome. Ao nomear uma referência e especificar apenas sua interface, permite que o autor da implementação do componente para adiar a ligação que faz referência a um serviço real até posteriormente. Em um momento posterior, o especialista em integração fará isto ao executar a ligação da referência com a interface de outro componente ou importação. Este acoplamento vago, que permite o adiamento da ligação e reutilização de implementações, é uma das principais razões para utilização da Service Component Architecture do WebSphere Integration Developer.

Um componente também pode ter propriedades e qualificadores. Um qualificador é uma diretiva de QoS (Quality of Service) em interfaces e referências para o tempo de execução.



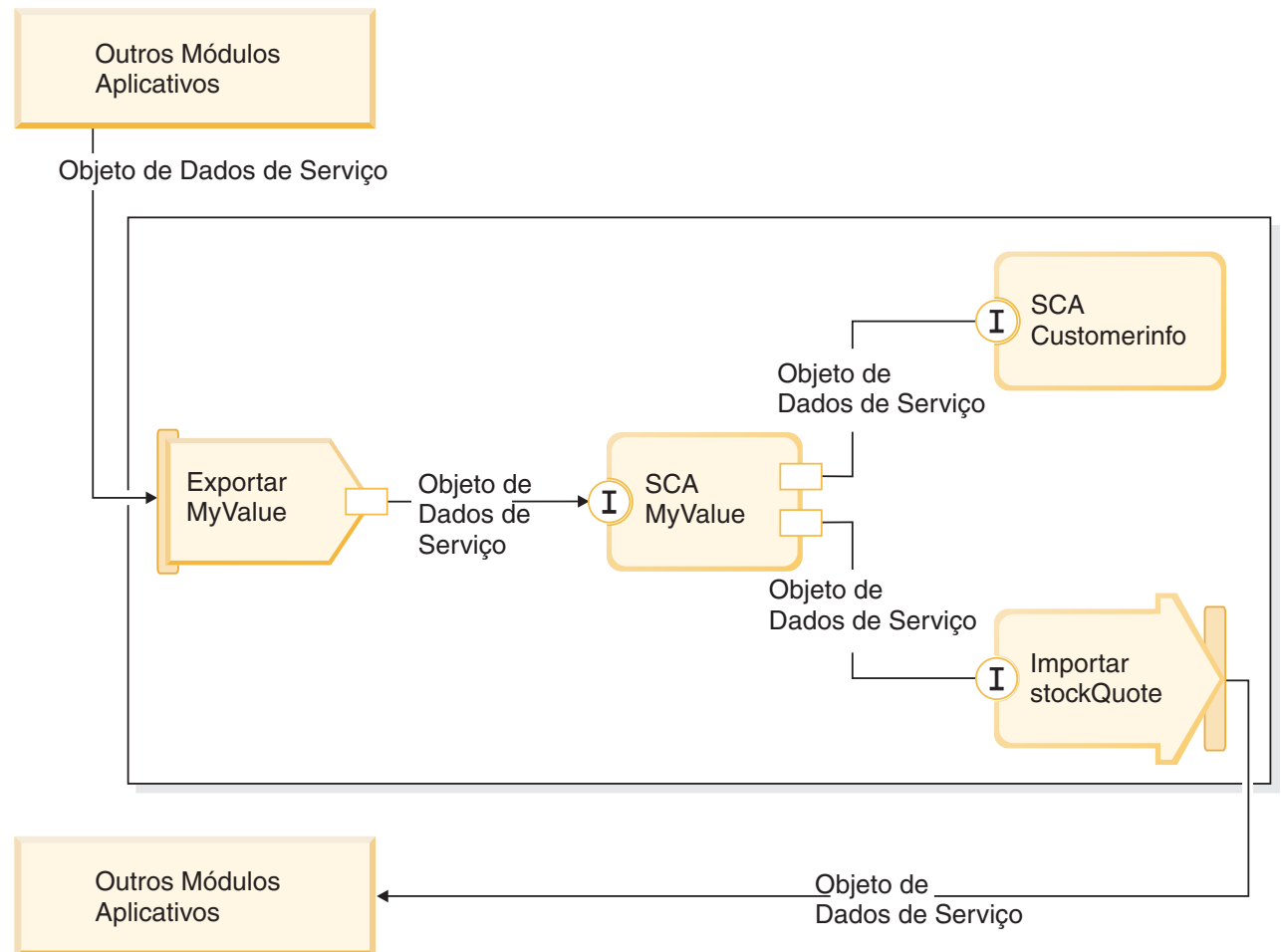
Objetos de Dados de Serviço

Os objetos de dados de serviço complementam a Service Component Architecture. A Service Component Architecture define os serviços como componentes e a conectividade entre eles. Os *objetos de dados de serviço* definem os dados fluindo entre os componentes.

Cada componente transmite informações como entrada e saída. Quando um serviço é chamado, os objetos de dados são transmitidos como um documento XML com codificação literal de documento ao utilizar um tipo de porta WSDL ou um objeto Java ao utilizar uma interface Java. Os objetos de dados são a forma preferida para dados e metadados nos serviços da Service Component Architecture. Semelhantes aos componentes, os objetos de dados de serviço separam o objeto de dados de sua implementação. Por exemplo, um componente interage com pedidos de compra enquanto o pedido de compra em si pode utilizar JDBC, EJB, e assim por diante, para executar as atualizações nos dados. Os objetos de dados de serviço permitem que o desenvolvedor de integração se concentre em trabalhar com artefatos de negócios. Na verdade, os objetos de dados de serviço são transparentes para o desenvolvedor de integração. Eles são definidos por um JSR (Java Specification Request) de objetos de dados de serviços.

No diagrama a seguir, os objetos de dados de serviço são transmitidos de um serviço externo para uma exportação, de uma exportação para um componente, de um componente para um componente, de um

componente para uma importação e de uma importação para um serviço. Importações e exportações são discutidas em uma seção Importações e Exportações subsequente.



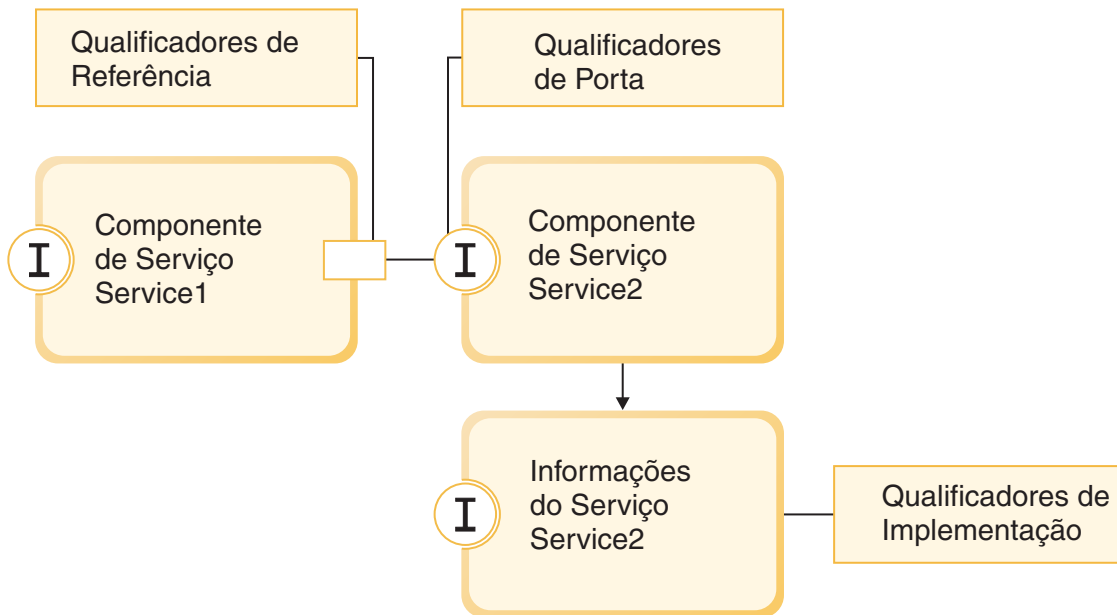
Qualificadores de Serviço

Um aplicativo comunica suas necessidades de QoS (Quality of Service) para o ambiente de tempo de execução especificando os *qualificadores de serviço*. Eles governam a interação entre um cliente de serviço e um serviço de destino.

Os qualificadores podem ser especificados em referências, interfaces e implementações do componente de serviço. Visto que a declaração dos valores de QoS são externas para uma implementação, você pode alterar esses valores sem alterar a implementação ou configurá-los de forma diferente quando várias instâncias da mesma implementação forem utilizadas em contextos diferentes.

Estas são as categorias dos qualificadores:

- Transação - regras para o tipo de transação
- Sessão de atividade - regras para unir a sessão ativa
- Segurança - regras para permissão
- Confiabilidade assíncrona - regras para entrega de mensagens assíncronas



Módulos

Um *módulo* é uma unidade de implementação que determina quais artefatos são empacotados juntos em um arquivo EAR (Enterprise Archive). Componentes dentro de um módulo são co-locados para obter desempenho e podem transmitir seus dados através de referências. Um módulo pode ser visto como um mecanismo de definição de escopo; ou seja, ele define um limite organizacional para os artefatos.

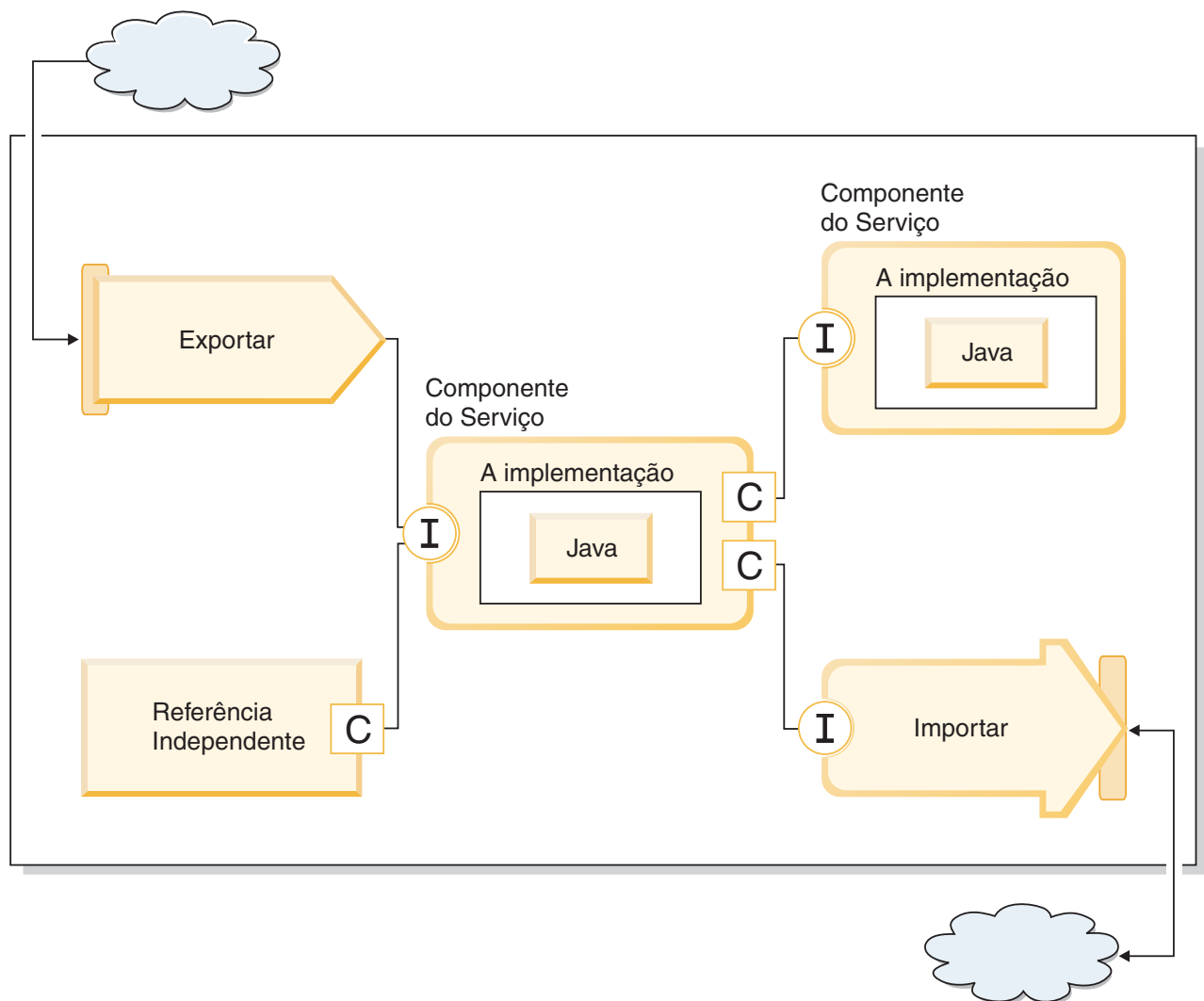
Um módulo é um composto de componentes de serviço, importações e exportações. Os componentes de serviços, importações e exportações residem no mesmo projeto e pasta raiz, que também contém a ligação que vincula os componentes e as ligações necessários para as importações e exportações. Um módulo também pode conter as implementações e interfaces referenciados por seus componentes, importações e exportações, ou pode ser colocado em outros projetos, como um projeto de biblioteca.

Há dois tipos de módulos. Primeiro, um módulo denominado *módulo* (algumas vezes referidos como um módulo de integração de negócios) que contém uma opção de diversos tipos de componentes, sempre utilizados para suportar um processo de negócios. Segundo, um módulo denominado *módulo de mediação*, que contém até um componente, um componente de fluxo de mediação, mais zero ou mais componentes Java que aumentam o componente de fluxo de mediação.

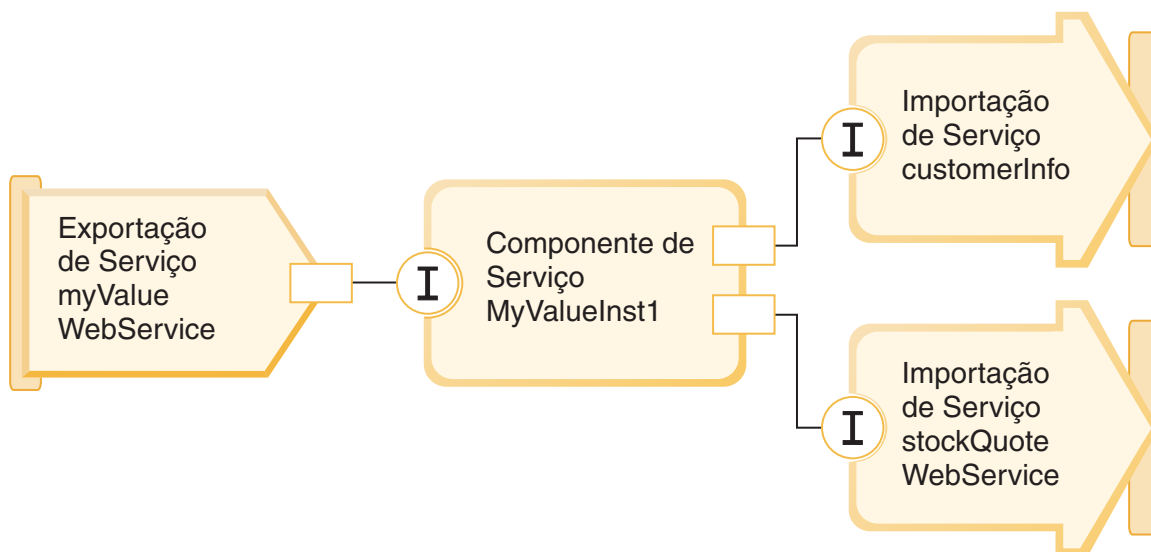
Por que há dois tipos de módulo? O primeiro tipo de módulo é projetado principalmente para processos de negócios. Um módulo de mediação é como um gateway para serviços externos existentes, que é comum em arquiteturas de barramento de serviços corporativas. Estes serviços externos ou exportações são acessados em um módulo de mediação pelas importações ou fornecedores de serviços. Ao desacoplar os solicitantes de serviço de cliente dos fornecedores de serviços através de um fluxo de mediação, seus aplicativos ganham flexibilidade e resiliência, um objetivo da arquitetura orientada a serviços. Por exemplo, seu fluxo de mediação pode registrar mensagens que chegam, rotear mensagens para um serviço específico determinado no tempo de execução ou transformar dados para torná-los adequados para transmissão para outro serviço. Estas funções podem ser incluídas e alteradas ao longo do tempo, sem modificar o solicitante ou os fornecedores de serviços.

Um módulo resulta em um aplicativo de serviço testado e implementado no WebSphere Process Server. Um módulo de mediação resulta em um aplicativo de serviço testado e implementado em um WebSphere Process Server ou em um servidor WebSphere Enterprise Service Bus. Ambos os tipos de módulos suportam importações e exportações.

Implementações, interfaces, objetos de negócios, mapas de objetos de negócios, funções, relacionamentos e outros artefatos sempre precisam ser compartilhados entre módulos. Uma *biblioteca* é um projeto utilizado para armazenar estes recursos compartilhados.



No diagrama a seguir, o módulo contém uma exportação, duas importações e um componente de serviço que as utiliza. A ligação é mostrada ligando as interfaces e referências.



Os artefatos de módulo e de mediação incluem:

- Definição do módulo - define o módulo.
- Componentes de serviço - definições dos serviços no módulo. O nome de um componente de serviço dentro de um módulo é exclusivo. Entretanto, um componente de serviço pode ter um nome de exibição arbitrários, que é normalmente um nome mais útil para um usuário.
- Importações - definições de importações, que são chamadas para serviços externos a este módulo. As importações possuem ligações, que são discutidas na seção Importações e Exportações.
- Exportações - definições de exportações, que são utilizadas para expor componentes aos responsáveis pelas chamadas que são externos a este módulo. As exportações possuem ligações, que são discutidas na seção Importações e Exportações.
- Referências - referências de um componente para outro no módulo.
- Referências independentes - aplicativos de referência que não estão definidos como componentes Service Component Architecture (por exemplo, JavaServer Pages), que possibilitam que estes aplicativos interajam com componentes Service Component Architecture. Pode haver apenas um artefato de referência independente por módulo.
- Outros artefatos - esses artefatos incluem arquivos WSDL, classes Java, arquivos XSD, processos BPEL, entre outros.

Importações e Exportações

Importações e exportações definem as interfaces externas ou pontos de acesso de um módulo. *Importações* identificam serviços fora de um módulo, portanto podem ser chamadas de dentro do módulo. *Exportações* permitem que componentes forneçam seus serviços para clientes externos. As informações sobre ligação são requeridas para uma importação ou exportação. Várias ligações estão disponíveis e um aviso é fornecido sobre qual tipo deve ser apropriado para seu aplicativo.

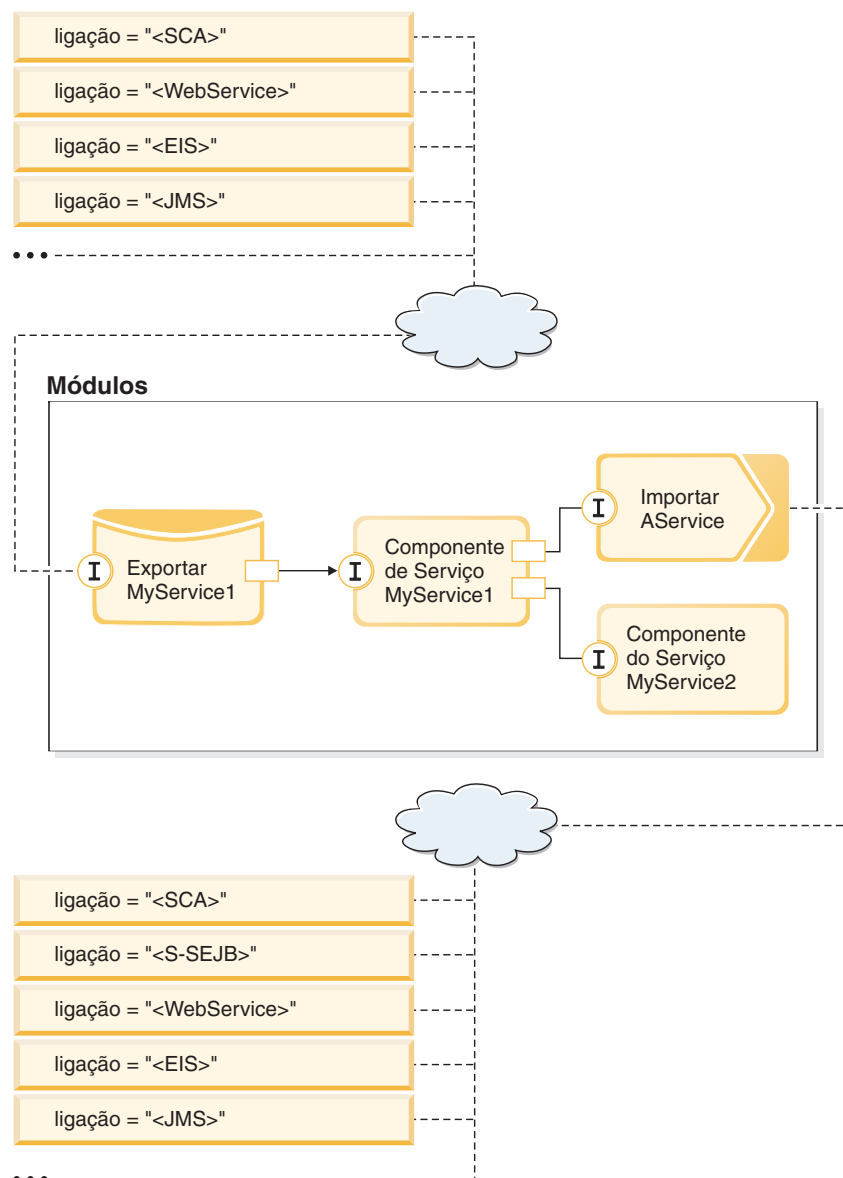
Tipos de Ligação da Importação e Exportação de Serviços

As importações e exportações exigem informações de ligação, que especificam os meios de transporte de dados dos módulos. Uma *ligação de importação* descreve a maneira específica de ligar um serviço externo a um componente de importação. Uma *ligação de exportação* descreve a maneira específica como os serviços de um módulo são disponibilizados para os clientes.

A SCA ou a ligação padrão permite que seu serviço se comunique com outros serviços em outros módulos. Uma importação com uma ligação SCA permite acessar um serviço em outro módulo. Uma

exportação com uma ligação SCA permite oferecer um serviço para outros módulos. Uma ligação de importação de serviço da Web permite ligar um serviço da Web externo a uma importação. Uma ligação de exportação de serviço da Web permite fornecer um serviço a clientes externos como um serviço da Web. O assistente de descoberta de serviços corporativos cria importações e exportações representando um serviço em um sistema EIS. As ligações criadas são para um tipo EIS ou JMS (Java Message Service). Uma ligação EIS fornece comunicação síncrona com o serviço no sistema EIS. Uma ligação JMS seria normalmente utilizada em interações com grandes sistemas EIS nos quais a comunicação assíncrona pelas filas de mensagens é crítica para a confiabilidade. Uma importação (embora não uma exportação) também pode ter uma ligação EJB de sessão sem preservação de estado.

O editor de montagem lista as ligações suportadas e simplifica a criação delas quando você deseja criar uma importação ou exportação. Uma visualização de propriedades no editor de montagem exibe as informações de ligação de qualquer importação ou exportação.



Selecionando Ligações Apropriadas

Quando uma determinada ligação é mais apropriada às necessidades do seu aplicativo é discutido nesta seção.

As ligações disponíveis no WebSphere Integration Developer fornecem uma variedade de opções. Esta lista o ajuda a identificar quando um tipo de ligação poderia ser mais adequado às necessidades do seu aplicativo do que outro tipo de ligação.

Considere uma ligação *SCA* quando estes fatores são aplicáveis:

- Todos os serviços estão contidos nos módulos do WebSphere Integration Developer; ou seja, não há serviços externos
- O desempenho é importante
- Os módulos estão firmemente acoplados.

Considere uma ligação de *Serviços da Web* quando estes fatores são aplicáveis:

- Você precisa acessar um serviço externo sobre a Internet ou fornecer um serviço sobre a Internet
- Os serviços estão vagamente acoplados
- O protocolo de um serviço externo que você está acessando ou um serviço que você deseja fornecer for SOAP/HTTP ou JMS/HTTP.

Considere uma ligação *EIS* quando estes fatores são aplicáveis:

- Você precisa acessar um serviço em um sistema EIS utilizando um adaptador de recursos
- O desempenho é mais importante do que a confiabilidade; ou seja, a transmissão de dados síncrona é preferida sobre a assíncrona.

Considere uma ligação *JMS* quando estes fatores são aplicáveis:

- Você precisa acessar um sistema de mensagens
- Os serviços estão vagamente acoplados
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Considere uma ligação *EJB de Sessão sem Preservação de Estado* quando estes fatores são aplicáveis:

- A ligação é para um serviço importado que é um próprio EJB
- O serviço importado é vagamente acoplado
- O estado do EJB não é importante.

Tipos de Implementação de Serviço

Os tipos de implementação de serviço são as implementações dos componentes de serviço.

As implementações padrão de serviços são descritas nesta seção. Essas implementações aparecerão em serviços no editor de montagem e/ou dentro de processos BPEL.

Objetos Java

Uma implementação de um componente no Java é referida como um objeto Java.

Uma implementação comum é um componente escrito em Java. Às vezes, esta implementação é conhecida pelo pseudônimo "plain old Java object" ou POJO. Geralmente, essa implementação tem um tipo de interface WSDL, embora também possa ter uma interface Java. Se houver várias interfaces especificadas, não é possível misturar interfaces WSDL com interfaces Java. No entanto, você pode "unir" um aplicativo criado com um conjunto de interfaces WSDL a um aplicativo com um conjunto de interfaces Java. Uma amostra listada na Samples Gallery da visualização Bem-vindo mostra como isso é possível.

Quando você trabalha com um objeto Java, o código permanece oculto dentro do contexto dos editores.

Um objeto Java pode ser utilizado em um módulo de mediação. Ele pode ser implementado em um WebSphere Process Server ou em um servidor WebSphere Enterprise Service Bus.

Processo BPEL

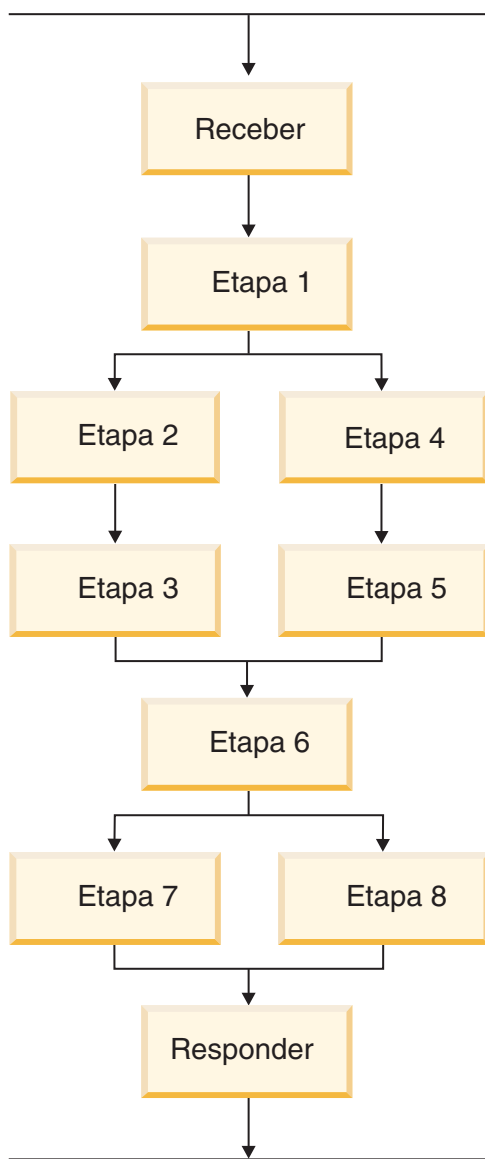
Um componente do *processo BPEL* implementa um processo de negócios.

Sua linguagem de implementação é o Business Process Execution Language for Web Services (BPEL4WS) padrão de mercado e suas extensões IBM. Um processo BPEL implementa um serviço potencialmente de longa execução através da utilização de mais serviços elementares. Um processo BPEL criado no editor de processo pode executar as seguintes ações:

- Descrever a orquestração de outros serviços utilizando gráficos de fluxo de controle
- Utilizar variáveis para manter o estado do processo
- Utilizar manipulação de erros sofisticada através de manipulação de falhas
- Suportar eventos assíncronos
- Correlacionar pedidos de entrada com a instância correta de um determinado processo utilizando conjuntos de correlação para marcar esses dados de negócios no pedido que identifica a instância (por exemplo, um ID do cliente)
- Fornecer transações estendidas através de suporte de compensação sofisticado

Além desses itens padrão do BPEL, o WebSphere Integration Developer também estende o BPEL para incluir pessoas em um processo com suporte à *tarefa humana*. Por exemplo, essa extensão poderia incluir em um processo o requisito de aprovação de empréstimo de uma pessoa.

O editor de processo utiliza representações visuais de constructos do BPEL para construir seu processo de negócios de maneira rápida e simples.

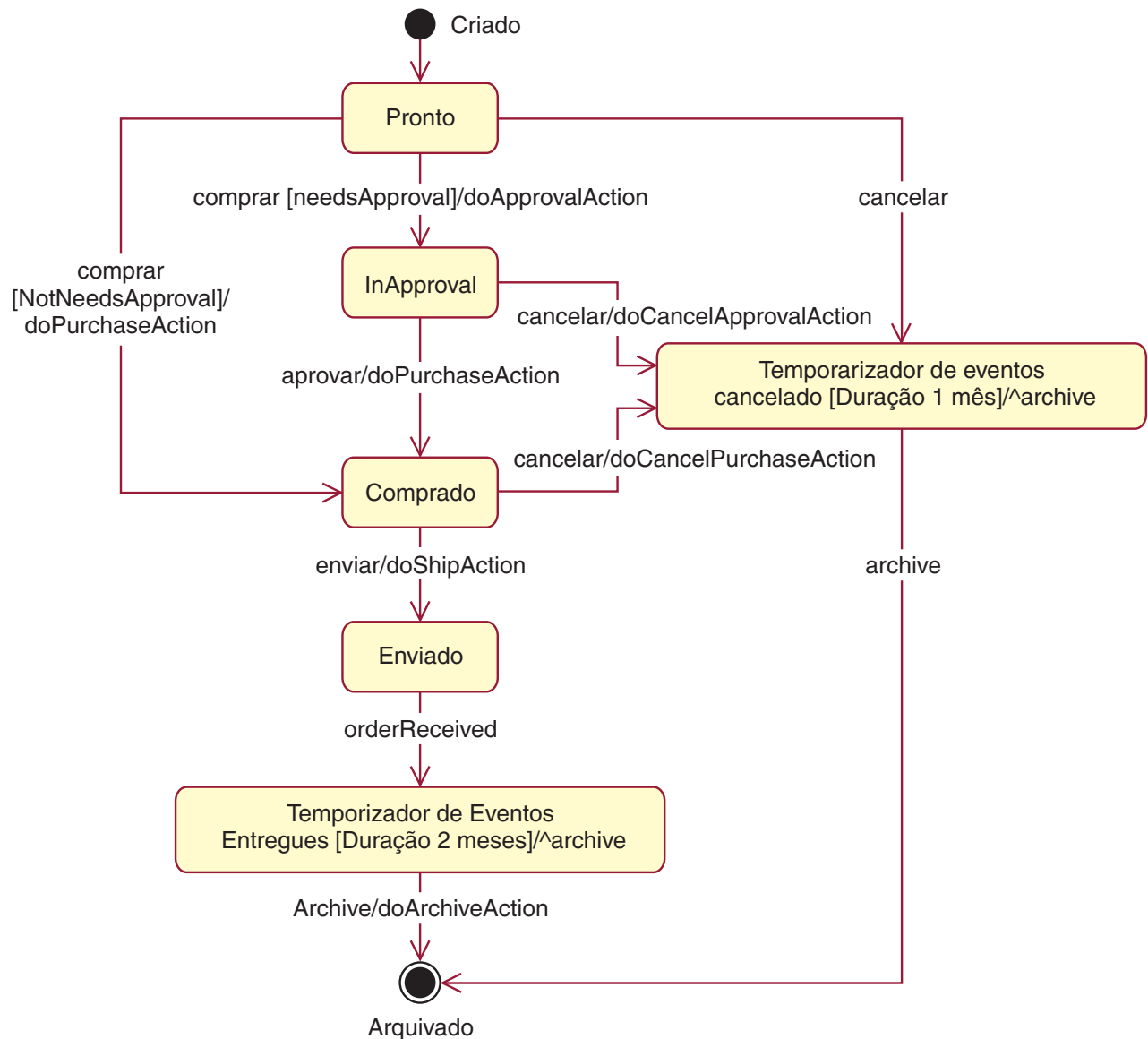


Um processo BPEL não pode ser utilizado em um módulo de mediação. Ele pode ser implementado apenas em um WebSphere Process Server.

Máquinas de Estado

Uma máquina de estado é uma maneira alternativa de criar um processo de negócios. Uma máquina de estado é ajustada para processos relacionados aos estados de alteração ao invés de um fluxo de controle. Um estado define o que um artefato pode fazer em um determinado momento. A *máquina de estado* é uma implementação desse conjunto de estados.

As máquinas de estado são uma maneira comum de mostrar um conjunto de estados relacionados em um processo. Uma máquina de estado conhecida é a máquina de bebidas. Você insere algumas moedas na máquina e, junto com sua bebida, que felizmente é entregue, você recebe seu troco exato conforme a máquina de estado automaticamente despeja as moedas que precisam ser devolvidas a você, de acordo com as moedas inseridas. No diagrama a seguir, uma máquina de estado típica é mostrada conforme criada pelo editor de máquina de estado. Na máquina de estado, um item é comprado e fornecido para um cliente.



Uma máquina de estado não pode ser utilizada em um módulo de mediação. Ele pode ser implementado apenas em um WebSphere Process Server.

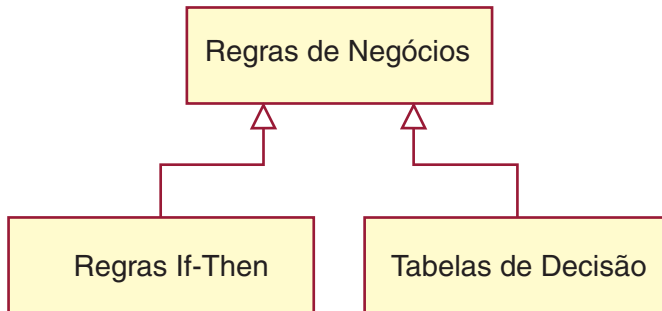
Regras de Negócios

As regras de negócios complementam processos de negócios e máquinas de estado. Se houver uma condição com uma variável, por exemplo, uma *regra de negócios* pode alterar o valor em tal variável no tempo de execução. Criada por uma linguagem de programação visual, uma regra de negócios toma uma decisão baseada no contexto. A decisão pode ser simples ou complexa. As regras de negócios são não-procedurais e as regras podem ser alteradas independente de um aplicativo.

As regras de negócios determinam o resultado de um processo com base em um contexto. Elas são utilizadas em situações diárias de negócios para a tomada de decisões, dado um conjunto específico de circunstâncias. Essa decisão pode exigir muitas regras para incluir todas as circunstâncias. As regras de negócios dentro de um processo de negócios permitem que os aplicativos respondam rapidamente às condições de negócios em alteração. Em uma companhia de seguros, por exemplo, uma regra de negócios para aprovar um seguro de automóvel para um candidato poderia ser: *Se o candidato for homem, tiver mais*

de 25 anos, se o carro for da categoria esportiva e se ele tiver feito seguros com a empresa nos últimos 5 anos, a proposta de seguro seria aprovada a um valor de \$100 por mês.

O WebSphere Integration Developer oferece inúmeras abordagens para a criação de regras de negócios. Você pode criar regras if-then ou tabelas de decisão, todas estarão adaptadas ao resultado do seu processo. Observe que essas regras são independentes do processo em si, o que significa que você pode alterar as regras a qualquer momento sem ter que refazer seu processo. Por exemplo, com base na localização do seu negócio, você pode ter uma regra que diz: *Se a data for entre 26 de dezembro e 1 de janeiro, ofereça um desconto nas vendas pós feriado de 20%*. No entanto, se as vendas continuassem muito baixas, você poderia modificar o desconto para 40% a qualquer momento.

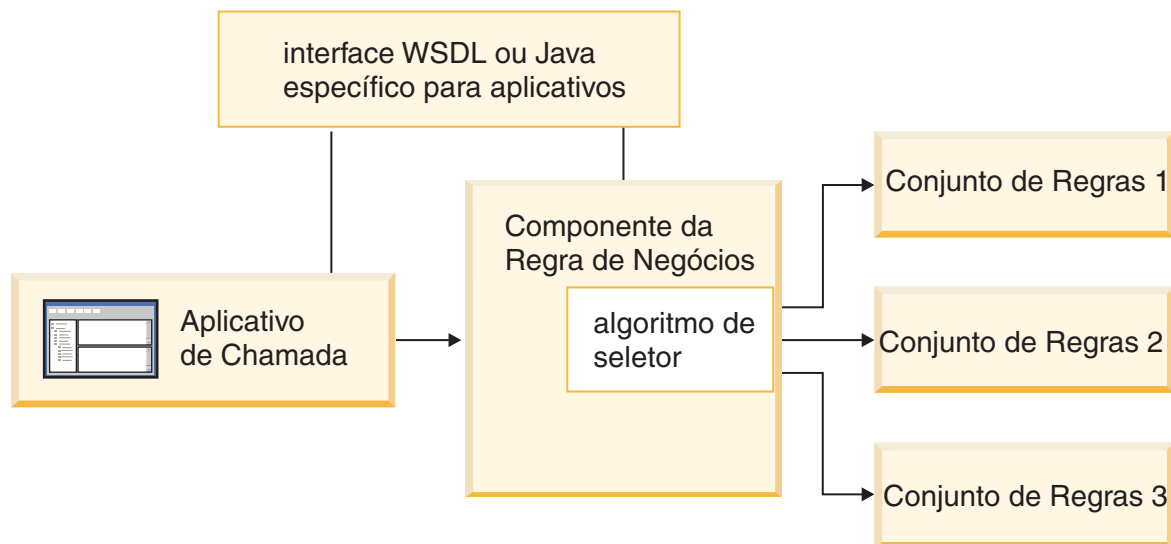


As regras de negócios não podem ser utilizadas em um módulo de mediação. Elas podem ser implementadas apenas em um WebSphere Process Server.

Seletores

Aplicativos integrados contêm muitas maneiras de interagir. Um *seletor* é utilizado para rotear uma operação de um aplicativo cliente para um dos diversos possíveis componentes para implementação.

O roteamento de um componente é baseado em datas. Por exemplo, aqui está uma rota baseada em uma data: *Duas semanas antes do início das aulas, oferecer um preço especial de volta às aulas em seus materiais escolares*. Os negócios podem ter muitas dessas rotas baseadas em datas. Um seletor toma uma decisão para selecionar uma rota sobre outra no tempo de execução com base em uma data. Por exemplo, se o período for exatamente antes do início das aulas, a oferta anterior de volta às aulas seria chamada. No entanto, se o período for de encerramento das aulas, a oferta seria para preparar os estudantes para o verão.

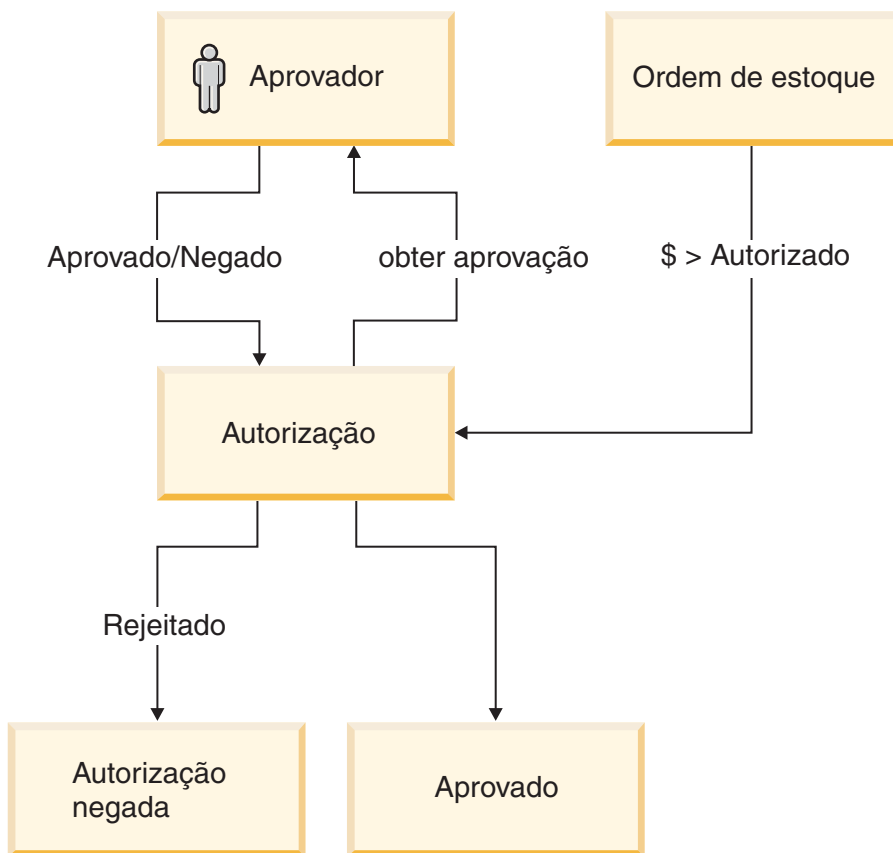


Um seletor não pode ser utilizado em um módulo mediação. Ele pode ser implementado apenas em um WebSphere Process Server.

Tarefa Humana

Um componente de *tarefa humana* implementa uma tarefa executada por uma pessoa. Ele representa o envolvimento de uma pessoa em um processo de negócios.

Ocasionalmente, as pessoas precisam intervir em um processo de negócios. Por exemplo, um cliente deseja comprar um item que está acima de seu limite de crédito. Uma tarefa humana permite que você intervenha e substitua uma regra de negócios que impede o cliente de fazer a compra. Uma tarefa humana pode ter atributos, como definir o proprietário da tarefa e fornecer um processo de escalção, caso a pessoa especificada não esteja disponível. O componente de tarefa humana reconhece a realidade de que muitos processos requerem intervenção humana para tarefas como revisão, pesquisa e aprovação.



Uma tarefa humana não pode ser utilizada em um módulo de mediação. Ele pode ser implementado apenas em um WebSphere Process Server.

Mapa de Interface

Um *mapa de interface* resolve diferenças entre as interfaces dos componentes em interação.

Diferenças entre as interfaces nos componentes que precisam interagir com outro são comuns. Estas diferenças ocorrem pois no WebSphere Integration Developer você está sempre montando componentes que foram criados para aplicativos diferentes. Reutilizá-los para criar um novo aplicativo é um dos pontos fortes do WebSphere Integration Developer, já que de outra forma, você poderia recodificar componentes semelhantes. Mas, normalmente, você precisa fazer alguns ajustes.

Por exemplo, dois componentes podem ser métodos que executam basicamente a mesma ação, mas possuem nomes diferentes como `getCredit` e `getCreditRating`. Eles também podem ter nomes de operações diferentes e as operações podem ter tipos de parâmetros diferentes. Um mapa de interface mapeia as operações e os parâmetros destes métodos, desta forma as diferenças são resolvidas e os dois componentes podem interagir. Um mapa de interface é como uma ponte entre as interfaces de dois componentes, permitindo que eles sejam conectados, independente das diferenças.

Um mapa de interface existe independente dos componentes que o utilizam, o que significa que os próprios componentes não precisam ser alterados.

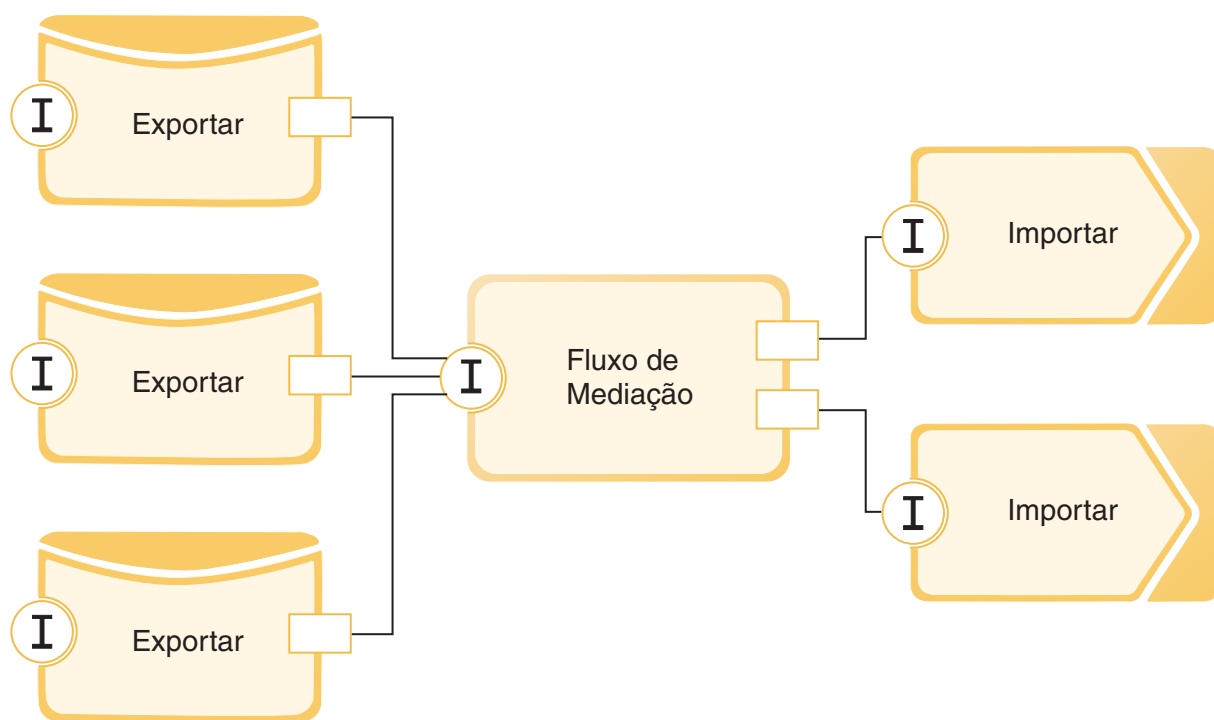
Um mapa de interface não pode ser utilizado em um módulo de mediação. Ele pode ser implementado apenas em um WebSphere Process Server.

Fluxo de Mediação

Mediação é uma maneira de mediar ou intervir dinamicamente entre serviços. Um *fluxo de mediação* implementa uma mediação.

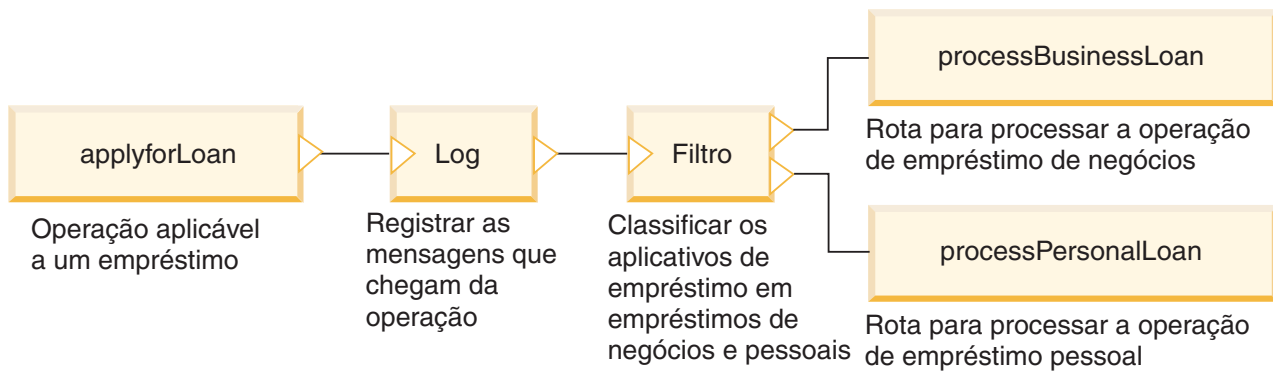
A mediação possui várias funções úteis. Por exemplo, você pode utilizar a mediação quando precisar transformar dados de um serviço em um formato aceitável para um serviço subsequente. O registro permite que você registre mensagens de um serviço antes que elas sejam enviadas para o próximo serviço. O roteamento permite que você roteie dados de um serviço para um serviço apropriado determinado pelo fluxo de mediação. Uma mediação opera independente dos serviços aos quais está conectada. Uma mediação no editor de montagem aparece como um fluxo de mediação entre exportações e importações.

No diagrama a seguir, três solicitantes de serviços ou exportações enviam seus dados de saída para a interface do componente de fluxo de mediação. O componente de fluxo de mediação então roteia os dados apropriados para dois provedores de serviços ou importações.



Um fluxo de mediação é um constructo semelhante a um fluxo criado com o editor de fluxo de mediação. Selecionar um componente de fluxo de mediação no editor de montagem ativa o editor de fluxo de mediação. No editor de fluxo de mediação, uma operação de um serviço, o solicitante de serviços ou a exportação, é mapeado para a operação de outro serviço, o provedor de serviços ou a importação, juntamente com funções fornecidas pela mediação do fluxo de mediação. Estas funções são denominadas *primitivas de mediação* e são conectadas em um fluxo de mediação, conforme mostrado no diagrama a seguir. As primitivas de mediação são fornecidas pela IBM ou você pode criar suas próprias primitivas customizadas. Primitivas de mediação podem atuar tanto no conteúdo como no contexto da mensagem, em que o contexto são informações específicas sobre a ligação como cabeçalhos SOAP ou JMS, ou propriedades definidas pelo usuário.

No diagrama que segue uma operação, `applyforLoan`, envia uma mensagem primeiro para uma primitiva de registro, `Log`, que registra a mensagem. `Log` envia a mensagem para a primitiva `Filter`, que, dependendo da mensagem, roteia a mensagem para uma operação `processBusinessLoan` ou `processPersonalLoan`.



Conforme discutido na seção Módulos, há um módulo de mediação para um componente de fluxo de mediação. Ele pode conter até um componente de fluxo de mediação mais zero ou mais componentes Java que aumentam o componente de fluxo de mediação. Um módulo de mediação pode ser implementado em um WebSphere Process Server ou em um servidor WebSphere Enterprise Service Bus.

Referências independentes

Referências Independentes são referências para aplicativos que não são definidas como componentes Service Component Architecture (por exemplo, JavaServer Pages ou servlets). Referências independentes permitem que estes aplicativos interajam com componentes da Service Component Architecture.

Referências independentes não possuem uma interface nem uma implementação (já que a implementação está fora do escopo do módulo). Um módulo pode conter referências independentes ou um artefato de referências independentes. Referências independentes possuem o valor prático de permitir que você utilize seus aplicativos existentes junto com componentes Service Component Architecture criados no WebSphere Integration Developer.

Referências independentes podem ser utilizadas em um módulo de mediação. Elas podem ser implementadas em um WebSphere Process Server ou em um servidor WebSphere Enterprise Service Bus.

Informações Relacionadas

Vários tópicos relacionados à arquitetura são discutidos nesta seção.

Estes tópicos fornecem algumas informações adicionais associadas à arquitetura do produto.

- “Desenvolvendo Serviços a Serem Utilizados com Serviços .NET”
- “Suporte Bidirecional” na página 25

Desenvolvendo Serviços a Serem Utilizados com Serviços .NET

Se você pretende desenvolver serviços para serem utilizados com os serviços .NET, você precisa estar atento a algumas considerações especiais. Há dois aspectos sobre estas considerações: trazer arquivos WSDL desenvolvidos em um ambiente de desenvolvimento .NET no ambiente de desenvolvimento do WebSphere Integration Developer e exportar arquivos WSDL do WebSphere Integration Developer de forma que seja possível utilizá-los com os serviços .NET. A principal diferença entre os dois ambiente de desenvolvimento é que o ambiente .NET utilize esquemas seqüenciais e o WebSphere Integration Developer não. Esquemas seqüenciais são uma maneira de incluir o esquema em um arquivo WSDL ao invés de especificar que ele deve ser importado como um arquivo separado. Foram fornecidas informações no WebSphere Integration Developer para ajudá-lo em ambos os casos.

Os Esquemas Seqüenciais na seção Criando Interfaces mostram como importar arquivos WSDL que contêm esquemas seqüenciais como os arquivos WSDL .NET teriam.

Criando Proxies para Trabalhar com Serviços .NET na seção Montando o Serviço de Negócios mostra como publicar seus serviços externamente de maneira que seja possível trabalhar com os serviços .NET. Este tópico está particularmente focado na criação de proxies que trabalharão com os serviços .NET.

Suporte Bidirecional

O WebSphere Integration Developer funciona em um ambiente multilíngüe. Isto significa que ele pode exibir e manipular dados representados em diferentes idiomas. Diversos idiomas que possuem script bidirecionais (por exemplo, árabe ou hebraico) estão incluídos neste suporte. Tais idiomas são escritos da direita para a esquerda, enquanto os números são segmentos incorporados de texto em latim (ou cirílico ou grego, e assim por diante) que são incorporados neste texto da esquerda para a direita.

Na Visão Geral do Suporte ao Script Bidirecional no IBM WebSphere Integration Developer o suporte para idiomas bidirecionais é discutido, incluindo a configuração requerida, alguns pontos técnicos específicos ao utilizar o suporte e limitações.

Capítulo 3. Aprendendo sobre as Ferramentas

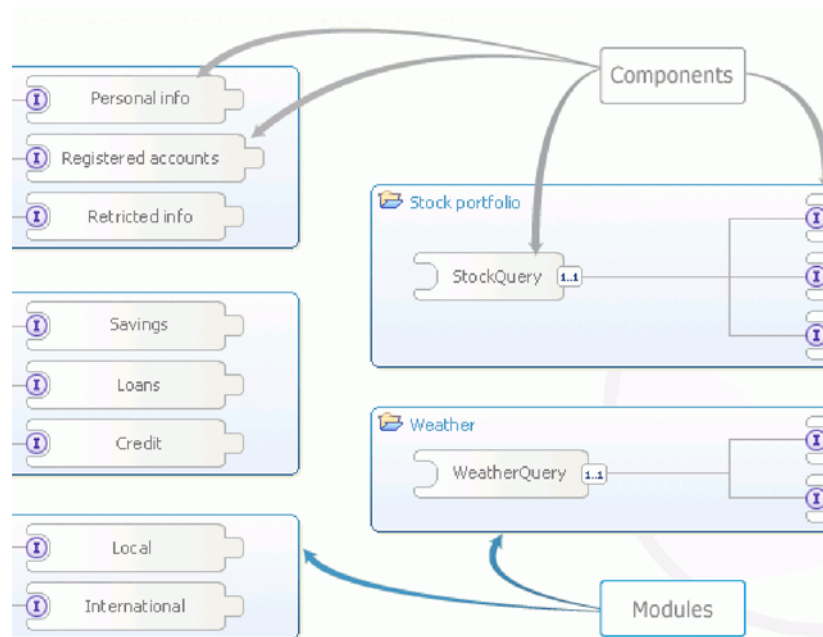
O WebSphere Integration Developer utiliza ferramentas visuais para criar aplicativos integrados baseados em arquitetura orientada a serviços.

A seção visão geral da visualização Bem-vindo apresenta o nível mais alto de entendimento das ferramentas do WebSphere Integration Developer. Neste caso, nível mais alto significa a apresentação mais abstrata das ferramentas. O gráfico explica o que as ferramentas fazem. Leia a seção visão geral caso ainda não conheça o produto e se quiser entender rapidamente o que suas ferramentas fazem. A seção de tutoriais, que também pode ser localizada na visualização Bem-vindo, permite que você observe as ferramentas em ação. A seção de amostras permite que você comece a utilizar as ferramentas, mas de uma forma à prova de falhas. Finalmente, conforme você começa a desenvolver aplicativos com o WebSphere Integration Developer, utilize o centro de informações para localizar informações detalhadas sobre cada ferramenta, incluindo informações conceituais, tarefas que você pode executar e informações de referência sobre cada ferramenta.

Visão Geral da Visualização Bem-vindo

A seção visão geral da visualização Bem-vindo é o exame de nível mais alto das ferramentas e dos recursos no produto. Dentro da visão geral, os gráficos são utilizados para conduzir o que as principais ferramentas fazem no produto.

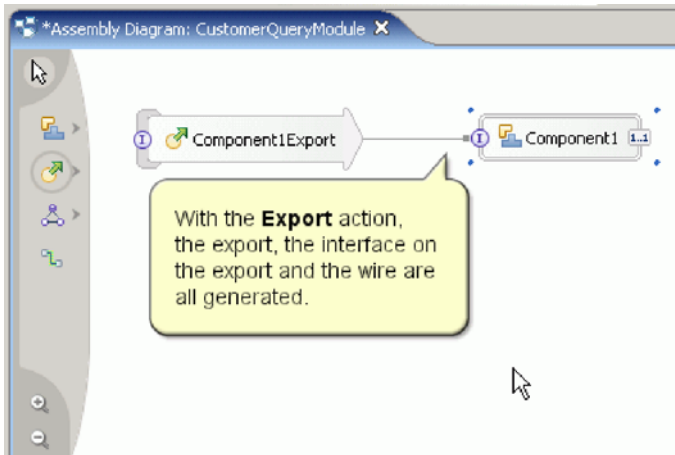
A seção de visão geral da visualização Bem-vindo é uma maneira rápida de aprender sobre o produto antes de começar a trabalhar com ele de uma maneira prática. A maioria dos usuários que não conhece o produto deve começar por ela. A seleção de um ícone na visão geral ativa um gráfico em uma ferramenta ou em um tópico. Por exemplo, o relacionamento dos componentes e módulos é mostrado no diagrama a seguir. Você também pode ativar um tour audiovisual do produto.



Tutoriais na Visualização Bem-vindo

Os tutoriais permitem que você veja ferramentas em ação de maneira segura contra falhas. Você seleciona um tutorial e, em seguida, observa como ele continua executando algumas tarefas como faria um usuário.

Após entender o produto no nível conceitual, você pode trabalhar com as ferramentas ativando um tutorial. Cada tutorial é uma apresentação em forma de filme de uma determinada ferramenta. Além de assistir ao tutorial, um texto instantâneo descreve o que você está vendo. Os tutoriais do WebSphere Integration Developer são localizados na seção Veja e Aprenda da galeria do tutorial da visualização Bem-vindo.



Amostras na Visualização Bem-vindo

Nossas amostras fazem uma exposição prática na utilização do WebSphere Integration Developer para desenvolver soluções de serviço de negócios em um ambiente à prova de falhas.

As amostras podem ser localizadas na Samples Gallery da visualização Bem-vindo. A seleção de uma amostra oferece várias opções. Você pode construí-la sozinho seguindo as instruções passo a passo ou pode tê-la construída para você.

As amostras variam em escopo. Uma amostra de tecnologia concentra-se em uma determinada ferramenta executando uma tarefa específica. Uma amostra de aplicativo utiliza várias ferramentas para obter um resultado mais complexo. Um cenário é uma amostra prolongada que constrói um grande aplicativo com várias ferramentas. Cada amostra lista a quantidade de tempo que levará para construir.

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 Informações

O centro de informações fornece informações completas sobre o produto. Além dos tutoriais e das amostras, tarefas conceituais adicionais e informações de referência podem ser localizadas para cada ferramenta para o WebSphere Integration Developer.

A navegação do centro de informações oferece acesso rápido às informações desejadas. Você também pode fazer sua procura em todo o centro de informações. Faça sua seleção a partir dos resultados da procura para ver vários tópicos contendo informações detalhadas sobre o assunto, além de links para informações relacionadas.



Avisos

A publicação XDoclet incluída neste produto IBM é utilizada com permissão e é coberta pela seguinte declaração de atribuição de direitos autorais: Direitos Autorais (c) 2000-2004, XDoclet Team. Todos os direitos reservados.

Partes baseadas em *Design Patterns: Elements of Reusable Object-Oriented Software*, de Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, Direitos Autorais (c) 1995 da Addison-Wesley Publishing Company, Inc. Todos os direitos reservados.

Direitos Restritos para Usuários do Governo dos Estados Unidos - Uso, duplicação e divulgação restritos pelo documento GSA ADP Schedule Contract com a IBM Corporation.

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre os produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM, não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitação de patentes pendente abrangendo o assunto descrito nesta documentação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença podem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
IBM Corporation
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

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

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS (OU CONDIÇÕES) DE NÃO-INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta disposição pode não se aplicar ao Cliente.

Estas informações podem conter imprecisões técnicas ou erros tipográficos. Periodicamente são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte deste produto IBM e sua utilização é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
IBM Corporation
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença do Programa Internacional IBM ou de qualquer outro contrato equivalente.

Todos os dados sobre desempenho aqui descritos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas de nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não-IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não-IBM. Dúvidas sobre os recursos de produtos não-IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos contêm nomes de indivíduos, empresas, marcas e produtos. Todos esses nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de exemplo na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de exemplo sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas. O Cliente pode copiar, modificar e distribuir estes programas de exemplo de qualquer maneira sem pagamento à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

Cada cópia ou parte destes programas de exemplo ou qualquer trabalho derivado deve incluir um aviso de copyright com os dizeres:

(C) (nome da empresa) (ano). Partes deste código são derivadas dos Programas de Exemplo da IBM Corp. (C) Direitos Autorais IBM Corp. 2000, 2005. Todos os direitos reservados.

Se estas informações estiverem sendo exibidas em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

Informações sobre a Interface de Programação

As informações sobre interface de programação destinam-se a facilitar a criação de software aplicativo utilizando este programa.

As interfaces de programação de uso geral permitem que o Cliente desenvolva o software aplicativo que obtém os serviços das ferramentas deste programa.

Entretanto, também podem estar contidas informações de diagnóstico, modificação e ajuste. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Aviso: Não utilize estas informações sobre diagnósticos, modificações e ajustes como uma interface de programação, pois elas estão sujeitas a alterações.

Marcas Registradas e Marcas de Serviço

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



Impresso em Brazil

S517-8346-01

