

IBM Business Process Manager  
Versão 8 Edição 5

*Descrição Geral do IBM Business  
Process Manager*

**IBM**



---

## Os manuais em PDF e o Information Center

Os manuais em PDF são fornecidos como meio conveniente de impressão e de leitura offline. Para consultar as informações mais actualizadas, acesse ao Information Center online.

Enquanto conjunto, os manuais em PDF contêm o mesmo conteúdo que o Information Center. Algumas ligações presentes nos manuais em PDF foram personalizadas para serem utilizadas nos Information Centers e poderão não funcionar correctamente.

A documentação em PDF fica disponível 3 meses após qualquer publicação importante do Information Center, tal como a Versão 7.0 ou a Versão 7.5.

A documentação em PDF é actualizada com menos frequência que o Information Center, mas com mais frequência que o Redbooks. Geralmente, os manuais em PDF são actualizados quando existe um conjunto suficientemente grande de alterações aos manuais.



# Índice

## Os manuais em PDF e o Information Center . . . . . iii

## Capítulo 1. Introdução ao IBM Business Process Manager. . . . . 1

Descrição geral do produto . . . . .	1
Configurações do IBM Business Process Manager . . . . .	3
Capacidades de configuração do IBM Business Process Manager . . . . .	4
Repositório do Process Center . . . . .	4
Process Server e ambientes em tempo de execução . . . . .	6
Ambientes de criação . . . . .	6
Ferramentas de administração . . . . .	7
Acessibilidade no IBM Business Process Manager . . . . .	9
Disponibilidade de idiomas nacionais no IBM Business Process Manager . . . . .	9
Descrição geral da gestão de processos empresariais . . . . .	10
Descrição geral do processo de modelação . . . . .	11
Desenvolvimento de processos com o Process Center . . . . .	12
Process Applications: descrição geral . . . . .	13
Executar e depurar processos . . . . .	19
Instalar e gerir Process Applications . . . . .	20
Criar, aceder e incorporar serviços . . . . .	22
Aceder a serviços externos a uma aplicação . . . . .	22
Criar ou chamar um serviço da Web . . . . .	27
Saiba mais sobre os conceitos chave . . . . .	28
Criação de versões . . . . .	29
Criação de versões das Process Applications . . . . .	29
Arquivo de versões de módulos e bibliotecas . . . . .	30
Módulos e bibliotecas associados a Process Applications ou Toolkits . . . . .	31
Convenções de nomenclatura . . . . .	31
Convenções de nomenclatura para implementações do servidor do Process Center . . . . .	32
Convenções de nomenclatura para implementações do Process Server. . . . .	35
Associações adequadas à versão . . . . .	35
Chamada para execução dinâmica adequada à versão . . . . .	38
Implementar Process Application com projectos e módulos Java . . . . .	38
Implementar Process Applications com regras empresariais e selectores . . . . .	38
Objectos de configuração . . . . .	39
Arquitectura de implementação. . . . .	39
Células . . . . .	39
Servidores . . . . .	39
Servidores autónomos . . . . .	40
Conjuntos . . . . .	40
Perfis . . . . .	41
Gestores de implementação . . . . .	42
Nós . . . . .	42
Nós geridos . . . . .	43

Nós não geridos . . . . .	43
Agentes de nós . . . . .	43
Considerações de nome para perfis, nós, servidores, sistemas centrais e células. . . . .	43
BPMN 2.0 . . . . .	48
Definições de Processo Empresarial (BPDs) . . . . .	50
Associações . . . . .	51
Exportar e importar descrição geral de associação . . . . .	53
Exportar e importar configuração de associação . . . . .	56
Transformação de formato de dados em importações e exportações . . . . .	57
Selectores de função em associações de exportação. . . . .	61
Processamento de falhas . . . . .	63
Interoperacionalidade entre módulos SCA e serviços Open SCA . . . . .	67
Tipos de associação. . . . .	70
Selecionar associações apropriadas . . . . .	70
Associações SCA . . . . .	72
Associações do serviço da Web . . . . .	72
Associações HTTP . . . . .	96
Associações EJB . . . . .	104
Associações EIS. . . . .	111
Associações JMS . . . . .	116
Associações genéricas JMS . . . . .	125
Associações JMS do WebSphere MQ. . . . .	131
Associações do WebSphere MQ . . . . .	138
Limitações de associações . . . . .	148
Objectos de negócio . . . . .	149
Definir objectos de negócio . . . . .	150
Trabalhar com objectos de negócio . . . . .	151
Objectos de negócio especiais . . . . .	152
Modo de análise do objecto de negócio. . . . .	153
Considerações ao escolher o modo de análise do objecto de negócio . . . . .	154
Benefícios da utilização do modo de análise passivo versus activo . . . . .	155
Considerações sobre desenvolvimento e migração de aplicações . . . . .	155
Relações . . . . .	157
Serviço de relações . . . . .	159
Gestor de relações. . . . .	160
Relações em ambientes de Implementação de Rede . . . . .	160
APIs do serviço de relações. . . . .	161
O barramento de serviços empresariais no IBM Business Process Manager . . . . .	161
Ligar serviços através de um barramento de serviço de empresas . . . . .	161
Infra-estrutura de mensagens de barramento de serviços da empresa . . . . .	162
Sistema central de destino para o serviço de mensagens ou fila . . . . .	163
Fornecedores JDBC . . . . .	163

Barramento de integração de serviço para IBM Business Process Manager . . . . .	164	Processamento de falhas. . . . .	213
Aplicações e módulos de serviços . . . . .	164	De que forma são processadas as falhas nas associações de exportação . . . . .	213
Importações e associações de importação	165	Como são processadas falhas em associações de importação . . . . .	215
Exportações e associações de exportações	167	Interoperacionalidade entre módulos SCA e serviços Open SCA . . . . .	217
Módulos de mediação . . . . .	168	Tipos de associação . . . . .	220
Primitivas de mediação . . . . .	170	Seleccionar associações apropriadas . . . . .	220
Encaminhamento dinâmico. . . . .	175	Associações SCA . . . . .	221
Controlo da política de mediação dos pedidos de serviço . . . . .	176	Associações do serviço da Web . . . . .	222
Registo e Repositório de Serviços		Descrição geral de associações de serviço da Web . . . . .	222
WebSphere . . . . .	176	Propagação de cabeçalho SOAP . . . . .	223
WebSphere eXtreme Scale . . . . .	177	Propagação de cabeçalho de transporte	226
Clientes do Message Service . . . . .	178	Trabalhar com associações do serviço da web (JAX-WS) . . . . .	228
		Anexos em mensagens SOAP . . . . .	231
		Utilização de associação de estilo de documento WSDL com mensagens multi-parte . . . . .	244
<b>Capítulo 2. Saiba mais sobre os conceitos chave . . . . .</b>	<b>179</b>	Associações HTTP. . . . .	245
Criação de versões . . . . .	179	Descrição geral das associações HTTP . . . . .	246
Criação de versões das Process Applications . . . . .	179	Cabeçalhos HTTP . . . . .	247
Arquivo de versões de módulos e bibliotecas	180	Associações de dados HTTP . . . . .	251
Módulos e bibliotecas associados a Process Applications ou Toolkits. . . . .	181	Associações EJB . . . . .	253
Convenções de nomenclatura . . . . .	181	Associações de importação EJB . . . . .	253
Convenções de nomenclatura para implementações do servidor do Process Center. . . . .	182	Associações de exportação EJB . . . . .	255
Convenções de nomenclatura para implementações do Process Server . . . . .	185	Propriedades de associação EJB . . . . .	256
Associações adequadas à versão . . . . .	186	Associações EIS . . . . .	260
Chamada para execução dinâmica adequada à versão . . . . .	188	Descrição geral das associações EI . . . . .	260
Implementar Process Application com projectos e módulos Java. . . . .	188	Funções-chave de associações ao EIS . . . . .	261
Implementar Process Applications com regras empresariais e selectores. . . . .	189	Propriedades dinâmicas da Especificação de interacção JCA e Especificação de ligação JCA . . . . .	264
Objectos de configuração . . . . .	189	Clientes externos com associações EIS . . . . .	265
Arquitectura de implementação . . . . .	189	Associações JMS . . . . .	266
Células . . . . .	189	Descrição geral de associações JMS . . . . .	266
Servidores . . . . .	190	Integração JMS e adaptadores de recurso	267
Servidores autónomos . . . . .	190	Associações de importação e exportação JMS . . . . .	267
Conjuntos . . . . .	191	Cabeçalhos JMS . . . . .	269
Perfis . . . . .	191	Esquema de correlação de destino de resposta dinâmico temporário JMS . . . . .	270
Gestores de implementação. . . . .	193	Clientes externos . . . . .	271
Nós . . . . .	193	Resolução de problemas de associação JMS . . . . .	272
Nós geridos . . . . .	193	Excepções de processamento . . . . .	273
Nós não geridos . . . . .	193	Associações genéricas JMS . . . . .	274
Agentes de nós. . . . .	194	Descrição geral das associações JMS genérico . . . . .	274
Considerações de nome para perfis, nós, servidores, sistemas centrais e células . . . . .	194	Funções-chave de associações JMS Genérico . . . . .	276
BPMN 2.0 . . . . .	198	Cabeçalhos JMS genéricos . . . . .	278
Definições de Processo Empresarial (BPDs) . . . . .	200	Resolução de problemas Generic	
Associações . . . . .	201	Associações JMS . . . . .	278
Exportar e importar descrição geral de associação . . . . .	203	Excepções de processamento . . . . .	280
Exportar e importar configuração de associação	206	Associações JMS do WebSphere MQ. . . . .	280
Transformação de formato de dados em importações e exportações . . . . .	207	Descrição geral das associações JMS do WebSphere MQ. . . . .	281
Rotinas de tratamento de dados . . . . .	207		
Associações de dados. . . . .	209		
Selectores de função em associações de exportação . . . . .	211		

Funções-chave de associações ao WebSphere MQ JMS . . . . .	283	Considerações sobre desenvolvimento e migração de aplicações . . . . .	304
Cabeçalhos JMS . . . . .	284	Relações . . . . .	306
Clientes externos . . . . .	285	Serviço de relações . . . . .	308
Resolução de problemas WebSphere MQ		Gestor de relações . . . . .	309
Associações JMS . . . . .	286	Relações em ambientes de Implementação de Rede . . . . .	309
Excepções de processamento . . . . .	287	APIs do serviço de relações. . . . .	310
Associações do WebSphere MQ . . . . .	287	O barramento de serviços empresariais no IBM Business Process Manager . . . . .	310
Descrição geral das associações do WebSphere MQ. . . . .	288	Ligar serviços através de um barramento de serviço de empresas . . . . .	310
Funções-chave de uma associação WebSphere MQ. . . . .	290	Infra-estrutura de mensagens de barramento de serviços da empresa . . . . .	311
Cabeçalhos WebSphere MQ. . . . .	292	Sistema central de destino para o serviço de mensagens ou fila . . . . .	312
Acrescentar estatisticamente MQCIH numa associação WebSphere MQ. . . . .	294	Armazenamentos de dados. . . . .	312
Clientes externos . . . . .	295	Fornecedores JDBC . . . . .	312
Resolução de problemas das associações WebSphere MQ. . . . .	295	Barramento de integração de serviço para IBM Business Process Manager . . . . .	313
Excepções de processamento . . . . .	296	Aplicações e módulos de serviços . . . . .	313
Limitações de associações . . . . .	297	Importações e associações de importação . . . . .	314
Limitações da associação MQ . . . . .	297	Exportações e associações de exportações . . . . .	316
Limitações das associações JMS, MQ JMS e JMS genérico . . . . .	297	Módulos de mediação . . . . .	317
Objectos de negócio . . . . .	299	Primitivas de mediação . . . . .	319
Definir objectos de negócio. . . . .	299	Encaminhamento dinâmico. . . . .	324
Trabalhar com objectos de negócio . . . . .	300	Controlo da política de mediação dos pedidos de serviço . . . . .	325
Objectos de negócio especiais . . . . .	302	Registo e Repositório de Serviços WebSphere . . . . .	325
Modo de análise do objecto de negócio. . . . .	302	WebSphere eXtreme Scale . . . . .	326
Considerações ao escolher o modo de análise do objecto de negócio . . . . .	303	Clientes do Message Service . . . . .	327
Benefícios da utilização do modo de análise passivo versus activo. . . . .	304		





---

## Capítulo 1. Introdução ao IBM Business Process Manager

Conheça as capacidades fornecidas pelo IBM® Business Process Manager para gestão de processos empresariais e como as várias fases da gestão de processos empresariais, como criar e implementar Process Applications, se relacionam entre si.

A Process Application é o contentor fundamental para processos e respectivos componentes no IBM Business Process Manager. Os designers de processos criam aplicações de processamento em ambientes de autoria e poderão incluir serviços, tarefas e artefactos necessários para suportar a execução.

Os serviços de integração avançada são implementados no Integration Designer da IBM e associados a aplicações de processamento. A partir do Process Center, as Process Applications são implementadas no Process Server, o qual é o ambiente em tempo de execução do processo do IBM Business Process Manager.

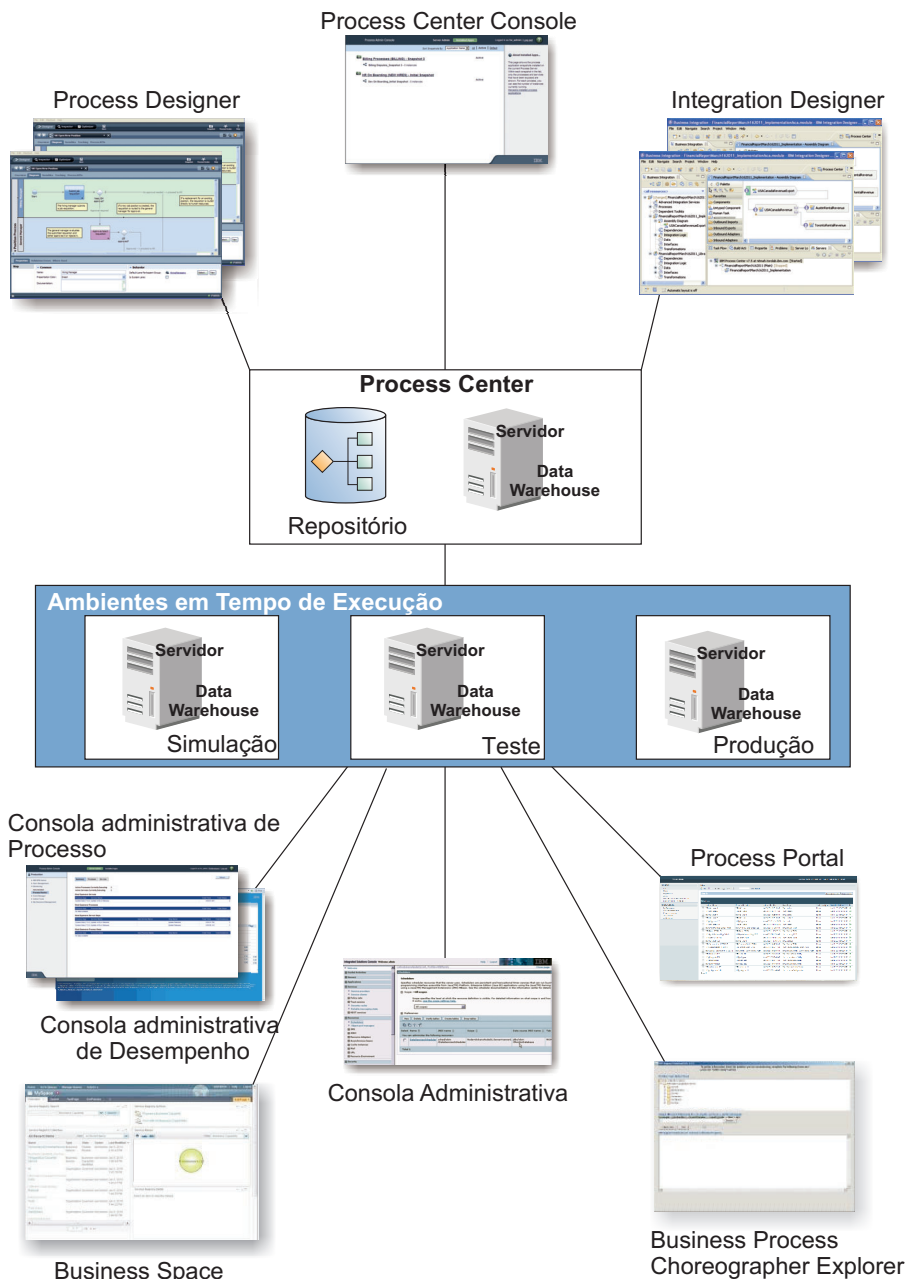
De forma semelhante, os processos automáticos criados no Integration Designer podem utilizar fluxos de actividades humanas desenvolvidos no IBM Process Designer.

---

### Descrição geral do produto

Os componentes do IBM Business Process Manager fornecem um repositório do BPM unificado; ferramentas para autores, administradores e utilizadores, bem como uma plataforma de tempo de execução. O produto pode ser configurado para suportar vários níveis de complexidade e envolvimento com gestão de processos empresariais.

O seguinte diagrama ilustra uma configuração típica do IBM Business Process Manager:



- A partir dos ambientes de autoria do IBM Process Designer e IBM Integration Designer, os programadores ligam ao IBM Process Center. A partir de quaisquer destas ferramentas de desenvolvimento de aplicação baseadas em GUI, os programadores podem criar, testar, depurar e implementar aplicações empresariais. Seleccione uma ferramenta ou a outra, dependendo do tipo de aplicação que está a desenvolver. Também pode haver casos em que utilizar ambas as ferramentas traz vantagens significativas.
- Nos ambientes de criação do Process Designer e do Integration Designer, os designers de processos e serviços podem criar Process Applications implementáveis e Toolkits reutilizáveis. As Process Applications contêm modelos de processos e implementações de serviços, incluindo os ficheiros de suporte que é requerido. As process applications estão armazenados no repositório Process Center para que possam ser partilhados.
- O Process Center inclui dois servidores, o servidor Process Center e o servidor Performance Data Warehouse. Estes servidores permitem aos programadores que trabalham no Process Designer executar as respectivas Process Applications e armazenar dados de desempenho para testes e reproduções

durante os esforços de desenvolvimento. O Performance Data Warehouse obtém os dados registados do servidor do Process Server ou Process Center em intervalos regulares.

- Process Center também suporta várias funções administrativas. A partir da Consola Process Center, os administradores instalam as Process Applications prontas para simulação, teste ou produção nos process servers. Os administradores também podem gerir instâncias em execução de process applications em ambientes configurados.
- Instale process applications num process server para simular, testar e produzir. Os ambientes em tempo de execução suportam processos BPMN (Business Process Model and Notation) 2.0. IBM Business Process Manager Advanced também suporta processos BPEL (Business Process Execution Language).
- A partir do Process Admin Console e do Performance Admin Console, os administradores podem gerir e manter todos os servidores de tempo de execução. Utilize a Consola de administração de processamento para gerir o servidor Process Center e process servers nos seus ambientes em tempo de execução. Utilize a Performance Admin Console para identificar estrangulamentos de desempenho e para capturar dados de instrumentação para uma futura análise.
- Utilize a consola administrativa para criar e gerir objectos tais como recursos, aplicações e servidores. Em adição, utilize a consola administrativa para ver mensagens de produto.
- Utilize espaço empresarial para criar espaço empresarial personalizado que fornece widgets para supervisionar ou administrar aspectos diferentes do sistema. Por exemplo, pode supervisionar actividades empresariais, serviços e condição de sistema ou administrar políticas de mediação e agendas empresariais. Também pode criar um espaço empresarial com o widgets Gestão de Tarefas Humanas e utilizá-lo para participar nos processos empresariais.
- Ao utilizar o Process Portal, os participantes no processo podem ligar-se ao servidor do Process Center ou Process Server em qualquer ambiente em tempo de execução configurado, se um processo esteja a ser desenvolvido, testado ou publicado num ambiente de produção.
- Gerir instâncias de processos BPEL (Business Process Execution Language) no Business Process Choreographer Explorer no Espaço empresarial.

## Configurações do IBM Business Process Manager

As diferentes configurações do IBM Business Process Manager estão correlacionadas com etapas e pontos de entrada típicos no programa de gestão de processos empresariais de uma empresa.

Tabela 1. Configurações do IBM Business Process Manager

Configuração	Fase
Advanced	<p><b>Transformação</b></p> <p>Conjunto completo de capacidades de gestão de processos empresariais</p> <ul style="list-style-type: none"> <li>• Suporte expandido para automatização de processos de elevado volume</li> <li>• Componentes SOA incorporados para integração e orquestração de serviços extensivos ao nível da empresa</li> </ul>
Standard	<p><b>Programa</b></p> <p>Configurado para projectos de gestão de processos empresariais típicos</p> <ul style="list-style-type: none"> <li>• Para programas de melhoria multi-projecto, com elevado envolvimento empresarial</li> <li>• Suporte de integração do sistema básico</li> <li>• Rentabilização rápida e produtividade do utilizador melhorada</li> </ul>
Express	<p><b>Projecto</b></p> <p>Configurado para o primeiro projecto de gestão de processos empresariais</p> <ul style="list-style-type: none"> <li>• Rentabilização rápida: produtividade do utilizador melhorada</li> <li>• Preço de entrada baixo</li> <li>• Fácil instalação e configuração</li> </ul>

## Capacidades de configuração do IBM Business Process Manager

Conheça os produtos e as capacidades oferecidos pela IBM para gestão de processos empresariais e escolha o mais adequado para a sua empresa.

O IBM Business Process Manager é uma plataforma única BPM que combina capacidades humanas e de integração num produto unificado. As diferentes configurações do produto estão disponíveis para diferentes utilizadores e satisfazem as diversas necessidades na empresa. As configurações do produto podem ser combinadas para ambientes de criação de colaboração e em tempo de execução implementados pela rede.

Tabela 2. Capacidades de configuração do IBM Business Process Manager

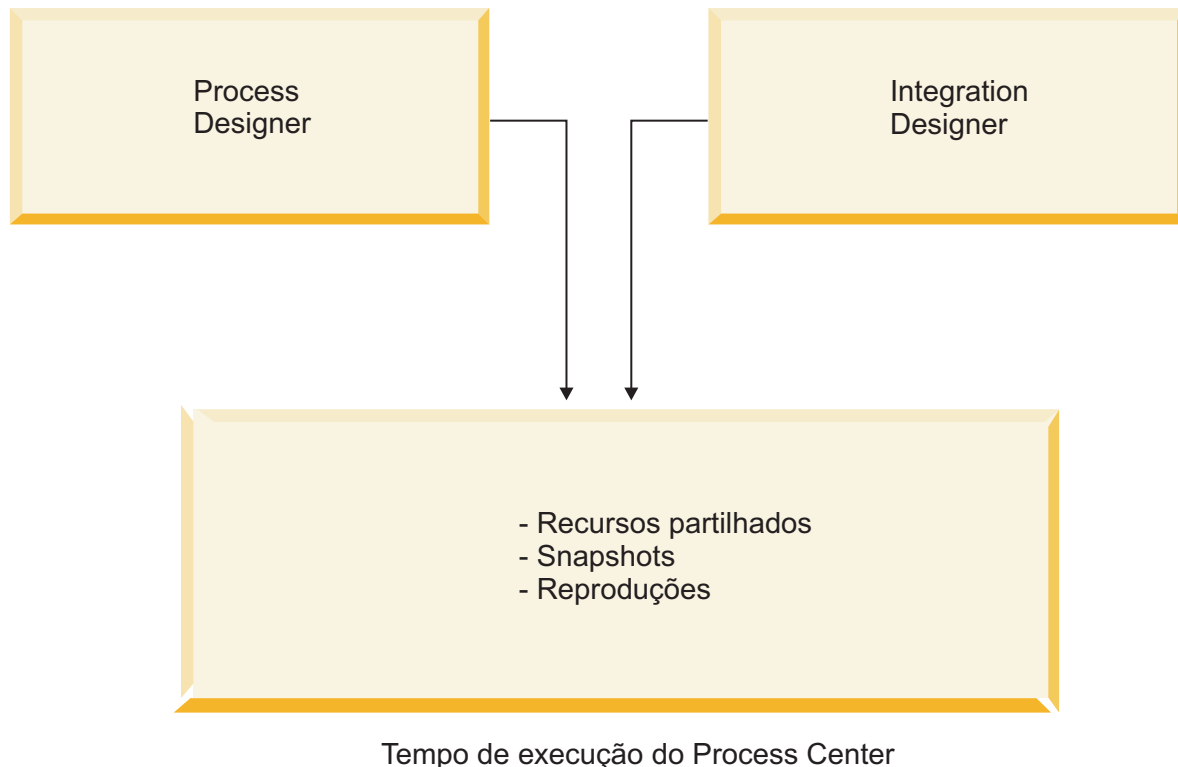
Função	Ava.	Pad.	Express
Execução compatível com o WebSphere Lombardi Edition	X	X	X
Process Designer (BPMN)	X	X	X
Edição de colaboração/reprodução imediata	X	X	X
Interfaces do utilizador "coach do processo" interactivas	X	X	X
Regras de processo baseadas em ODM	X	X	X
Process Portal	X	X	X
Monitorização e relatórios em tempo real	X	X	X
Análise e optimizador do desempenho	X	X	X
Performance Data Warehouse	X	X	X
Process Center/repositório de recursos partilhados	X	X	X
Autores e utilizadores finais de processos ilimitados	X	X	200 utilizadores/3 autores
Elevada disponibilidade: criação de conjuntos e núcleos ilimitados	X	X	<ul style="list-style-type: none"><li>• Produção de 4 núcleos</li><li>• Desenvolvimento de 2 núcleos</li><li>• Sem conjunto</li></ul>
Execução compatível com o WebSphere Process Server	X		
Integration Designer (BPEL/SOA)	X		
ESB (Enterprise Service Bus) incorporado	X		
Suporte de transacções	X		
Adaptadores de integração	X		
Interface do utilizador do Flexible Business Space	X		
Plataforma de suporte avançada (Linux on System z, IBM AIX, Solaris)	X	X	X

## Repositório do Process Center

O Process Center inclui um repositório para todos os processos, serviços e outros recursos criados nos ambientes de criação do IBM Business Process Manager, Process Designer e Integration Designer.

O Process Center é um componente de software que executa como um servidor onde o Process Designer e o Integration Designer partilham elementos, activamente deixando-os desenvolver processos de negócio cooperativamente de um modo altamente interactivo. Estes processos empresariais podem utilizar pontos de monitorização criados com o Business Monitor development toolkit. O resultado é um processo empresarial que pode ser examinado em tempo de execução para verificar a eficácia em condições de trabalho reais.

No diagrama que se segue, poderá ver vários componentes relacionados que permitem criar processos empresariais complexos.



A consola do Process Center fornece as ferramentas necessárias para manter o repositório.

- Na consola do Process Center é possível criar Process Applications e Toolkits, bem como conceder aos outros utilizadores acesso a essas Process Applications e Toolkits.
- Nos ambientes de criação, é possível criar modelos de processos, serviços e outros recursos nas Process Applications.
- Process Center inclui um servidor Process Center e o Performance Data Warehouse, permitindo aos utilizadores nos ambientes de criação executar processos e armazenar dados de desempenho para efeitos de teste e reprodução.
- Na Process Center Console, os administradores instalam as Process Applications prontas para teste ou produção nos servidores de processo nesses ambientes.
- Na Process Center Console, os administradores gerem as instâncias das Process Applications em execução nos ambientes configurados.

A Process Center Console fornece uma localização cómoda na qual é possível criar e manter contentores de elevado nível, como Process Applications e Toolkits. Os administradores que não trabalham activamente na vista do Designer podem utilizar a consola do Process Center para fornecer um contexto no qual os analistas e programadores do BPM podem criar os respectivos processos e implementações subjacentes. Outra tarefa principal para os administradores é a gestão do acesso ao repositório do Process Center através da configuração da autorização apropriada para os utilizadores e grupos.

Utilizadores com autorização adequada podem efectuar algumas tarefas administrativas directamente no Process Designer e Integration Designer. Por exemplo, um programador com acesso de escrita à Process Application que pretenda capturar o estado de todos os recursos do projecto numa etapa significativa de desenvolvimento pode criar um Snapshot enquanto trabalha na vista do Designer.

## Process Server e ambientes em tempo de execução

O Process Server fornece um único ambiente em tempo de execução do BPM que pode suportar um vasto número de processos empresariais, orquestração de serviços e capacidades de integração.

Nos ambientes de criação, o servidor de processos integrado no Process Center permite executar processos à medida que são criados. Quando estiver pronto, poderá instalar e executar esses mesmos processos nos servidores de processos nos ambientes em tempo de execução. O componente Business Performance Data Warehouse recolhe e agrega os dados dos processos em execução nos servidores de processos. É possível utilizar estes dados para melhorar os processos empresariais.

A Process Admin Console permite gerir os servidores de processos nos ambientes em tempo de execução, por exemplo, simulação, teste e produção, bem como o servidor de processos que faz parte do Process Center.

## Ambientes de criação

O IBM Business Process Manager Advanced oferece dois ambientes de criação. Utilize o IBM Process Designer para modelar processos empresariais de forma eficiente que envolvam tarefas humanas. Utilize o IBM Integration Designer para criar serviços auto-inclusos ou que chamem outros serviços existentes, como serviços da Web, aplicações de recursos empresariais ou aplicações executadas em CICS e IMS.

- “Process Designer”
- “Integration Designer” na página 7

## Process Designer

O Process Designer está disponível em todas as edições do produto. O IBM Business Process Manager Advanced também oferece o Integration Designer com os respectivos editores e adaptadores associados.

Um processo é a unidade de lógica principal no IBM Business Process Manager. É o contentor para todos os componentes de uma definição de processo, incluindo serviços, actividades e portas de ligação, temporizador, mensagem, eventos de excepção, linhas ordenadas, regras e variáveis. Quando se modela um processo, está-se a criar uma Definição de processo empresarial (BPD, Business Process Definition) reutilizável. Tanto o Process Designer como o Integration Designer podem criar modelos de processos que possam conter tarefas humanas.

O Process Designer ajuda-o a desenvolver processos empresariais. Com uma ferramenta orientada para gráficos fácil de utilizar, pode criar uma sequência de acções que compõem um processo empresarial e redesenhar esse processo ao longo do tempo à medida que as circunstâncias mudam. Se uma ou mais actividades requerem acesso a grandes sistemas backend ou serviços que forneçam dados para o processo empresarial, por exemplo, para obter informações sobre clientes, pode colmatar essa necessidade utilizando o Integration Designer. Utilizando uma interface simples, uma actividade no Process Designer pode chamar um serviço criado no Integration Designer. Esse serviço pode utilizar fluxos de mediação para transformar, encaminhar e melhorar dados e adaptadores para aceder a vários sistemas backend da forma padrão. Resumindo, o Process Designer concentra-se no processo empresarial e o Integration Designer concentra-se nos serviços automáticos para complementar o processo empresarial. Consulte Introdução ao IBM Process Designer.

Todos os projectos do Process Designer estão contidos na Process Application. Essas Process Applications e artefactos associados são armazenados no repositório do Process Center. As Process Applications podem partilhar recursos que tenham sido colocados em Toolkits.

O IBM Business Process Manager fornece várias interfaces de utilizador para permitir modelar, implementar, simular e inspecionar processos empresariais. Pode criar e gerir Process Applications,

Toolkits, controlos e Snapshots no Process Center Console. Pode criar modelos de processo, relatórios e serviços simples no Process Designer. Pode executar e depurar processos no Inspector. Também pode executar simulações no Optimizer.

As Process Applications desenvolvidas no Process Designer podem ser executadas em qualquer altura no servidor do Process Center ou guardadas num Snapshot e implementadas no Process Server. O mesmo acontece nos serviços desenvolvidos no Integration Designer e associados a Process Applications.

## **Integration Designer**

O Integration Designer fornece editores e ajudas para os programadores poderem criar serviços e processos automáticos complexos (tais como módulos SCAe processos BPEL). Está disponível como um componente do pacote do IBM Business Process Manager Advanced ou como um conjunto de ferramentas autónomo para outros utilizadores.

O IBM Integration Designer foi concebido como ambiente de desenvolvimento de integração completo para as aplicações integradas. As aplicações integradas não são simples. Podem chamar aplicações no Enterprise Information Systems (EIS), envolver processos empresariais em departamentos ou empresas e chamar aplicações local ou remotamente escritas em várias linguagens e executadas em diversos sistemas operativos. Os componentes são criados e montados noutras aplicações integradas (ou seja, aplicações criadas a partir de um conjunto de componentes) através de editores visuais. Os editores visuais apresentam uma camada de abstracção entre os componentes e as respectivas implementações. Um programador que utilize as ferramentas pode montar uma aplicação integrada sem conhecimentos detalhados da implementação subjacente de cada componente.

As ferramentas do Integration Designer são baseadas numa arquitectura orientada para os serviços. Os componentes são serviços e uma aplicação integrada que envolva muitos componentes também é um serviço. Os serviços criados estão em conformidade com as normas da indústria. Os processos BPEL, que também se tornam componentes, são criados de forma semelhante através de ferramentas visuais fáceis de utilizar em conformidade com o Business Process Execution Language norma da indústria.

No paradigma do Integration Designer, os componentes são montados em módulos. As importações e exportações são utilizadas para partilhar dados entre módulos. Os artefactos colocados numa biblioteca podem ser partilhados entre módulos.

Os módulos e as bibliotecas podem ser associados a uma Process Application para utilização com o Process Center e utilizados como serviços pelos processos criados no Process Designer. Nesses casos, também podem ser implementados na Process Application.

Em alternativa, os módulos e as bibliotecas podem ser implementados directamente no ambiente de teste ou no Process Server. Pode utilizar módulos de mediação para criar fluxos de mediação, os quais pode implementar no WebSphere Enterprise Service Bus ou no Process Server.

O IBM Integration Designer também fornece a capacidade de criar tipos de dados e mapas XML que podem ser implementados na aplicação WebSphere DataPower. Também se pode transferir ficheiros do e para o WebSphere DataPower.

## **Ferramentas de administração**

O IBM Business Process Manager inclui um conjunto de ferramentas de administração para o ajudar a completar tarefas desde a instalação e a gestão de snapshots até processos administrativos e trabalho com recursos no seu ambiente de TI.

### **Ferramentas da linha de comandos**

O IBM Business Process Manager disponibiliza ferramentas de linha de comandos, interfaces de script e interfaces de programação para administrar o ambiente em tempo de execução.



- As ferramentas da linha de comandos são programas simples que executam a partir de um pedido de confirmação do comando do sistema operativo para executar tarefas específicas. Ao utilizar estas ferramentas, pode iniciar e interromper servidores de aplicação, verificar o estado do servidor, adicionar ou remover nós e outras tarefas.
- O programa de script administrativo WebSphere (wsadmin) é um ambiente interpretador de comandos não-geográfico que lhe permite executar opções administrativas numa linguagem de script (Jython ou Jacl) e submeter programas de linguagem script para execução. Suporta as mesmas tarefas que a consola administrativa, assim como várias tarefas da consola do Process Center. A ferramenta wsadmin destina-se a ambientes de produção e operações não assistidas.
- As interfaces de programação administrativa são um conjunto de classes e métodos Java ao abrigo da especificação Java Management Extensions (JMX) que disponibilizam suporte para administrar a arquitectura de componente de serviço (SCA) e objectos de negócios. Cada interface de programação inclui uma descrição do seu objectivo, um exemplo que demonstra a forma de utilização da interface ou classe e referências a descrições do método individual.

### **Consola do Process Center**

A consola do Process Center disponibiliza uma localização conveniente para os utilizadores criarem e manterem itens de biblioteca de elevado nível, tal como aplicações e conjuntos de ferramentas de processamento. Ajuda ao fornecer uma estrutura na qual os analistas e os programadores do BPM podem criar os próprios processos e as implementações subjacentes. Além disso, a consola do Process Center disponibiliza ferramentas para fazer a manutenção do repositório, incluindo a definição da autorização apropriada para utilizadores e grupos.

Aceder à consola do Process Center através do navegador da Web (por exemplo, <http://host:9080/ProcessCenter>).

### **Consola de administração de processamento**

A consola de administração de processamento é utilizada para administrar os servidores de processamento no seu ambiente, incluindo os utilizadores e snapshots para cada servidor. Além disso, disponibiliza ferramentas para o ajudar a gerir filas e memórias cache.

A consola de administração de processamento inclui um Inspector de processamento, uma ferramenta para verificar e gerir as ocorrências do processo para aplicações de processamento executadas num Process Server específico.

Aceder à Process Admin Console através de um navegador da Web (por exemplo, <http://host:9080/ProcessAdmin>).

### **Business Performance Admin Console**

A Business Performance Admin Console inclui ferramentas para gerir os Performance Data Warehouses no ambiente. Pode utilizar esta ferramenta para gerir filas de servidor e monitorizar o desempenho do servidor.

Aceder à Business Performance Admin Console através de um navegador da Web (por exemplo, <http://host:9080/PerformanceAdmin>).

### **Consola administrativa do servidor da aplicação WebSphere**

A consola administrativa é utilizada para administrar aplicações, serviços e outros recursos no âmbito de uma célula, nó, servidor ou conjunto de unidades. Pode utilizar a consola com servidores autónomos e com gestores de implementação que gerem todos os servidores numa célula num ambiente de rede.

Se possui um perfil autónomo instalado, possui um nó único no seu domínio administrativos, conhecido como célula. Utilize a consola administrativa para gerir aplicações, barramentos, servidores e recursos no domínio administrativo. De forma semelhante, se possui uma célula de implementação de rede instalada e configurada, possui um nó de gestão de implementação ou mais nós geridos na mesma célula. Utilize a consola administrativa para gerir aplicações, definir nós geridos na célula e monitorizar e controlar esses nós e os seus recursos.



Aceder a esta consola através de um navegador da Web (por exemplo, <http://host:9060/ibm/console> ou <https://host:9043/ibm/console>).

### **Explorador coreógrafo de processo empresarial e explorador de arquivo de processo empresarial**

Dependendo da sua função de utilizador, pode utilizar estas interfaces de cliente para gerir processos BPEL e tarefas humanas criadas no IBM Integration Designer, trabalhar com as tarefas que lhe foram atribuídas, visualizar os processos BPEL terminados e tarefas humanas que estão na base de dados do arquivo ou eliminar processos e tarefas do arquivo.

### **Widgets de administração**

Widgets de administração oferece uma maneira de gerir e monitorizar certos componente da solução geral de gestão de processos empresariais, incluindo módulos e serviços de Advanced Integration Service. Utilize estes widgets num espaço empresarial para fornecer visibilidade nos módulos e aplicação de serviços e para responder a perguntas como estas:

- Que serviços são consumidos ou expostos por um módulo e quais são os tempos de resposta e produtividade num certo espaço de tempo para estes serviços?
- Qual é o estado de um módulo?
- Existem alguns eventos falhados no módulo?
- Que mediações de políticas estão associadas ao módulo?
- Que processos BPEL e tarefas humanas são utilizadas num módulo?
- Existem agendas empresariais ou regras de negócio no módulo?

Utilize um ou mais dos widgets para obter um snapshot da condição geral do sistema da solução empresarial, incluindo o estado da topologia (ambientes de implementação, conjuntos de unidades), aplicações de sistema (por exemplo, o Business Process Choreographer ou o gestor de eventos que falhou), origens de dados, motores de mensagens e filas de mensagens.

### **Gestor de regras de processo empresarial**

O gestor de regras de processo empresarial é uma ferramenta baseada na Web que ajuda o analista de negócios a procurar e modificar os valores de regra de negócios. A ferramenta é uma opção do servidor de processo da IBM que pode seleccionar para instalar o tempo de criação do perfil após instalar o servidor.

## **Acessibilidade no IBM Business Process Manager**

As funções de acessibilidade ajudam os utilizadores com deficiências físicas, como mobilidade restrita ou visão limitada, a utilizar produtos de tecnologias de informação com êxito.

A IBM esforça-se por fornecer produtos acessíveis a todas as pessoas, independentemente da idade ou da capacidade.

Para obter mais detalhes sobre funções de acessibilidade para este produto, consulte Funções de acessibilidade no IBM Business Process Manager.

## **Disponibilidade de idiomas nacionais no IBM Business Process Manager**

O IBM Business Process Manager suporta os idiomas que se seguem. A documentação pode não estar totalmente traduzida.

- Chinês simplificado
- Chinês tradicional
- Checo
- Inglês (EUA)
- Francês
- Alemão

- Húngaro
- Italiano
- Japonês
- Coreano
- Polaco
- Português do Brasil
- Russo
- Espanhol

O IBM Business Process Manager fornece suporte parcial dos idiomas que se seguem. A documentação pode não estar totalmente traduzida.

- Árabe (traduzido para widgets de tarefas humanas do BPEL, widgets do Business Process Choreographer Explorer, framework do Business Space e widgets do Space Monitor)
- Dinamarquês (traduzido para widgets do Business Space Monitor e framework do Business Space)
- Neerlandês (traduzido para framework do Process Designer, Process Center, BPM Modeler, Service Modeler, JSEditor, Process Designer e Business Space)
- Finlandês (traduzido para widgets de monitorização do Business Space, framework do Business Space, BPD Modeler, Service Modeler, JSEditor e Process Designer)
- Grego (traduzido para Process Designer, Process Center e Business Space)
- Hebraico (traduzido para widgets de tarefas humanas do BPEL, Business Process Choreographer Explorer, framework do Business Space e mecanismos do Business Space Monitor)
- Norueguês (traduzido para widgets do Business Space Monitor e framework do Business Space)
- Português-Portugal (Process Designer, Process Center, BPD Modeler, Service Modeler e JSEditor)
- Romeno (traduzido para operações em tempo de execução)
- Eslovaco (traduzido para Business Space, BPD Modeler, Service Modeler, JSEditor e Process Designer)
- Sueco (traduzido para widgets do Business Space Monitor e framework do Business Space)
- Turco (traduzido para Business Space)

**Nota:** Para o locale turco, deve definir a entrada **case-insensitive-security-cache** no ficheiro `60Database.xml` para **false** para permitir que os utilizadores e as palavras-passe contenham a letra *i*. O ficheiro `60Database.xml` está localizado no directório `install_root\profiles\profile_name\config\cells\cell_name\nodes\node_name\servers\server_name\process-center\config\system\`.

O IBM Business Process Manager fornece suporte para ser possível aos utilizadores introduzir cadeias bidireccionais no ambiente do Process Designer, coaches e Process Portal. Fornece APIs de JavaScript para manipulação de testes de linguagem bidireccionais.

Os Coaches e o Process Portal suportam a utilização de calendários hebraicos e árabes.

---

## Descrição geral da gestão de processos empresariais

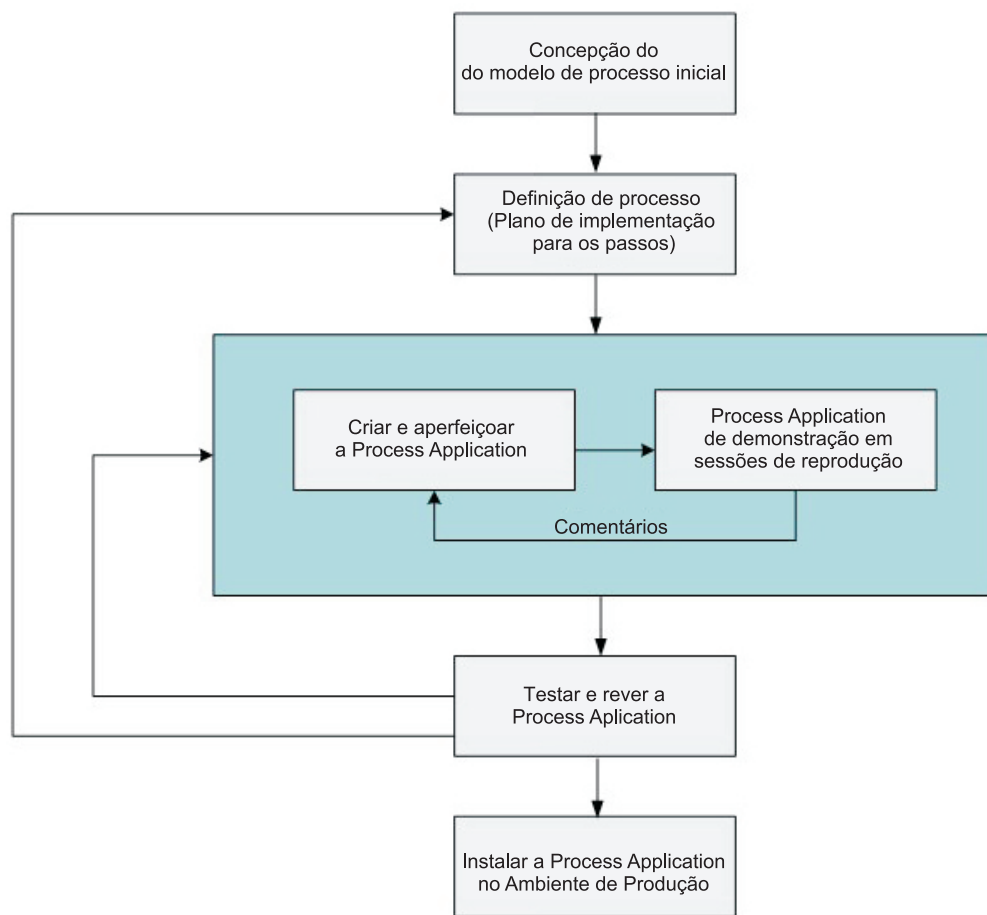
Quando desenvolver processos no Process Designer, é necessário planear a eventual instalação das Process Applications nos servidores nos ambientes de teste e de produção.

Process Designer pode ser encontrado em IBM Business Process Manager Express, IBM Business Process Manager Standard e IBM Business Process Manager Advanced. Nesta secção vamos concentrar-nos na versão Avançada, que é concebido para automatização de elevado volume e utilização de serviços complexos devolvidos em Integration Designer. Tem componentes SOA incorporados que podem ser utilizados para a integração de serviços extensivos ao nível da empresa. A versão Padrão pode ser utilizado por muitos profissionais de empresas cooperativamente para desenvolver múltiplos processos

sofisticados. Tem integração de sistema básica. A versão Express é para um número de utilizadores reduzido num único servidor que estão a ser introduzidos nos processos empresariais ou não necessitam de acesso a muitos sistemas externos.

O diagrama seguinte apresenta o ciclo de vida de um esforço de desenvolvimento de processos típico. Inclui passos para criar e aperfeiçoar um serviço de instalação para que possa instalar as Process Applications no ambiente de produção.

Como mostra este diagrama, pode trabalhar exclusivamente no ambiente de desenvolvimento. É necessário configurar Process Servers para os ambientes de teste e produção.



## Descrição geral do processo de modelação

Um processo é uma grande unidade de lógica no IBM Business Process Manager. É o contendor para todos os componentes de uma definição de processo, incluindo serviços, actividades e portas de ligação, temporizador, mensagem, eventos de excepção, linhas ordenadas, regras e variáveis. Quando se modela um processo, está-se a criar uma Definição de processo empresarial (BPD, Business Process Definition) reutilizável.

Componentes de processo permitem que defina o processo de fluxo de trabalho para utilizadores finais, criando lógica dentro de um processo e integração com outras aplicações e origem de dados. Para compreender o que ocorre dentro de um processo em tempo de execução, é importante compreender os componentes que constituem o processo em tempo de concepção.

## **Construir processos no IBM BPM**

Muitas pessoas diferentes de diversas organizações estão normalmente envolvidas no desenvolvimento de processos através do IBM BPM. A maior preocupação é garantir que se está a construir a melhor solução possível para reunir os objectivos do seu projecto. Para garantir resultados de sucesso, os membros da equipa devem trabalhar juntos para capturar requisitos de processo e frequentemente desenvolver o modelo e a sua implementação.

## **Reutilizar itens no Process Designer**

Process Designer permite aos programadores do processo reutilizar itens existentes tanto dentro como para lá das Process Applications. Por exemplo, se souber que já existem vários serviços, isso inclui Coaches e outros itens partilhados que você e outros programadores precisam, pode aceder e reutilizar esses itens ao concluí-los num Toolkit. Depois, a partir da sua Process Application, pode adicionar uma dependência ao Toolkit no qual os itens partilhados residem. Isto permite que escolha um dos serviços existentes ao escolher a implementação para uma actividade. Os itens no Toolkit também podem ser utilizados por outros programadores que trabalham em Process Applications diferentes.

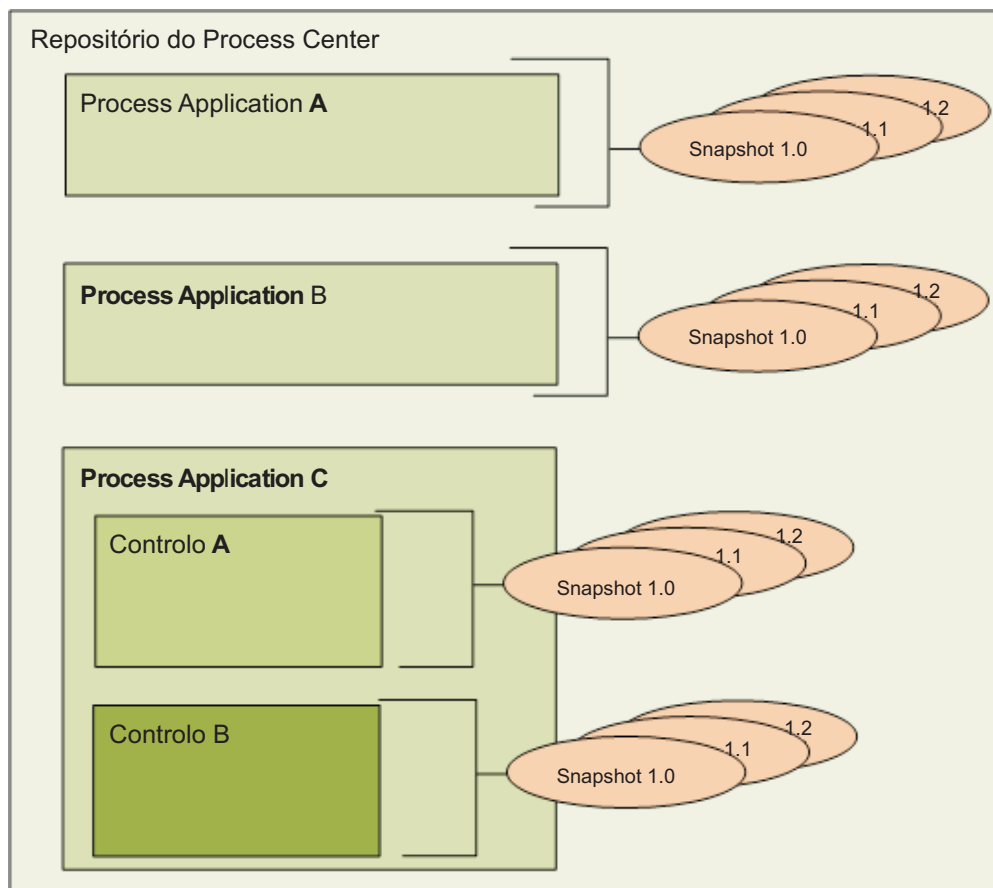
## **Utilizar o Designer no IBM Process Designer**

A interface do Designer fornece as ferramentas necessárias que precisa para modelar o seu processo no IBM BPM.

## **Desenvolvimento de processos com o Process Center**

O IBM Process Center serve de repositório central para todos os elementos do activo, de projectos criados no Process Designer. Quando vários clientes Process Designer se ligam ao Process Center, os utilizadores podem partilhar itens, assim como, processos e serviços, e podem ainda visualizar mudanças realizadas por outros utilizadores, à medida que vão surgindo. O Process Center também pode ser utilizado como repositório para elementos do activo criados em IBM Integration Designer.

Quando desenvolver processos no Process Designer, há uma hierarquia disponível no repositório do Process Center que é concebido para ajudá-lo a criar os seus projectos. A figura seguinte fornece uma descrição geral do conceito da hierarquia do repositório



Como poderá ver a partir do diagrama de procedimentos, Process Center o repositório inclui os seguintes artefactos:

Tipo de conteúdo	Descrição
Process Applications	Contentores para os módulos de processo e implementações de suporte que os analistas e os programadores de BPM criam no Designer no IBM Process Designer.
Controlos	Subdivisões opcionais numa Process Application, baseada em versões das tarefas de equipa ou da Process Application. Quando activadas, as pistas permitem que ocorra desenvolvimento paralelo. Os administradores determinam se são necessárias pistas adicionais e, como tal, activadas para cada Process Application.
Snapshots	Gravam o estado dos itens numa Process Application ou controlo num determinado momento. Geralmente, os Snapshots representam um marco ou são utilizados para reproduções ou para instalações. É possível comparar Snapshots e reverter os Snapshots anteriores. Caso um administrador activar o rastreio para uma Process Application, será utilizado um Snapshot como base para um novo rastreio.

## Process Applications: descrição geral

Uma Process Application é um contentor para modelos de processos e as suas implementações de suporte; é guardada no repositório. Depois dos artefactos terem sido compostos ou criados de outra forma, eles são agregados numa Process Application.

As Process Applications contêm alguns, ou todos, dos seguintes artefactos:

- Um ou mais modelos de processo, também chamados Business Process Definitions (BPDs)
- Referências a toolkits

- Os serviços necessários para implementar actividades ou integrar com outros servidores, incluindo os Advanced Integration Services
- Um ou mais controlos
- Módulos e bibliotecas da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) (compostos no IBM Integration Designer)
- Um modelo do IBM Business Monitor para monitorizar o desempenho empresarial
- Quaisquer outros itens necessários para executar o processo

Para visualizar um vídeo introdutório sobre o desenvolvimento iterativo de process applications e toolkits com sugestões, snapshots e controlos, visualize o “Desenvolvimento iterativo de Process Applications e Toolkits”, disponível no YouTube ou no IBM Education Assistant information center. Uma transcrição do vídeo disponível.

## **Sugestões, snapshots e sugestões da Process application**

Quaisquer alterações são gravadas de forma dinâmica no repositório do Process Center, na sugestão que é a versão em funcionamento actualmente da process application. Pode utilizar sessões de reprodução na sugestão para testar instantaneamente e gerir a versão funcional actual da Process Application.

A process application permanece ao nível da sugestão até decidir criar um snapshot, que regista o estado de itens de biblioteca numa process application ou controlo num determinado momento. Tipicamente, tira um snapshot cada vez que pretende testar a integração ou pretende instalar a num process center server or um process server para testar, simular ou produzir.

**Nota:** A sugestão é um Snapshot especial; é o único tipo de Snapshot em que se pode alterar os conteúdos, mas só pode executado no Process Center Server. Não pode instalar uma sugestão num process server.

Por predefinição, cada process application tem um único controlo, chamado Main (Principal). Se pretende permitir o desenvolvimento em paralelo numa process application, pode criar pistas adicionais. Estas subdivisões opcionais na process application mantenha as alterações isoladas. Por exemplo, imagine que a sua empresa está em processo de renovar; durante esta transição, a process applications actual tem de ser mantida enquanto novas versões estão a ser desenvolvidas baseadas na entidade empresarial actualizada. Nesta situação, uma equipa pode estar a executar pequenas correcções na versão actual de uma process application (na pista Principal) enquanto outra equipa está a construir uma nova versão da process application numa pista diferente.

## **Toolkits para process applications**

Toolkits são contentores que armazenam itens da biblioteca (por exemplo, BPDs) para reutilização pelas process applications ou outros toolkits. Process applications podem partilhar itens da biblioteca de um ou mais toolkits, e toolkits podem partilhar itens de biblioteca de outros toolkits. Se tem acesso a um toolkit, pode criar uma dependência no mesmo e utilizar os itens da biblioteca desse toolkit na process application.

## **Process Applications e aplicações de nível empresarial**

Uma Process Application tem uma aplicação de nível empresarial (BLA - business level application), que age como um contentor para a Process Application e os seus elementos do activo (os elementos do activo incluem coisas como modelos de supervisão, módulos SCA, toolkits e bibliotecas). Cada snapshot de process application tem o seu próprio BLA. Muitas das tarefas de administração para um Snapshot (por exemplo, pará-lo ou iniciá-lo num servidor de produção) são efectuadas ao nível da BLA, permitindo uma gestão mais rápida e simples do Snapshot e todos os seus elementos do activo.

## Desenvolvimento iterativo de process applications e de toolkits

Tabela 3. Introdução

Cena	Áudio	Acção no ecrã
1	Bem-vindo à introdução de desenvolvimento iterativo de process applications e de toolkits no IBM Business Process Manager.	O ecrã de boas-vindas mostra o título do vídeo, <i>Desenvolvimento iterativo de process applications e de toolkits</i> e o subtítulo <i>Saiba mais sobre como utilizar os controlos e os snapshots superiores</i> . Copyright 2013, IBM Corporation.
2	Neste vídeo, irá saber mais sobre as process applications e os toolkits e sobre como pode utilizar as sugestões, os snapshots e os controlos para gerir o respectivo ambiente de desenvolvimento.	É apresentada uma lista marcada dos tópicos referidos no vídeo.
3	Os processos evoluem com o passar do tempo e alteram-se as necessidades da sua empresa. À medida que as suas process applications e toolkits passam pelo ciclo de desenvolvimento, de teste e de produção, pode gerir os recursos e as versões dos processos no IBM Business Process Manager.	É apresentado um diagrama do ciclo de concepção iterativo. As setas movem-se através do diagrama para apresentar a natureza cíclica do processo de concepção.
4	No IBM Business Process Manager, as process applications servem de contentores para os modelos de processo e suportam as implementações que os programadores criam no IBM Process Designer.	É apresentado o separador Process Apps na consola do Process Center. O cursor sobrepõe-se ao nome de uma process application e, em seguida, sobrepõe-se uma ligação que o utilizador pode utilizar para a abrir no IBM Process Designer. É apresentado o texto no ecrã "Process application: contentor instalável para o modelo de processo".
5	Os toolkits são contentores que mantêm os mesmos artefactos que as process applications. Contudo, ao contrário das process applications, estes não podem ser instalados e serem executados no Process Server. Os toolkits mantêm artefactos que podem ser reutilizados por uma ou mais process applications. Os toolkits são, em seguida, instalados indirectamente com cada process application que refira o mesmo.	É apresentado o separador Toolkits na consola do Process Center. O cursor sobrepõe-se ao nome de uma toolkit e, em seguida, sobrepõe-se uma ligação que o utilizador pode utilizar para a abrir no IBM Process Designer. É apresentado o texto no ecrã "Toolkit: contentor para artefactos reutilizáveis".

Tabela 4. Sugestões, snapshots e controlos

Cena	Áudio	Acção no ecrã
6	O IBM Business Process Manager suporta uma abordagem iterativa relativamente ao desenvolvimento de process applications e de toolkits com funcionalidades como: <ul style="list-style-type: none"> <li>• A sugestão que é a versão funcional actual</li> <li>• Snapshots, que registam o estado dos itens da biblioteca até um certo momento no tempo</li> <li>• Controlos, que são ramificações operacionais numa process application ou num toolkit quem podem ser utilizadas para desenvolver, de forma concorrente, múltiplas versões</li> </ul>	É apresentada uma lista marcada de funcionalidades incluídas no IBM BPM que suportam uma abordagem iterativa ao desenvolvimento de process applications e de toolkits.



Tabela 4. Sugestões, snapshots e controlos (continuação)

Cena	Áudio	Ação no ecrã
7	Quando desenvolver uma process application ou um toolkit no IBM Process Designer, quaisquer alterações que efectue são guardadas no repositório do Process Center. Esta versão funcional actual da process application ou do toolkit é denominada de sugestão.	É apresentado um diagrama no IBM Process Designer. Abre-se um componente de diagrama e a palavra <i>dados</i> é alterada para a palavra <i>data</i> num Coach. As alterações são, em seguida, guardadas. É apresentado o texto no ecrã “Sugestão: versão funcional actual”.
8	Pode utilizar sessões de reprodução na sugestão para testar instantaneamente a versão funcional actual da Process Application ou do Toolkit. Só pode executar a sugestão no servidor do Process Center e não pode instalar uma sugestão no Process Server.	Abre-se a página superior do diagrama de processo e é iniciada a sessão de reprodução. Isto abre a interface do Inspector e apresenta o Coach alterado numa janela do navegador da Web. É apresentado o texto no ecrã “Sugestão: reprodução para o teste instantâneo”.
9	Quando tiver concluído as alterações na process application ou no toolkit, pode criar um snapshot dos mesmos. O snapshot regista o estado de todos os itens na process application ou no toolkit num momento específico do tempo. Após criar um snapshot, os seus processos e os itens relacionados continuam disponíveis para serem editados na sugestão. No Process Designer, os snapshots disponíveis são apresentados sob Histórico da revisão. Apesar de poder visualizar estes snapshots, os mesmos não são editáveis.	Fecha-se a janela do navegador e é criado um snapshot da process application. É apresentada a interface do Designer e o cursor sobrepõe-se ao nome do novo snapshot sob a secção Histórico de revisão. É apresentado o texto no ecrã “Snapshot: regista o estado de todos os itens”.
10	Também pode comparar snapshots para visualizar as informações, tal como a hora em que foram criados e se existem processos adicionados aos mesmos.	É apresentada a vista de comparação de snapshots. O cursor sobrepõe-se à data do snapshot que foi criada e o nome de um item no processo que foi alterado. É apresentado o texto no ecrã “Comparar diferenças entre snapshots”.
11	Quando cria um novo de um toolkit, as process applications que são referidas nesse toolkit irão continuar a utilizar o snapshot antigo. Quando abre a process application no IBM Process Designer para editar a mesma, é apresentado um aviso que o notifica que está disponível uma nova versão do toolkit.	Abre-se a consola do Process Center. É apresentada a página Toolkits e abre-se um toolkit. É criado um novo snapshot do toolkit. Abre-se a página Process Apps e abre-se uma process application no Process Designer. O cursor sobrepõe-se sobre o aviso que surge e que é relacionado com os itens do menu. É apresentado o texto no ecrã “As Process Applications mantêm as referências aos snapshots de toolkits antigos”.
12	Quando a process application ou o toolkit está pronto a ser utilizado, um administrador pode instalar um ou mais snapshots dos mesmos no Process Server através da consola do Process Center. As process applications instaladas podem fazer referências a snapshots dos toolkits e não à versão funcional actual de um toolkit.	Abre-se a consola do Process Center e, em seguida, abre-se uma process application. O cursor sobrepõe-se a uma ligação num snapshot no qual o administrador pode fazer clique para instalar o snapshot. É apresentado o texto no ecrã “Instalar snapshots de process applications no Process Server”.
13	Os administradores também podem gerir snapshots a partir da consola do Process Center, tal como criar, arquivar e exportar snapshots.	Abre-se o menu pendente ao lado do nome do snapshot e o cursor sobrepõe-se a cada opção no menu. É apresentado o texto “Gerir snapshots através da consola do Process Center”.



Tabela 4. Sugestões, snapshots e controles (continuação)

Cena	Áudio	Ação no ecrã
14	Por predefinição, cada process application ou toolkit tem um único controlo denominado Principal (Main). Os administradores podem activar a criação de controlos adicionais para permitir o desenvolvimento paralelo. Estas subdivisões adicionais mantêm as alterações isoladas.	Abre-se a página Gerir da process application. É verificada a opção “Permitir que os utilizadores criem controlos nesta process app”. É apresentado o texto no ecrã “Controlos: subdivisões opcionais que mantêm as alterações isoladas”.
15	Crie um novo controlo a partir de um snapshot. Isto copia o snapshot seleccionado para o novo controlo. Os outros snapshots do controlo de origem não são copiados para o novo controlo.	Abre-se a página Snapshots da process application. É criado um novo controlo denominado Controlo 2 (Track 2). Abre-se o novo controlo e o cursor sobrepõe-se ao nome do snapshot copiado. É apresentado o texto no ecrã “Criar um controlo a partir de um snapshot”.
16	Cada controlo tem uma sugestão ou uma versão funcional distinta. Numa process application ou num toolkit com múltiplos controlos, pode copiar recursos a partir de um snapshot num controlo para a sugestão de outro controlo. Assim que o fizer, os recursos que foram revistos substituem os recursos que já estão no controlo de destino e são adicionados os novos recursos.	Abre-se o novo snapshot e é apresentada uma página com as definições e os itens da process application. É apresentada da definição do processo empresarial. Os recursos são copiados para a sugestão do controlo Principal (Main). Em seguida, são apresentadas as páginas onde aparecem as listas com os recursos novos, actualizados e em conflito. É apresentado o texto no ecrã “Copiar recursos para outros controlos”.
17	A hierarquia dos controlos e dos snapshots da process application no repositório do Process Center é apresentada aqui. Neste diagrama, as process applications A e B têm um controlo único predefinido, múltiplos snapshots e uma única sugestão. A Process Application C tem dois controlos, cada um com múltiplos snapshots e uma sugestão. Os snapshots e as sugestões em cada controlo são independentes de outros.	É apresentado um diagrama que mostra a relação entre as sugestões, os snapshots e os controlos. O diagrama mostra as representações das seguintes process applications: <ul style="list-style-type: none"> <li>• Process Application A, que tem um controlo, quatro snapshots e uma sugestão</li> <li>• Process Application B, que tem um controlo, três snapshots e uma sugestão</li> <li>• Process Application C, que tem dois controlos. Um controlo tem três snapshots e uma sugestão e outro controlo tem dois snapshots e uma sugestão.</li> </ul>

Tabela 5. Cenário da utilização de sugestões, snapshots e controlos

Cena	Áudio	Ação no ecrã
18	Agora que já sabe o que são as process applications e os toolkits e como pode utilizar as sugestões, os snapshots e os controlos, vamos verificar quando podem ser utilizados. Iremos acompanhar o Robert, um programador empresarial para a empresa ABC.	É apresentada uma fotografia com um homem a sorrir. A fotografia tem a etiqueta “Robert, programador empresarial, Empresa ABC”.

Tabela 5. Cenário da utilização de sugestões, snapshots e controlos (continuação)

Cena	Áudio	Acção no ecrã
19	<p>O Robert estava a desenvolver um processo que modela a forma como a empresa contrata colaboradores. O seu processo tem um controlo predefinido, Principal (Main). O Robert efectua as alterações ao processo na sugestão. Para se certificar de que o processo é executado sem problemas, ele inicia uma sessão de reprodução, que executa a sugestão no servidor do Process Center.</p>	<p>É apresentado um diagrama que representa o ambiente de desenvolvimento de uma process application.</p> <p>Sob a secção denominada Desenvolvimento, é apresentado um conjunto com três rectângulos imbricados. O rectângulo externo representa o repositório do Process Center, o rectângulo do meio representa a process application e o rectângulo interno representa o controlo Principal (Main).</p> <p>À direita da secção Desenvolvimento, existe uma secção denominada Teste. Esta secção contém um único rectângulo que representa o servidor do Process Center.</p> <p>Sob a secção Teste existe uma secção denominada Produção. Esta secção contém um único rectângulo que representa o Process Server.</p> <p>Para mostrar o Robert a efectuar as alterações à sugestão, é apresentado um triângulo que representa a sugestão no controlo Principal (Main). Quando o Robert executar uma sessão de reprodução, é apresentada uma seta que representa a sugestão a ser copiada para o servidor do Process Center.</p>
20	<p>O Robert e a sua equipa terminaram o desenvolvimento e o teste da process application e esta está pronta para ser implementada no ambiente de produção. Para a implementar, o Robert cria um snapshot da process application. Em seguida, a Alice, a administradora, instala o snapshot no Process Server. A Empresa ABC pode, agora, utilizar o processo do Robert para guiar a contratação de novos colaboradores.</p>	<p>Para mostrar o Robert a criar um snapshot, é apresentado círculo que representa um snapshot ao lado da sugestão no controlo Principal (Main). Para mostrar a Alice a instalar o snapshot, é apresentada uma seta que representa o snapshot a ser copiado para o Process Server.</p>
21	<p>Mais tarde, a Empresa ABC anuncia que irá ser renovada e que, no futuro, será conhecida como Empresa DEF. Durante a transição, o processo de contratação que o Robert criou deve ser mantido enquanto é desenvolvida uma nova versão com base na identidade empresarial actualizada. A equipa de manutenção efectua correcções menores ao processo original no controlo Principal (Main) enquanto o Robert, e a sua equipa, criam uma versão renovada aplicação de contratação no controlo Renovação da marca DEF.</p>	<p>Surge um snapshot adicional para mostrar que o desenvolvimento foi efectuado nesta process application. É adicionado um rectângulo que representa o controlo Renovação da marca DEF dentro da process application. O controlo Renovação da marca DEF tem, inicialmente, um snapshot e uma sugestão e, em seguida, é adicionado outro snapshot para mostrar o desenvolvimento no controlo.</p>

Tabela 5. Cenário da utilização de sugestões, snapshots e controlos (continuação)

Cena	Áudio	Acção no ecrã
22	Tanto a equipa de manutenção como a de renovação da marca podem executar, em separado, sessões de reprodução no servidor do Process Center. Quando as process applications estão prontas a serem implementadas, cada controlo pode instalar um ou mais snapshots no Process Server.	As setas mostra a direcção das sugestões de ambos os controlos a serem executados no servidor do Process Center ao mesmo tempo. Em seguida, as setas mostra os snapshots de cada controlo a serem executados no Process Server ao mesmo tempo.

Tabela 6. Conclusão

Cena	Áudio	Acção no ecrã
23	Neste vídeo, abordamos as process applications e os toolkits e a forma como o IBM Business Process Manager suporta o desenvolvimento de processos iterativos com sugestões, snapshots e controlos.	É apresentada uma lista marcada dos tópicos referidos no vídeo.
24	Para obter mais informações sobre como o IBM Business Process Manager utiliza sugestões, snapshots e controlos, procure os nossos vídeos YouTube ou visite estes recursos oficiais.	É apresentada uma lista marcada dos seguintes recursos adicionais: <ul style="list-style-type: none"> <li>• Canal do YouTube WebSphereEducation</li> <li>• Documentação do IBM Business Process Manager V8.5</li> <li>• IBM developerWorks</li> <li>• IBM Education Assistant</li> </ul>

## Executar e depurar processos

Ao utilizar o Inspector, os programadores individuais podem executar processos e serviços no servidor Process Center ou em tempo de execução do Process Server remoto.

O Inspector de IBM Process Designer é essencial para uma abordagem interactiva do processo de desenvolvimento. Uma equipa de desenvolvimento pode utilizar o Inspector para demonstrar implementação e concepção do processo actual em sessões de reprodução. Estas sessões ajudam a capturar informação importante das diferentes partes interessadas num processo, tais como gestão, utilizador final e analistas financeiros. Ao tomar uma abordagem iterativa do processo de desenvolvimento, garante que as Process Applications atingem os objectivos e cobrem as necessidades de todos os envolvidos.

O Inspector de IBM Process Designer inclui várias ferramentas que permitem completar tarefas, como as que se seguem, em cada um dos ambientes configurados:

Tarefa	Descrição
Gerir ocorrências de processos	Quando executa um processo, pode visualizar todas as ocorrências previamente executadas e em execução actualmente nos servidores do IBM Business Process Manager no respectivo ambiente. É possível gerir as ocorrências, por exemplo, parando-as e retomando-as mais tarde. Também pode gerir ocorrências previamente executadas ao filtrar ou eliminar relatórios específicos.

Tarefa	Descrição
Avançar e depurar um processo	Para uma ocorrência seleccionada, procure o passo que está a ser executado e avance no processo, avaliando a execução do processo passo-a-passo. Uma apresentação em árvore do processo em combinação com os indicadores chamados símbolo no diagrama do mesmo, ajudam a perceber em que passo do processo se encontra. Além disso, também tem a vantagem de verificar as variáveis usadas em cada passo e os valores correspondentes (onde for aplicável).

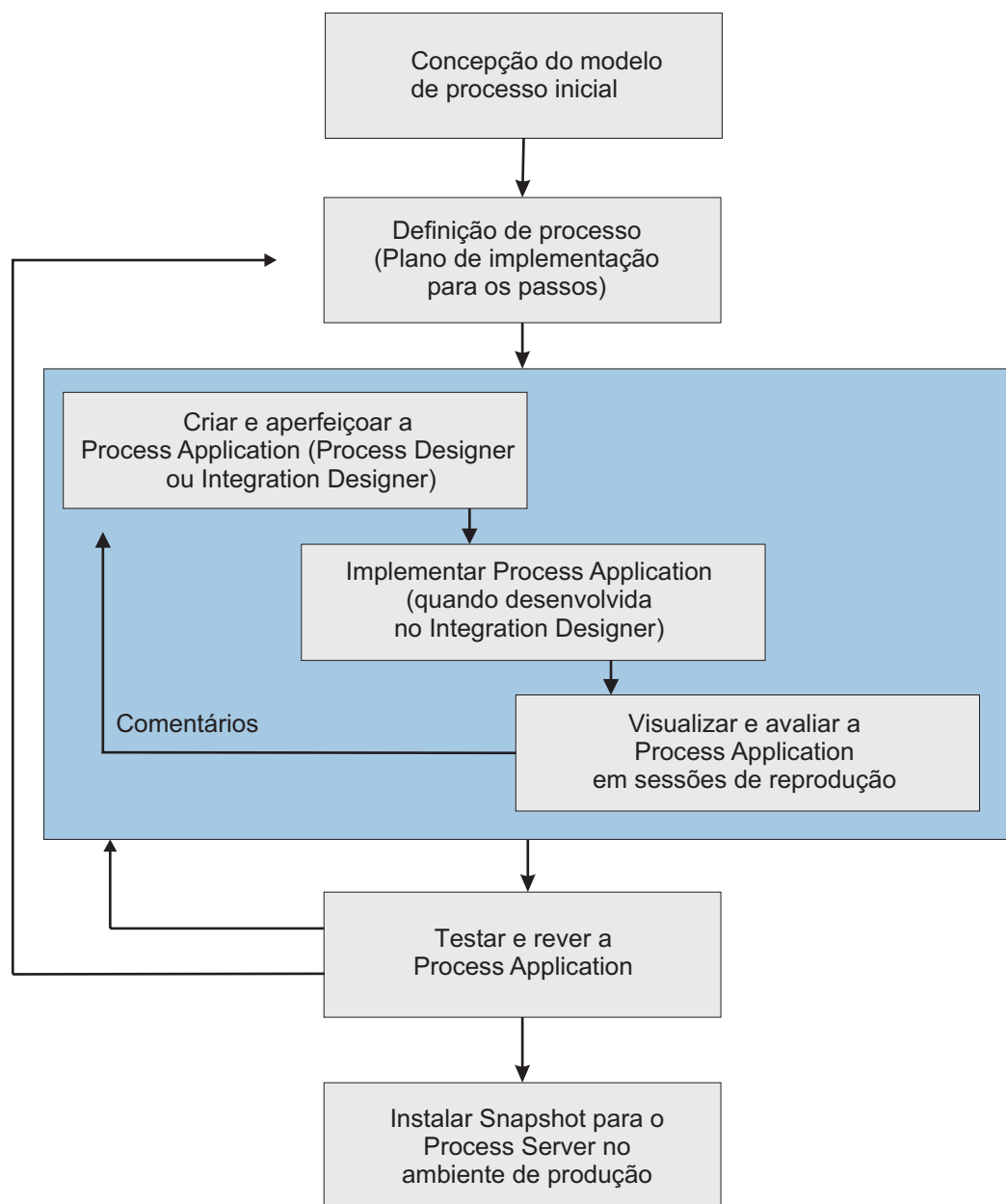
Se estiver a trabalhar no IBM Integration Designer, pode utilizar o Inspector se os respectivos projectos estiverem associados a uma Process Application. Também tem outras ferramentas de depuração e teste à disposição. Para obter mais informações sobre estas ferramentas do Integration Designer, consulte "Módulos de teste" e "Utilizar o depurador de integração para determinação do problema" nas ligações relacionadas.

## Instalar e gerir Process Applications

O ciclo de vida da process application inclui a instalação, a administração e a não implementação de snapshots. As considerações de versão também fazem parte do ciclo de vida.

Ao desenvolver processos, pode aproveitar totalmente a abordagem iterativa suportada pelas ferramentas dentro do Process Designer. Os processos evoluem ao longo do tempo, inicialmente de um estado de desenvolvimento e, posteriormente, para testes e, por fim, para a produção. Mesmo na produção, os processos podem continuar a evoluir devido a necessidades em alteração. É importante estar preparado para o ciclo de vida contínuo dos seus processos e irá ajudá-lo a concebê-los eficazmente desde o início.

A seguinte figura ilustra a abordagem iterativa ao desenvolvimento de processos.



Uma configuração do Business Process Manager típico inclui três ambientes para suportar o desenvolvimento e eventual instalação dos processos.

Ambiente	Descrição
Desenvolvimento	Crie e aperfeiçoe Process Applications no IBM Process Designer. Crie os modelos de processos e implemente os passos nos modelos que utilizam o Designer. Utilizando o Inspector, demonstre o progresso de desenvolvimento em sessões de reprodução, de forma que possa avaliar rapidamente e aperfeiçoar o protótipo. Através da Process Center Console, instale as Process Applications em Process Servers de teste ou produção.
Teste	Através da Process Center Console, instale as Process Applications no Process Server no ambiente de teste para implementar testes de garantia de qualidade formais. Pode utilizar o Inspector para ajudar a verificar e resolver questões.

Ambiente	Descrição
Produção	Quando todos os problemas reportados no teste formal estiverem resolvidos, utilize a Process Center Console para instalar as Process Applications no Process Server no seu ambiente de produção. Pode utilizar o Inspector para investigar e resolver quaisquer questões participadas no ambiente de produção.

Se pretender testar, instalar ou administrar um Snapshot da Process Application que contém conteúdo do IBM Business Process Manager Advanced ou um modelo do IBM Business Monitor, o utilizador ou grupo ao qual pertence tem de ser atribuído à função de segurança administrativa do Configurador, Operador e Implementador. Se não estiver actualmente atribuído a todas essas funções, clique em **Users and Groups (Utilizadores e grupos)** na consola administrativa do WebSphere para modificar as funções do utilizador ou grupo. Consulte Funções de segurança IBM Business Process Manager .

## Estratégias de publicação e edição

Para garantir que as Process Applications que implementar e instalar possuam os padrões de qualidade da organização, considere definir uma estratégia de publicação e edição. Quando tiver identificado os objectivos e requisitos para publicação e edição de Process Applications novas e actualizadas, pode automatizar os processos necessários para aprovar e lançar os programas.

Por exemplo, pode querer encaminhar um processo para vários gestores em várias estruturas de relatório diferentes na organização. Só depois de cada gestor terminar sessão no processo novo ou actualizado é que pode ser instalado no ambiente de produção e aplicado aos utilizadores finais. Pode criar e implementar os passos envolvidos nessa revisão no IBM Business Process Manager Advanced para garantir que todas as directrizes empresariais foram cumpridas e que possui as assinaturas necessárias. O passo final na revisão pode ser a notificação à equipa de TI que a Process Application está pronta para ser instalada.

## Criar, aceder e incorporar serviços

Processos empresariais muitas vezes utilizam serviços, que fornecem funções necessárias para o processo de negócio. Estes serviços aparecem num diagrama de processo empresarial como uma actividade ou passo. Por exemplo, um serviço em Process Designer pode invocar um serviço da Web externo ou pode invocar um serviço automatizado e complexo concebido em Integration Designer.

### Aceder a serviços externos a uma aplicação

Este cenário aborda diferentes formas de aceder a serviços que são externos à aplicação e fornece tarefas de elevado nível para aceder a estes serviços externos.

**Nota:** Este cenário aplica-se ao IBM Business Process Manager Advanced.

Numa aplicação empresarial integrada, os *serviços empresariais* interagem uns com os outros para fornecerem uma função necessária. Um serviço empresarial executa uma função ou tarefa que se pode repetir e que contribui para atingir um objectivo empresarial. Contudo, o trabalho de localização de um serviço e de estabelecimento de ligação ao mesmo não está relacionado com a função empresarial. A separação da função empresarial da tarefa de gestão de ligações ao serviço fornece flexibilidade para uma solução.

A interacção do serviço começa quando um *solicitador de serviços* envia um pedido para um *fornecedor de serviços* de execução de uma função empresarial. Este pedido é enviado sob a forma de uma *mensagem*, que define a função a ser executada. O fornecedor de serviços executa a função solicitada e envia o resultado numa mensagem ao solicitador de serviços. Normalmente, as mensagens necessitam de ser processadas para permitir que os serviços permutem dados e implementar outras funções de TI de baixo nível que são independentes dos dados e funções empresariais. Por exemplo, encaminhamento, conversão de protocolos, transformação, nova tentativa de uma chamada falhada e chamada de serviços dinâmicos. Este processamento é conhecido como *mediação*.



Existem dois tipos de módulos no IBM Integration Designer; módulos (ou módulos de integração empresariais), que são principalmente concebidos para conterem lógica empresarial (como processos empresariais, regras empresariais e máquinas automáticas) e módulos de mediação, que implementam fluxos de mediação. Apesar de existir alguma sobreposição de funções entre os dois tipos de módulos, de forma geral, recomenda-se que a lógica empresarial seja isolada nos módulos empresariais e que a lógica de mediação seja executada pelos módulos de mediação.

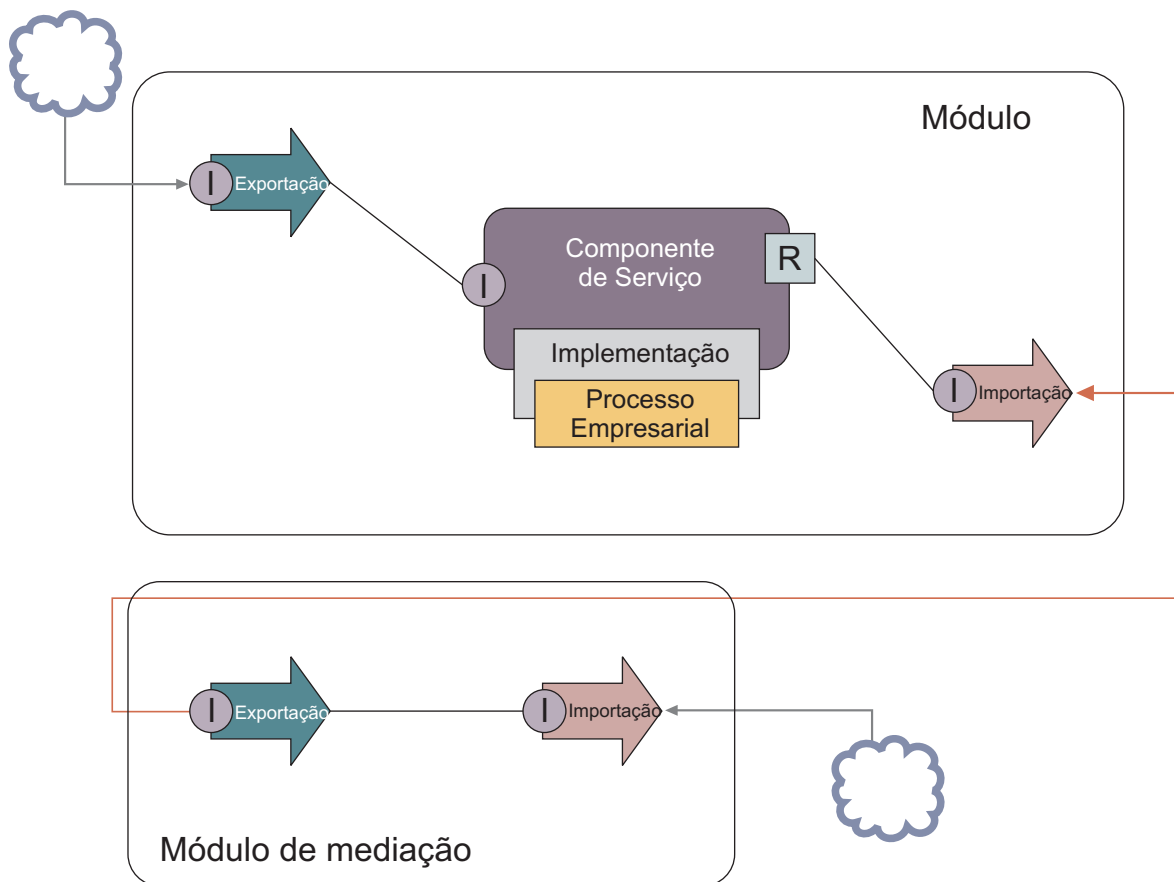
No entanto, nem sempre existe uma separação clara entre a lógica de mediação e o negócio. Nestes casos, considere a quantidade de *estados* ou dados nas variáveis que necessitará de ser processada entre as chamadas de serviço. De forma geral, se não for necessário ou for pouco o processamento de estados, considere a utilização de um componente de fluxo de mediação. Se necessitar de armazenar o estado entre chamadas de serviços, ou se tiver dados que necessitarão de ser armazenados em variáveis e processados, considere a utilização de um componente de processo empresarial. Por exemplo, se estiver a chamar vários serviços e a gravar as informações que são devolvidas de cada um, para que depois de chamar todos os serviços, e se pretender executar mais processamentos com os dados devolvidos, utilize um processo empresarial onde possa facilmente atribuir as informações devolvidas às variáveis. Por outras palavras, quanto tem demasiados estados significa que ultrapassou a linha da lógica empresarial. As seguintes secções ajudam a clarificar esta orientação.

Não existe nenhum cenário de integração e não existe nenhuma resposta tecnicamente incorrecta. As directrizes abordadas aqui constituem boas práticas para activar a flexibilidade e reutilização e são apresentadas para consideração do utilizador. Como normalmente, deverá considerar cuidadosamente os benefícios e desvantagens da implementação destes padrões na sua aplicação de integração empresarial. Consideremos algumas situações.

### **Aceder a um componente de SCA**

Um exemplo básico de acesso a um serviço é quando uma importação chama outro componente de SCA, sem necessitar de nenhuma transformação de dados. Mesmo nesta situação, é possível aceder ao serviço externo a partir de um módulo de mediação, em vez de aceder directamente a partir de um módulo empresarial. Isto irá permitir flexibilidade no futuro para alterar o destino final do serviço ou as qualidades do serviço ou administração (por exemplo, adicionar um registo) sem influenciar os componentes empresariais que consomem o serviço. Este padrão arquitectónico é conhecido como "separação de interesses".

Antes de decidir implementar este padrão, pondere os benefícios do mesmo relativamente aos potenciais efeitos gerais apresentados por outro módulo. Se o seu requisito principal for a flexibilidade e for efectuar alterações frequentes nos serviços acedidos, considere a utilização de um módulo separado, conforme é apresentado aqui. Se o desempenho for o mais importante e estiver disponível para actualizar e voltar a implementar a lógica empresarial, considere a utilização de um único módulo.



Seguem-se as tarefas de nível elevado para concretizar este exemplo.

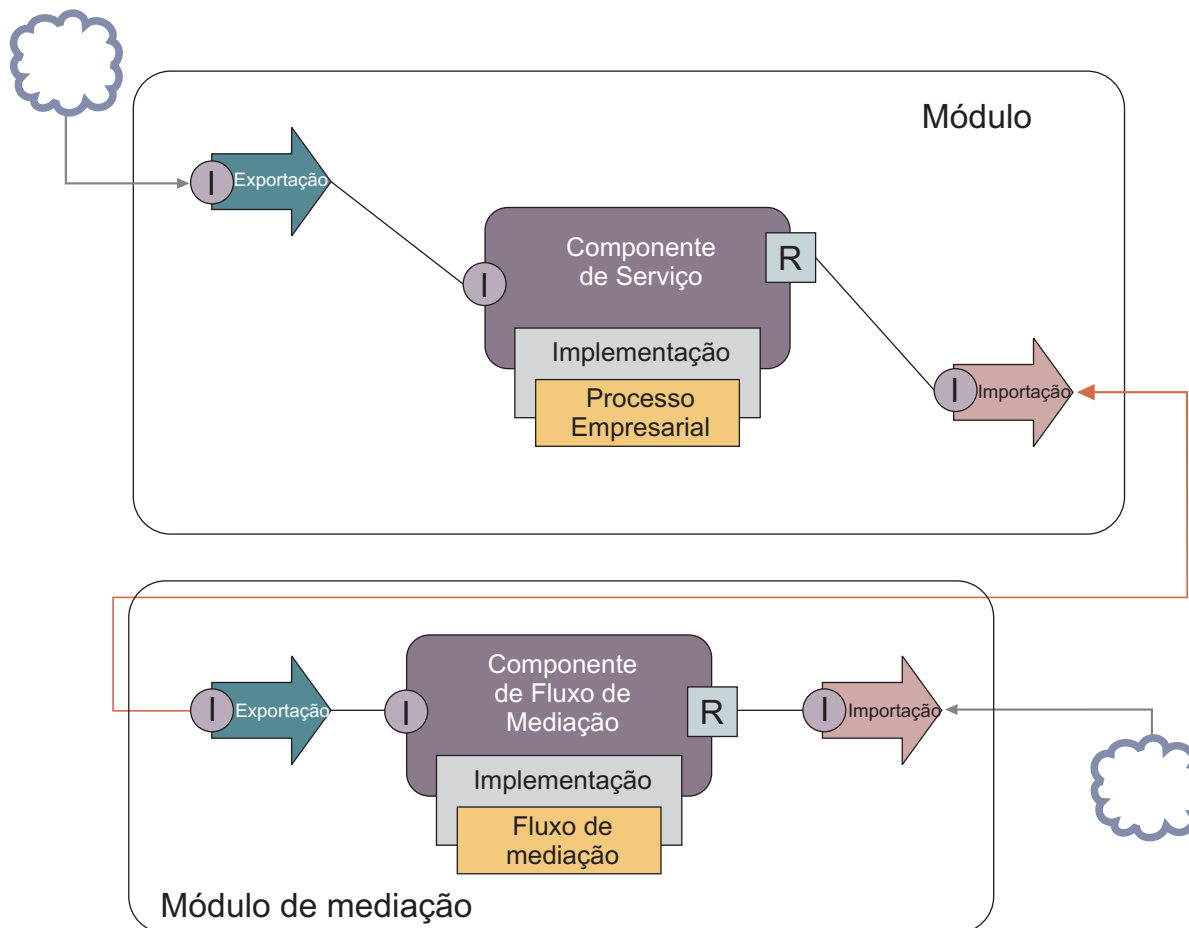
1. Crie um módulo de mediação. Para obter instruções passo a passo, consulte Criar módulos de mediação .
2. No módulo de mediação, crie uma importação com a associação apropriada para o serviço externo ao qual pretende aceder. Para obter instruções passo a passo, consulte Criar importações. Para obter mais informações sobre associações, consulte Associações
3. Crie uma exportação e atribua-lhe a mesma interface que a importação. Para obter instruções passo a passo, consulte Criar exportações.
4. Gere uma associação de SCA para a exportação. Para obter instruções passo a passo, consulte Gerar associações de SCA
5. No conjunto do módulo de mediação, ligue a exportação à importação. Guarde o módulo de mediação.
6. Crie um módulo. Para obter instruções passo a passo, consulte Criar um módulo para serviços empresariais
7. Adicione uma exportação e um componente.
8. Na vista do Business integration, arraste a exportação que criou no módulo de mediação (no passo 4) para o conjunto do módulo. Será criada uma importação com a mesma associação que a exportação.
9. Ligue a exportação ao componente e o componente à importação.
10. Adicione a implementação do componente. Para obter informações sobre tipos de implementação, consulte Implementações

Posteriormente, pode adicionar uma lógica de mediação como, por exemplo, registo ou encaminhamento para o módulo de mediação sem afectar o módulo empresarial.



## Adicionar mediação

Por vezes, chamar simplesmente um serviço externo não é suficiente. Por vezes, é necessário processar primeiro, adicionando um módulo de mediação como intermediário entre o solicitador e o fornecedor de serviços.



Seguem-se algumas das funções que o fluxo de mediação intermediário irá executar:

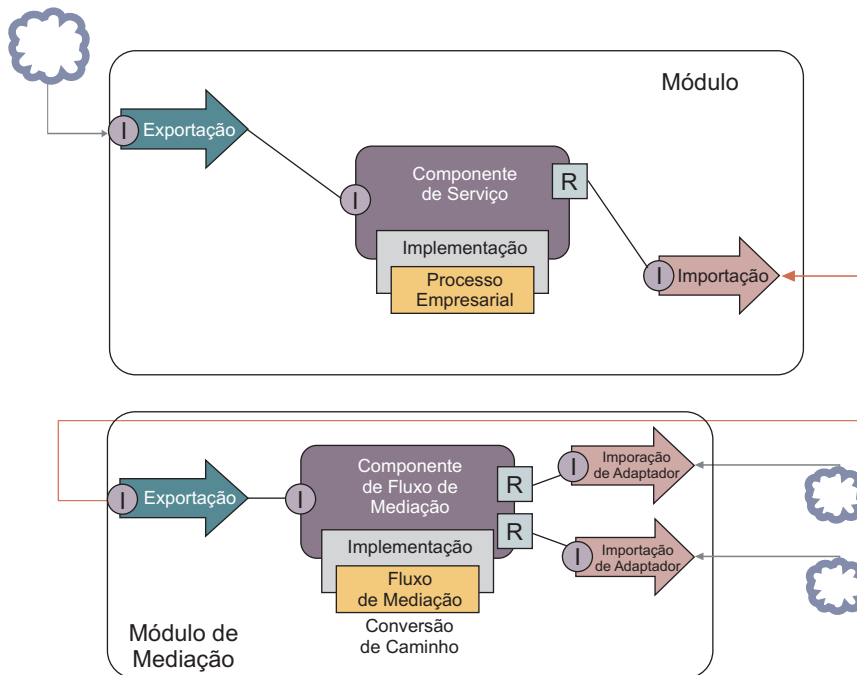
- Definição de cabeçalhos de protocolo. Para obter mais informações, consulte o tópico Protocol conversion (Conversão de protocolo) no Information Center do WebSphere Enterprise Service Bus.
- Transformação do parâmetro ou interface utilizando uma Correspondência de Objectos de Negócio ou uma Primitiva de Mapeamento. Transformar mensagens
- Seleção de um determinado serviço a partir de uma lista estática, utilizando uma primitiva de Filtro de Mensagem. Filtro de mensagens
- Chamada de vários serviços para agregar os resultados, utilizando as primitivas Dispersar e Concentrar. Agregar e difundir mensagens
- Resolução de falhas de chamada de serviços obtendo o mesmo serviço ou através da chamada de um serviço diferente, utilizando uma primitiva de Invocação do Serviço. Repetir uma chamada de serviço falhada
- Encaminhamento dinâmico escolhendo qual o serviço a utilizar no tempo de execução, em vez do tempo de integração, o que permite que os serviços sejam unidos de forma mais livre e os negócios reajam mais rapidamente à mudança. É possível adicionar novos serviços sem tocar nos módulos que foram implementados no ambiente em tempo de execução. O encaminhamento dinâmico é mais

poderoso quando é utilizado com um registo, o que requer a utilização da primitiva de mediação Procura do Destino Final. Seleccionar destinos finais de forma dinâmica

## Aceder ao Enterprise Information Systems

Os serviços e os artefactos nos sistemas externos podem ser importados para o Integration Designer. Um assistente descobre aplicações e dados num Enterprise Information System (EIS), e permite-lhe criar serviços a partir das aplicações e dos dados que foram descobertos. Os artefactos gerados são interfaces e objectos de negócio que podem ser utilizados por componentes num módulo.

A utilização de um módulo de mediação intermediário entre um módulo e um sistema central torna-o mais reutilizável. No exemplo abaixo, um fluxo de mediação é utilizado para encaminhar para o sistema central correcto e transformar os dados no formato requerido pelo sistema central.



Seguem-se as tarefas de nível elevado para este exemplo:

1. Utilize o assistente do serviço externo para ligar ao sistema central. A utilização do assistente do serviço externo para aceder a serviços externos segue um padrão semelhante, independentemente do adaptador que estiver a utilizar. Para obter mais informações sobre como utilizar o assistente do serviço, consulte Padrão de acesso a serviços externos com adaptadores
2. Crie um módulo. Para obter instruções passo a passo, consulte Criar um módulo para serviços empresariais
3. Adicione uma exportação, um componente e uma importação à associação de SCA. Para obter mais informações, consulte Chamada de serviços
4. Adicione uma interface à exportação e ligue a exportação ao componente.
5. Adicione a implementação do componente. Na implementação, defina uma propriedade que indique qual o serviço do sistema central que será acedido. Para obter informações sobre tipos de implementação, consulte Implementações
6. Crie um módulo de mediação com uma exportação que possua uma associação de SCA e a mesma interface que a importação do módulo que criou no passo 2.
7. Ligue a exportação a um componente do fluxo de mediação.
8. Crie uma importação para cada sistema central ao qual pretenda aceder utilizando o adaptador de saída apropriado a partir da paleta do editor de montagem.

9. Ligue o componente do fluxo de mediação às importações.
10. Implemente o componente do fluxo de mediação. Utilize uma primitiva de Filtro de mensagem para escolher a importação baseada num conjunto de propriedade na lógica empresarial e utilize uma Primitiva de mapeamento para cada importação de adaptador. Filtro de mensagens.
11. No módulo, seleccione a exportação do módulo de mediação como o serviço a ser importado para o módulo. Para obter informações passo a passo, consulte Chamar um serviço a partir de outro módulo.

Posteriormente, pode efectuar alterações como, por exemplo, adicionar um adaptador ou alterar um adaptador para apontar para um sistema central diferente, com o mínimo impacto para a lógica empresarial.

## **Aceder aos sistemas de serviço de mensagens**

Para que o seu módulo de Arquitectura de Componente de Serviço (SCA, Service Component Architecture) comunique com um cliente de serviço de mensagens JMS, MQ ou MQ JMS existente, necessita de criar interfaces, objectos de negócio e associações para importações e exportações. Consulte Corresponder uma mensagem a uma interface de SCA.

Os fluxos de mediação utilizam mensagens que fornecem acesso às informações do cabeçalho e contexto para além de objectos de negócio. Se pretender aceder às informações do cabeçalho JMS ou a uma propriedade JMS personalizada, utilize um fluxo de mediação. Se estiver a executar uma integração num sistema MQ e pretender aceder às informações do cabeçalho MQ, utilize um fluxo de mediação.

## **Criar ou chamar um serviço da Web**

Os serviços da Web são aplicações auto-inclusas que executam funções de negócio que vão desde uma consulta simples a interações de processos empresariais complexos. Pode chamar um serviço da Web existente ou desenvolver um novo serviço da Web para satisfazer as suas necessidades. Este cenário irá descrever os passos e fornecer-lhe informações adicionais.

Embora possa não criar todos os seus serviços de raiz utilizando o IBM Integration Designer, alguns serviços serão de facto criados desta forma. Quando está a trabalhar com o editor de montagem e o editor de processos empresariais para montar serviços num processo empresarial, provavelmente achará que faltam alguns serviços. Deste modo, poderá ser útil criar esses serviços em falta utilizando as ferramentas do IBM Integration Designer. O contrário também se aplica - depois de criar um novo processo, poderá decidir que seria útil expor todos ou um subconjunto das operações do processo como um serviço para consumo de outros.

**Nota:** Este cenário aplica-se aos utilizadores do IBM Integration Designer para IBM Process Server.

Existem vários motivos para o desenvolvimento de serviços da Web ao utilizar o IBM Integration Designer:

- A criação de serviços no IBM Integration Designer permite-lhe implementar o serviço utilizando regras empresariais.
- O desenvolvimento no IBM Integration Designer permite-lhe desenvolver um serviço de Java™ e expô-lo tanto como um serviço da Web como através de SCA.
- A correspondência de interfaces sem codificação é uma vantagem. Pode retirar todas as correspondências de dados do código Java e deixar um programa de caixa negra Java simples para o programador Java.
- IBM Integration Designer mostra todos os serviços e relações num só local.
- A capacidade de reconstrução também irá ajudar no desenvolvimento de serviços da Web ao utilizar o IBM Integration Designer.

Tenha em mente que os serviços da Web não devem ser considerados como a solução para todos os problemas de integração. No entanto, tal como com qualquer outra tecnologia ou abordagem arquitectónica, existem vantagens inerentes na utilização dos serviços da Web no local e momento certos.

## **Exportações, importações e associações**

O IBM Integration Designer permite-lhe importar serviços da Web padrão e utilizá-los nas suas aplicações compostas.

No IBM Integration Designer, utilize o editor de montagem para programar serviços. Siga o processo padrão para criar módulos, módulos de mediação, bibliotecas e componentes. Em seguida, pode utilizar exportações, importações e associações para partilhar e aceder a esses serviços. Os passos para essas tarefas básicas estão listados abaixo e as ligações conduzem a informações mais detalhadas de cada tarefa.

Pode utilizar uma das duas associações para serviços da Web - uma associação de serviço da Web ou uma associação de HTTP. Uma associação de serviço da Web fornece uma especificação para transmissão de mensagens de e para um serviço da Web. As ferramentas ajudam a gerar automaticamente uma associação de serviço da Web. Uma associação de HTTP é um protocolo padrão de pedido e resposta entre os clientes e o servidor, conforme definido pelo protocolo HTTP publicado pelo consórcio da World Wide Web (W3C). Necessitará de fornecer algumas informações iniciais da configuração da associação se utilizar a associação de HTTP.

1. Criar uma exportação para publicar o serviço do módulo para ser utilizado por outros módulos.
2. Gerar uma associação para a exportação.
  - Gerar uma associação de serviço da Web para a exportação.
  - Gerar uma associação de exportação de HTTP.
3. Criar uma importação para chamar um serviço existente que não faça parte do módulo que está a montar.
  - Gerar uma associação de serviço da Web para a importação.
  - Gerar uma associação de importação de HTTP.

Leia o tópico ligado se pretende Chamar um serviço da Web das Páginas JavaServer.

## **Capacidades de programação de serviços da Web**

Quando abrir um editor associado a um processo de criação de serviços da Web, poderá encontrar a janela Confirmar activação que apresenta a seguinte informação:

Esta acção requer a activação de "Implementação de serviços da Web".  
Activar a capacidade requerida?

O IBM Integration Designer fornece uma função de filtragem conhecida como *capacidades*. Nas definições de Preferências, as funções e as ferramentas estão categorizadas em capacidades e é possível activar e desactivar categorias das capacidades ou funções do subconjunto de qualquer categoria. Consulte Capacidades para obter mais informações.

---

## **Saiba mais sobre os conceitos chave**

Utilize esta secção como ponto inicial para investigar as tecnologias utilizada no e pelo IBM Business Process Manager.

### **Criar cenários**

Utilize cenários para compreender e trabalhar com componentes e produtos da família de gestão de processos de negócio.

## Criação de versões

O ciclo de vida de uma process application começa com a criação da process application e continua através de um ciclo de actualização, implementação, co-implementação, anulação da implementação e arquivo da process application. O *Arquivo de versões* é um mecanismo utilizado para gerir o ciclo de vida da process application através da identificação, de forma exclusiva, das versões exclusivas da process application.

O modo como o arquivo de versões funciona no IBM Business Process Manager depende daquilo que se implementar numa Process Application, implementada a partir do repositório em IBM Process Center, ou uma aplicação empresarial directamente implementada a partir do IBM Integration Designer.

As Process Applications e Toolkits que implementar num ambiente em tempo de execução a partir do Process Center são, por predefinição, alvo de arquivo de versões. Para aplicações empresariais, pode seleccionar arquivar as versões de módulos e bibliotecas em IBM Integration Designer.

Em acréscimo, pode criar versões de uma tarefa humana ou máquina de estado, de modo a que possam coexistir versões múltiplas da tarefa ou máquina de estado no ambiente em tempo de execução.

### Criação de versões das Process Applications

A criação de versões fornece a capacidade do ambiente em tempo de execução de identificar Snapshots no ciclo de vida de uma Process Application e a ser capaz de executar de forma concorrente múltiplos snapshots num process server.

Para entender como as process applications com versões, é importante lembrar-se de que uma process application é um contentor que retém vários artefactos utilizados em ou por process application (por exemplo, modelos de processos, serviços, pistas, modelos de supervisão e referências de toolkit). Qualquer arquivo de versões é feito neste nível de contentor, não ao nível dos artefactos individuais. Para process applications, isto significa que o arquivo de versões acontece quando tira um snapshot.

Pode comparar snapshots para determinar diferenças entre as versões. Por exemplo, se um programador corrigiu um problema com um serviço e tirou, nessa altura, um Snapshot da sua Process Application ou Toolkit e, depois, outro programador fez várias alterações ao mesmo serviço e tirou um novo Snapshot, o gestor de projecto pode comparar os dois Snapshots para determinar que alterações foram feitas e por quem foram feitas. Se o gestor de projecto tiver decidido que as alterações adicionais ao serviço não foram satisfatórias, este pode voltar ao Snapshot da correcção original.

Pode executar diferentes versões (snapshots) de uma process application concorrentemente num servidor; quando instala um novo snapshot, ou para remover o original ou deixá-lo a executar.

### Contexto da versão

Cada snapshot tem metadados únicos para identificar a versão (referida como contexto de versão). Atribua esse identificador, mas a IBM recomenda a utilização de um sistema de versão numérica de três dígitos no formato <major>.<minor>.<service>. Consulte os tópicos sobre as convenções de nomenclatura para uma descrição mais detalhada deste esquema de criação de versões.

O IBM Business Process Manager atribui um espaço de nomes global para cada Process Application. O espaço de nomes global é, especificamente, a sugestão da Process Application ou um Snapshot em particular da Process Application. O nome da versão utilizado pelo servidor não pode ser maior do que sete caracteres, para que o nome atribuído seja um acrónimo que utiliza caracteres do nome atribuído ao Snapshot. Os acrónimos dos Snapshot são idênticos aos seus nomes, se estes estiverem de acordo com o estilo IBM VRM e não tiverem mais de sete caracteres. Por exemplo, um nome do Snapshot de 1.0.0 terá um acrónimo de 1.0.0, e um nome de Snapshot de 10.3.0 terá o acrónimo de 10.3.0. O acrónimo do Snapshot terá a garantia de ser único dentro do contexto da Process Application dentro do âmbito do Process Center Server. Por essa razão, o utilizador não pode editar o acrónimo do Snapshot.

## Considerações de arquivo de versões para o process applications em múltiplos conjuntos

Pode instalar a mesma versão de uma Process Application em vários conjuntos dentro da mesma célula. Para diferenciar entre estas múltiplas instalações da mesma versão da Process Application, crie um Snapshot para cada instalação e inclua um Snapshot de uma única célula no nome do Snapshot (por exemplo, v1.0\_cell1\_1 e v1.0\_cell1\_2). Cada snapshot é uma versão de uma process application (de uma perspectiva de gestão pura do ciclo de vida), mas o conteúdo e a função são os mesmos.

Quando instala uma Process Application num conjunto, é executada uma sincronização automática dos nós.

## Considerações de arquivo de versões para toolkits Process Designer

Lembre-se de que esses snapshots da process application são tipicamente tirados quando está pronto para testar ou instalar. Snapshots de Toolkit, no entanto, são normalmente tirados quando esse toolkit está pronto a ser utilizado pela process applications. Depois, se quiser actualizar o Toolkit, deve tirar outro Snapshot da "sugestão" quando estiver pronto, e os proprietários das Process Applications e Toolkits podem decidir se querem passar para o novo Snapshot.

## Arquivo de versões de módulos e bibliotecas

Se um módulo ou biblioteca estiverem numa process application ou num toolkit, este obtém o ciclo de vida da process application ou do toolkit (versões, snapshots, controlos e outros). Os nomes do módulo e biblioteca devem ser únicos no âmbito de uma Process Application ou Toolkit.

Este tópico descreve o arquivo de versões de módulos e bibliotecas que são utilizadas com Process Application. Tenha em atenção, porém, que se implementar módulos directamente de IBM Integration Designer para o Process Server, pode continuar a seguir o procedimento de atribuir números de versão a módulos durante a implementação, como descrito em "Criar versões arquivadas de módulos e bibliotecas".

Um módulo ou biblioteca associado com IBM Process Center deve ter as respectivas bibliotecas dependentes na mesma Process Application ou num Toolkit dependente.

A tabela seguinte lista as selecções que pode efectuar no editor de dependência em IBM Integration Designer quando uma biblioteca é associada a uma Process Application ou Toolkit:

*Tabela 7. Dependências para Módulo, Process App ou Toolkit e Bibliotecas globais*

Âmbito da biblioteca	Descrição	Pode depender de . . .
Módulo	Existe uma cópia desta biblioteca no servidor para cada módulo o utilizar.	Uma biblioteca de âmbito de módulo pode depender de todos os tipos de bibliotecas.
Process App ou Toolkit	A biblioteca é partilhada entre todos os módulos no âmbito da Process Application ou Toolkit. Esta definição entra em vigor caso a implementação seja efectuada através do IBM Process Center. Se a implementação ocorrer fora do IBM Process Center, a biblioteca é copiada para cada módulo. <b>Nota:</b> As bibliotecas criadas na versão 8 do IBM Integration Designer têm um nível de partilha de <b>Process App or Toolkit (Process App ou Toolkit)</b> por predefinição.	Uma biblioteca de este tipo apenas pode depender de bibliotecas globais.

Tabela 7. Dependências para Módulo, Process App ou Toolkit e Bibliotecas globais (continuação)

Âmbito da biblioteca	Descrição	Pode depender de . . .
Global	A biblioteca é partilhada entre todos os módulos em execução.	Uma biblioteca global apenas pode depender de outras bibliotecas globais. <b>Nota:</b> Deve configurar uma biblioteca partilhada em WebSphere de modo a implementar a biblioteca global. Consulte “Dependências de módulo e biblioteca” para obter mais informações.

## Módulos e bibliotecas associados a Process Applications ou Toolkits

Não é necessário criar versões de módulos e bibliotecas associados a Process Applications ou Toolkits.

Não é necessário criar versões dos módulos e bibliotecas associados a uma Process Application ou um Toolkit. Na realidade, não é possível criar uma versão de um módulo ou de uma biblioteca associada a uma Process Application ou a um Toolkit no editor de dependências. Os módulos e as bibliotecas associados a uma Process Application ou um Toolkit utilizam Snapshots, uma função no Process Center, para atingir o mesmo resultado que uma versão.

As bibliotecas associadas a uma Process Application ou a um Toolkit não terão um número de versão necessário na secção Bibliotecas do editor de dependências porque não é necessária nenhuma versão.

## Convenções de nomenclatura

É utilizada uma convenção de nomenclatura para diferenciar as diferentes versões de uma process application à medida que percorre o ciclo de vida de actualização, implementação, co-implementação, não implementação e arquivo.

Esta secção proporciona-lhe as convenções que são utilizadas para identificar, exclusivamente, versões de uma Process Application.

Um *contexto de versão* é uma combinação de acrónimos que descrevem, exclusivamente, uma Process Application ou Toolkit. Cada tipo de acrónimo tem uma convenção de nomenclatura. O acrónimo é limitado a um comprimento máximo de sete caracteres do conjunto de caracteres [A-Z0-9\_], excepto para o acrónimo do Snapshot, que também pode incluir um ponto final.

- O acrónimo da Process Application é criado quando a Process Application é criada. Pode ter um comprimento máximo de sete caracteres.
- O acrónimo do Snapshot é automaticamente criado quando o Snapshot é criado. Pode ter um comprimento máximo de sete caracteres.

Se o nome do Snapshot cumprir os critérios de um acrónimo de Snapshot válido, o nome e o acrónimo do Snapshot serão o mesmo.

**Nota:** Quando utilizar a função de encaminhamento adequada à versão do componente do fluxo de mediação, atribua um nome ao seu Snapshot para que esteja em conformidade com o esquema da <versão>.<edição>.<modificação> (por exemplo, 1.0.0). Devido ao acrónimo do Snapshot ser limitado a sete caracteres, os valores dos dígitos são limitados a um máximo de cinco dígitos no total (cinco dígitos mais dois pontos finais). Assim, deve ser tido cuidado quando os campos dos dígitos são aumentados, pois tudo além dos primeiros sete caracteres é truncado.

Por exemplo, o nome do Snapshot 11.22.33 resulta num acrónimo de Snapshot 11.22.3.

- O acrónimo da sugestão é criado automaticamente a partir do primeiro carácter de cada palavra do nome da sugestão. Por exemplo, uma nova sugestão criada com o nome **A minha nova sugestão** resultaria num valor de acrónimo de AMNS.



O nome e acrónimo predefinidos da sugestão são **Principal**. A implementação num servidor IBM Process Center inclui o acrónimo de sugestão no contexto de arquivo de versões caso o acrónimo da versão não seja **Principal**.

Uma definição de um processo empresarial numa Process Application é normalmente identificada pelo acrónimo do nome da Process Application, pelo acrónimo do Snapshot e pelo nome da definição do processo empresarial. Seleccione nomes exclusivos para as suas definições de processo empresarial, sempre que possível. Quando existirem nomes duplicados, poderá encontrar os seguintes problemas:

- Poderá ser incapaz de expor as definições do processo empresarial como serviços da Web sem algum tipo de mediação.
- Também poderá ser incapaz de chamar uma definição de processo empresarial criada no IBM Process Designer a partir de um processo BPEL criado no IBM Integration Designer.

O contexto da versão varia consoante o modo como a Process Application é implementada.

### **Convenções de nomenclatura para implementações do servidor do Process Center:**

No servidor IBM Process Center, pode implementar um Snapshot de uma Process Application e um Snapshot de um Toolkit. Adicionalmente, pode implementar a sugestão de uma Process Application ou a sugestão de um Toolkit. (Uma *sugestão* é a versão funcional actual da sua Process Application ou Toolkit.) O contexto da versão varia, consoante o tipo de implementação.

Para Process Application, a sugestão da Process Application ou o Snapshot específico da Process Application é usado para identificar, exclusivamente, a versão.

Os toolkits podem ser implementados com uma ou mais aplicações, mas o ciclo de vida de cada toolkit está ligado ao ciclo de vida da process application. Cada Process Application tem a sua própria cópia do Toolkit ou Toolkits dependentes implementados no servidor. Um Toolkit implementado não é partilhado entre Process Application.

Se a sugestão associada à Process Application for denominada algo que não a predefinição de **Principal**, o acrónimo da sugestão também faz parte do contexto da versão.

Para obter mais informações, consulte a secção “Exemplos” na página 33, mais à frente neste tópico.

### **Snapshots da Process Application**

Para implementações do Snapshot da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo da sugestão da Process Application (se for utilizada outra sugestão que não **Principal**)
- Acrónimo do snapshot da Process Application

### **Toolkits autónomos**

Para implementações de Snapshot de Toolkit, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome de Toolkit
- Acrónimo da sugestão de Toolkit (se for utilizada outra sugestão que não **Principal**)
- Acrónimo do Snapshot do Toolkit

### **Sugestões**

Sugestões da Process Application utilizadas durante os testes iterativos no Process Designer. Apenas podem ser implementadas nos servidores do Process Center.



Para implementações da sugestão da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo da sugestão da Process Application (se for utilizada outra sugestão que não **Principal**)
- "Sugestão"

As sugestões de Toolkit também são utilizadas para testes iterativos no Process Designer. Não são implementadas num servidor de produção.

Para implementações da sugestão do Toolkit, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome de Toolkit
- Acrónimo da sugestão de Toolkit (se for utilizada outra sugestão que não **Principal**)
- "Sugestão"

## Exemplos

Os recursos devem ser exclusivamente designados e identificados externamente utilizando o contexto da versão.

- A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados. Neste exemplo, uma sugestão da Process Application utiliza o nome predefinido da sugestão (**Principal**):

*Tabela 8. Sugestão da Process Application com nome predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Principal</b>
Acrónimo da sugestão da Process Application	"" (quando a pista é <b>Principal</b> )
Snapshot da Process Application	
Acrónimo do snapshot da Process Application	<b>Sugestão</b>

Quaisquer módulos SCA associados a esta sugestão da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 9. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-Sugestão-M1	PA1-Sugestão-M1.ear
M2	PA1-Sugestão-M2	PA1-Sugestão-M2.ear

- A tabela seguinte mostra um exemplo de uma sugestão de Process Application que utiliza um nome de sugestão não predefinido:

*Tabela 10. Sugestão da Process Application com nome não predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Sugestão1</b>
Acrónimo da sugestão da Process Application	<b>T1</b>
Snapshot da Process Application	
Acrónimo do snapshot da Process Application	<b>Sugestão</b>

Quaisquer módulos SCA associados a esta sugestão da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 11. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-T1-Sugestão-M1	PA1-T1-Sugestão-M1.ear
M2	PA1-T1-Sugestão-M2	PA1-T1-Sugestão-M2.ear

As convenções de nomenclatura semelhantes aplicam-se às implementações de sugestões de Toolkit e de Snapshots avançadas. Estas também podem ser aplicadas em Snapshots avançados instaladas no Process Server.

- A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados. Neste exemplo, um snapshot da Process Application utiliza o nome predefinido da sugestão (**Principal**):

*Tabela 12. Snapshot da Process Application com nome predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Principal</b>
Acrónimo da sugestão da Process Application	"" (quando a pista é <b>Principal</b> )
Snapshot da Process Application	<b>Process Shapshot V1</b>
Acrónimo do snapshot da Process Application	<b>PSV1</b>

Quaisquer módulos SCA associados a este snapshot da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 13. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- A tabela seguinte mostra um exemplo de um snapshot de Process Application que utiliza um nome de sugestão não predefinido:

*Tabela 14. Snapshot da Process Application com nome não predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Sugestão1</b>
Acrónimo da sugestão da Process Application	<b>T1</b>
Snapshot da Process Application	<b>Process Snapshot V1</b>
Acrónimo do snapshot da Process Application	<b>PSV1</b>

Quaisquer módulos SCA associados a este snapshot da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 15. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear

Tabela 15. Módulos SCA e ficheiros EAR adequados à versão (continuação)

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

### Convenções de nomenclatura para implementações do Process Server:

No Process Server, pode implementar o Snapshot de uma Process Application. O acrónimo do Snapshot da Process Application é utilizado para identificar, exclusivamente, a versão.

Para implementações do Snapshot da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo do Snapshot da Process Application

Os recursos devem ser exclusivamente designados e identificados externamente utilizando o contexto da versão. A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados:

Tabela 16. Exemplo de nomes e acrónimos

Tipo de nome	Exemplo
Nome de Process Application	Process Application 1
Acrónimo do nome da Process Application	PA1
Snapshot da Process Application	1.0.0
Acrónimo do Snapshot da Process Application	1.0.0

Um recurso, como um módulo ou biblioteca, tem o contexto da versão como parte da sua identidade.

A tabela seguinte mostra um exemplo de dois módulos e de como os ficheiros EAR associados incluem o contexto da versão:

Tabela 17. Módulos SCA e ficheiros EAR adequados à versão

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-1.0.0-M1	PA1-1.0.0-M1.ear
M2	PA1-1.0.0-M2	PA1-1.0.0-M2.ear

A tabela seguinte mostra um exemplo de duas bibliotecas de âmbito da Process Application e de como os ficheiros JAR associados incluem o contexto da versão:

Tabela 18. Bibliotecas de âmbito do processo e ficheiros JAR adequados à versão

Nome da biblioteca de âmbito da Process Application SCA	Nome adequado à versão	Nome JAR adequado à versão
Lib1	PA1-1.0.0-Lib1	PA1-1.0.0-Lib1.jar
Lib2	PA1-1.0.0-Lib2	PA1-1.0.0-Lib2.jar

### Associações adequadas à versão

As Process Applications podem conter módulos SCA que incluam associações de importação e de exportação. Quando co-implementar as aplicações, a associação para cada versão da aplicação deve ser única. Algumas associações são automaticamente actualizadas durante a implementação para garantir a exclusividade entre versões. Noutros casos, tem de actualizar a associação após a implementação para garantir a respectiva exclusividade.

Uma associação *adequada à versão* é agrupada numa versão específica de uma Process Application, o que garante a respectiva exclusividade entre Process Applications. As seguintes secções descrevem as associações que são automaticamente actualizadas para serem adequadas à versão e quaisquer acções que necessite de tomar, no tempo de execução, quando uma associação não for adequada à versão. Para mais informações sobre questões a ter em conta quando cria módulos, consulte “Considerações ao utilizar associações”.

## SCA

O nome do destino de uma associação SCA é automaticamente alterado para ser adequado à versão durante a implementação, caso as associações de importação e de exportação do módulo sejam definidas no mesmo âmbito da Process Application.

Se as associações não forem definidas no mesmo âmbito da Process Application, é registada uma mensagem informativa. Deve modificar a associação de importação após a implementação para alterar o endereço do destino final. Pode utilizar a consola administrativa para alterar o endereço do destino final.

### Serviço da Web (JAX-WS ou JAX-RPC)

O nome do endereço do destino final de uma associação de serviço da Web é automaticamente alterado para ser adequado à versão, durante a implementação, caso todas as seguintes condições forem verdadeiras:

- Seguiu a convenção de nomenclatura predefinida para o endereço:  
`http://ip:port/ModuleNameWeb/sca/ExportName`
- O endereço do destino final é SOAP/HTTP.
- As associações de importação e exportação do módulo são definidas no mesmo âmbito da Process Application.

Se estas condições não forem verdadeiras, é registada uma mensagem informativa. A acção que adoptar, de seguida, dependerá do modo como estiver a implementar a sua Process Application:

- Se estiver a co-implementar a sua Process Application, deve alterar o nome, manualmente, ao URL de destino final de SOAP/HTTP ou à fila de destino de SOAP/JMS, para que seja único entre as versões da Process Application. Pode utilizar a consola administrativa após a implementação para alterar o endereço do destino final.
- Se apenas estiver a implementar uma versão única da Process Application, pode ignorar esta mensagem

Para a co-implementação do Snapshot da associação do serviço da Web de SOAP/ JMS, a acção que tomar depende de como estiver a implementar a sua Process Application:

- Se a importação e exportação do destino estiverem na mesma Process Application, execute os seguintes passos antes de publicar a Process Application para Process Center e crie o Snapshot:
  1. Altere o URL do destino final da exportação. Certifique-se de que o destino e o fornecedor de ligação são únicos.
  2. Altere o URL do destino final da importação para que seja igual ao que especificou para a exportação no passo anterior.
- Se a importação e exportação do destino estiverem em diferentes Process Applications, execute os seguintes passos:
  1. Altere o URL do destino final da exportação. Certifique-se de que o destino e o fornecedor de ligação são únicos.
  2. Publique a Process Application para Process Center.
  3. Crie o Snapshot.
  4. Implemente a Process Application para o Process Server.

5. Utilize a consola administrativa da WebSphere para alterar o URL do destino final da importação correspondente para que seja igual ao que especificou para a exportação.

## HTTP

O nome do endereço do URL do destino final de uma associação HTTP é automaticamente alterado para ser adequado à versão, durante a implementação, caso todas as seguintes condições forem verdadeiras:

- Seguiu a convenção de nomenclatura predefinida para o endereço:  
**`http(s)://ip:port/ModuleNameWeb/contextPathInExport`**
- As associações de importação e exportação do módulo são definidas no mesmo âmbito da Process Application.

Se estas condições não forem verdadeiras, é registada uma mensagem informativa. A acção que adoptar, de seguida, dependerá do modo como estiver a implementar a sua Process Application:

- Se estiver a co-implementar a sua Process Application, deve alterar o nome, manualmente, ao URL de destino final para que seja único entre as versões da Process Application. Pode utilizar a consola administrativa após a implementação para alterar o endereço do destino final.
- Se apenas estiver a implementar uma versão única da Process Application, pode ignorar esta mensagem

## JMS e JMS genérico

Os JMS gerados pelo sistema e as associações de JMS genéricos são automaticamente adequados à versão.

**Nota:** Para JMS definidos pelo utilizador e associações de JMS genéricos, não ocorre qualquer alteração automática do nome durante a implementação para permitir às associações serem adequadas à versão. Se a associação for definida pelo utilizador, deve alterar o nome aos seguintes atributos, para que sejam únicos entre as versões da Process Application:

- Configuração do destino final
- Fila de destino de recepção
- Nome da porta de recepção (se definido)

Defina o destino de Enviar correspondente se alterar o destino final do módulo de destino.

## MQ/JMS e MQ

Não ocorre nenhuma alteração automática do nome durante a implementação para permitir às associações de tipo MQ/JMS ou MQ serem adequadas à versão.

Deve alterar o nome aos seguintes atributos, para que sejam únicos entre as versões da Process Application:

- Configuração do destino final
- Fila de destino de recepção

Defina o destino de Enviar correspondente se alterar o destino final do módulo de destino.

## EJB

Não ocorre nenhuma alteração automática do nome durante a implementação para permitir às associações de tipo EJB serem adequadas à versão.

Deve alterar o nome do atributo JNDI para que seja único entre as versões da Process Application.

Tenha em atenção que as aplicações do cliente também necessitam de ser actualizadas para utilizarem os novos nomes JDNI.

## EIS

O nome de um adaptador de recurso é automaticamente alterado para ser adequado à versão, durante a implementação, desde que o nome de recurso predefinido (*ModuleNameApp:Adapter Description*) não tenha sido modificado.

Se o nome de recurso predefinido tiver sido modificado, os nomes do adaptador de recurso devem ser únicos entre as versões da Process Application.

Se os nomes do adaptador de recurso não forem únicos, é registada uma mensagem informativa durante a implementação, para o alertar. Pode alterar, manualmente, os nomes dos adaptadores de recurso, após a implementação, utilizando a consola administrativa.

## Chamada para execução dinâmica adequada à versão

Pode configurar os componentes do fluxo de mediação para que encaminhem mensagens para os destinos finais que estejam dinamicamente determinados no tempo de execução. Quando criar o módulo de mediação, configura a procura do destino final para utilizar o encaminhamento adequado à versão.

Se utilizar o estilo IBM\_VRM (*<version>.<release>.<modification>*) para o Snapshot, pode exportar o ficheiro EAR da Process Application para WebSphere Service Registry and Repository (WSRR). Quando cria o módulo de mediação, configura, de seguida, a procura do destino final para utilizar o encaminhamento adequado à versão. Por exemplo, seleccione **Return endpoint matching latest compatible version of SCA module-based services (Repor destino final para corresponder à versão mais recente compatível dos serviços com base em módulo SCA)** a partir do **Match Policy field (campo Política de correspondência)** e selecciona **SCA** para **Binding Type (Tipo de associação)**.

As versões futuras da Process Application são implementadas no servidor e publicadas para WSRR e a procura do destino final do módulo de mediação chama, dinamicamente, a versão compatível mais recente do destino final de serviço.

Tenha em atenção que, como alternativa, pode definir o destino em SMOHeader e o valor pode ser transportado pela mensagem de pedido.

## Implementar Process Application com projectos e módulos Java

As Process Application podem conter módulos Java EE e projectos de Java personalizados. Quando co-implementar as aplicações, o módulo Java EE para cada versão da aplicação deve ser único.

Tenha em atenção que os módulos Java EE e os projectos de Java personalizados são implementados num servidor caso sejam implementados com um módulo SCA que tenha uma dependência declarada sobre estes. Se não seleccionar **Deploy with module (Implementar com módulo)** (que é a predefinição) quando declarar a dependência, necessita de implementar o módulo ou projecto manualmente.

## Implementar Process Applications com regras empresariais e selectores

Se estiver a implementar versões múltiplas de uma Process Application que inclua uma regra de negócio ou componente de selector, tenha em atenção ao modo como os metadados são utilizados pelas versões.

Os metadados dinâmicos para uma regra de negócio ou componente de selector são definidos no tempo de execução pelo nome do componente, espaço de nomes de destino do componente e tipo de componente. Se duas ou mais versões de uma Process Application que contenham uma regra de negócio ou selector forem implementadas para o mesmo ambiente em tempo de execução, partilharão a mesma lógica de regra (regra de negócio) ou metadados de encaminhamento (selector).

Para permitir a utilização, por parte de cada versão da regra de negócio ou componente de selector da Process Application, dos respectivos metadados dinâmicos (lógica de regra ou encaminhamento), reconstrua o espaço de nomes de destino para que seja único para cada versão da Process Application.

## Objectos de configuração

Pode utilizar os comandos AdminConfig (wsadmin) na ferramenta de administração da linha de comandos do WebSphere (wsadmin) para aceder e modificar propriedades de bases de dados e segurança no IBM Business Process Manager.

O termo *configuration object* refere um objecto que é acedido através dos comandos AdminConfig wsadmin. Para obter mais informações, consulte Security configuration properties.

## Arquitectura de implementação

A arquitectura de implementação do IBM Business Process Manager consiste em processos de software denominados servidores, unidades topológicas referenciadas como nós e células e o repositório de configuração utilizado para armazenar informações de configuração.

### Células

No IBM Business Process Manager, as *células* são agrupamentos lógicos de um ou mais nós numa rede distribuída.

Uma célula é um conceito de configuração, uma forma para os administradores associarem logicamente nós, uns com os outros. Os administradores definem os nós que formam uma célula, de acordo com o critério específico indicado para cada ambiente organizacional.

Os dados de configuração administrativa são guardados em ficheiros XML. Uma célula retém os ficheiros de configuração principais para cada um dos servidores em cada um dos nós da célula. Cada nó e servidor possuem também ficheiros de configuração local próprios. As alterações efectuadas a um nó local ou a um ficheiro de configuração do servidor, são temporárias, caso o servidor faça parte da célula. Enquanto estiverem em vigor, as alterações locais substituem as configurações das células. As alterações ao servidor principal e ficheiros de configuração de nó principais efectuadas ao nível da célula, substituem quaisquer alterações temporárias ao nó durante a sincronização dos documentos de configuração da célula com os nós. A sincronização durante eventos designados, tal como quando o servidor é iniciado.

### Servidores

Os servidores fornecem a funcionalidade nuclear do IBM Business Process Manager. Os Process Servers expandem ou aumentam capacidade de um servidor de aplicações para gerir módulos da Arquitectura de Componente de Serviço (SCA, Service Component Architecture). Outros servidores (gestores de implementação e agentes de nós), são utilizados para gerir Process Servers.

Um Process Server pode ser um *servidor autónomo* ou um *servidor gerido*. Um servidor gerido pode opcionalmente ser membro de um *conjunto*. Um conjunto de servidores geridos, conjunto de unidades de servidores e outros middleware é denominado um *ambiente de implementação*. Num ambiente de implementação, cada um dos servidores geridos ou conjuntos de unidades é configurado para uma função específica no ambiente de implementação (por exemplo, sistema central de destino, sistema central do módulo de aplicações ou servidor Common Event Infrastructure). Um servidor autónomo é configurado para fornecer todas as funções requeridas.

Os servidores fornecem o ambiente em tempo real para módulos SCA, para os recursos que são utilizados por esses módulos (origens de dados, especificações de activação e destinos JMS) e para recursos fornecidos pela IBM (destinos de mensagens, contentores Business Process Choreographer e servidores Common Event Infrastructure).



Um *agente do nó* é um agente administrativo que representa um nó para o sistema e gere os servidores nesse nó. Os agentes de nós supervisionam os servidores num sistema central e encaminham pedidos administrativos para os servidores. O agente do nó é criado quando um nó está federado a um gestor de implementação.

Um *gestor de implementação* é um agente administrativo que fornece uma vista de gestão centralizada para múltiplos servidores e conjuntos de unidades.

Um servidor autónomo é definido por um perfil autónomo; um gestor de implementação é definido por um perfil de gestor de implementação; os servidores geridos são criados num *nó gerido*, que é definido por um perfil de nós geridos.

### **Servidores autónomos:**

Um servidor autónomo fornece um ambiente para implementação de módulos SCA num processo de servidor. Este processo de servidor inclui, mas não se limita a, uma consola administrativa, um destino de implementação, um suporte de serviço de mensagem, o gestor de regras de processo empresarial e um servidor Common Event Infrastructure (infraestrutura de evento comum).

Um servidor autónomo é simples de configurar e apresenta uma consola Quick Start a partir da qual poderá iniciar e parar o servidor, abrir a galeria de exemplos e a consola administrativa. Se instalar os exemplos do IBM Business Process Manager e, em seguida, abrir a galeria de exemplos, é aplicada uma solução de exemplo no servidor autónomo. É possível explorar os recursos utilizados para este exemplo na consola administrativa.

É possível implementar soluções próprias a um servidor autónomo, mas um servidor autónomo não pode fornecer a capacidade, escalabilidade ou robustez requerida num ambiente de produção. É melhor utilizar um ambiente de implementação de rede para o ambiente de produção.

É possível começar com um servidor autónomo e posteriormente incluí-lo num ambiente de implementação de rede, federando-o à célula do gestor de implementação, *desde que não tenham sido federados quaisquer outros nós a essa célula*. Não é possível federar múltiplos servidores autónomos numa célula. Para federar o servidor autónomo, utilize a consola administrativa do gestor de implementação ou o comando **addNode**. O servidor autónomo não deverá estar em execução ao federá-lo utilizando o comando **addNode**.

Um servidor autónomo é definido por um perfil do servidor autónomo.

### **Conjuntos:**

Os conjuntos de unidades são grupos de servidores geridos em conjunto e participam na gestão da carga de trabalho.

Um conjunto pode conter nós ou servidores de aplicações individuais. Um nó é normalmente um sistema informático físico com um endereço de IP de sistema central distinto que executa um ou mais servidores de aplicações. Os conjuntos de unidades podem ser agrupados sob a configuração de uma célula, que associa logicamente muitos servidores e conjuntos de unidades com configurações e aplicações diferentes entre eles, dependendo do critério do administrador e dos requisitos de cada ambiente organizacional.

Os conjuntos são responsáveis pelo equilíbrio da carga de trabalho entre os servidores. Os servidores que fazem parte de um conjunto são denominados membros do conjunto. Durante a instalação de uma aplicação num conjunto de unidades, a aplicação é automaticamente instalada em cada um dos membros do conjunto de unidades.

Como cada membro de conjunto de unidades contém as mesmas aplicações, é possível distribuir tarefas cliente de acordo com as capacidades de cada máquina atribuindo cargas a cada um dos servidores.



A atribuição de cargas aos servidores num conjunto melhora o desempenho e redireccionamento em caso de falha. As tarefas são atribuídas aos servidores que possuem a capacidade para executar as operações de tarefas. Se um dos servidores não estiver disponível para executar a tarefa, esta é atribuída a outro membro do conjunto. Esta capacidade de redistribuição tem vantagens óbvias sobre a execução de um único servidor de aplicações que pode ficar sobrecarregado caso sejam efectuados demasiados pedidos.

## Perfis

Um perfil define um ambiente em tempo de execução único, com ficheiros de comando, ficheiros de configuração e ficheiros de registo separados. Os perfis definem três tipos diferentes de ambientes em sistemas IBM Business Process Manager: servidor autónomo, gestor de implementação e nó gerido.

Ao utilizar perfis, é possível ter mais do que um ambiente em tempo de execução no sistema, sem ter de instalar múltiplas cópias dos ficheiros binários do IBM Business Process Manager.

Utilize o Utilitários de linha de comandos BPMConfig para criar perfis do IBM BPM. O utilitário da linha de comandos **manageprofiles** ou a respectiva interface gráfica do utilizador, a Ferramenta de Gestão de Perfis (PMT, Profile Management Tool), pode ser utilizada como uma forma alternativa para criar um gestor de implementação ou os perfis de nós geridos. O PMT já não é suportado para a criação de perfis autónomos.

**Nota:** nas plataformas distribuídas, cada perfil tem um nome único. Em z/OS, todos os perfis são nomeados como “predefinidos”; não pode mudar o nome, editar, copiar ou eliminar perfis em z/OS.

## Tipos de perfis

Os seguintes tipos de perfis do IBM BPM estão disponíveis com o IBM Business Process Manager V8.5:

### Perfil autónomo do IBM BPM

O perfil autónomo define os servidores autónomos com capacidades e funcionalidades específicas para as configurações do IBM BPM Express. Pode criar o perfil autónomo através do modelo do perfil BPM/BpmServer, que só é suportado no IBM BPM Express.

### Perfil do gestor de implementação do IBM BPM

O perfil do gestor de implementação define um gestor de implementação que fornece uma interface administrativa para um grupo lógico de servidores numa ou mais estações. Pode criar o perfil do gestor de implementação através do modelo do perfil BPM/BpmDmgr, que só é suportado no IBM BPM Standard e no IBM BPM Advanced.

### Perfil do nó gerido do IBM BPM

O perfil do nó gerido define um nó gerido quando o nó é federado para um gestor de implementação. Pode criar um perfil de nó gerido através do modelo de perfil BPM/BpmNode, que só é suportado no IBM BPM Standard e no IBM BPM Advanced.

Os tipos de perfil específicos estão disponíveis com cada configuração do IBM BPM.

*Tabela 19. Tipos de perfis disponíveis do IBM BPM*

Configuração do IBM BPM	Tipos de perfis		
	Autónomas	Gestor de implementação	Nó gerido
IBM BPM Express	Sim	Não	Não
IBM BPM Standard	Não	Sim	Sim
IBM BPM Advanced	Não	Sim	Sim
Ambiente de Teste de Unidade (UTE, Unit Test Environment) para o IBM Integration Designer	Sim	Opcional	Opcional

## O directório do perfil

Todos os perfis no sistema possuem um directório próprio que contém todos os ficheiros correspondentes. A localização do directório do perfil é especificada durante a criação do perfil. Por predefinição, é no directório `profiles` que o IBM Business Process Manager é instalado. Por exemplo, o perfil `Dmgr` está localizado em `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

## O perfil predefinido

O primeiro perfil criado numa instalação do IBM Business Process Manager é o *perfil predefinido*. O perfil predefinido é o destino predefinido para comandos emitidos a partir do directório `bin` no directório onde foi instalado o IBM Business Process Manager. Se existir apenas um perfil no sistema, todos os comandos operam nesse perfil. Se for criado outro perfil, poderá torná-lo predefinido.

**Nota:** O perfil predefinido não é necessariamente um perfil cujo nome é “default”.

## Aumentar perfis

Se já tiver criado um perfil de gestor de implementação, um perfil de nós geridos ou um perfil de servidor autónomo para o WebSphere Application Server Network Deployment, pode *aumentá-lo* para suportar o IBM Business Process Manager para além da função existente. Para aumentar um perfil, é necessário instalar primeiro o IBM Business Process Manager. Em seguida, utilize o utilitário da linha de comandos **manageprofiles** para aumentar um perfil autónomo ou utilize a Ferramenta de Gestão de Perfis (PMT, Profile Management Tool) ou o utilitário da linha de comandos **manageprofiles** para aumentar um gestor de implementação ou um perfil de nós geridos.

**Importante:** Num ambiente de implementação da rede, tem de aumentar primeiro o perfil do gestor de implementação e, em seguida aumentar os perfis de nós geridos.

**Restrição:** Não pode aumentar um perfil de gestor de implementação ou autónomo onde o registo de utilizadores VMM da WebSphere predefinido foi alterado, por exemplo, para utilizar LDAP.

## Gestores de implementação

Um gestor de implementação é um servidor que gere operações para um grupo lógico ou célula de outros servidores. O gestor de implementação é a localização central para administrar os servidores e conjuntos.

Quando é criado um ambiente de implementação, o perfil do agente de implementação é o primeiro perfil a ser criado. Cada ambiente de implementação que cria tem uma consola de Quick Start, a partir da qual poderá iniciar e parar o gestor de implementação e iniciar a respectiva consola administrativa. Pode utilizar a consola administrativa do gestor de implementação para gerir os servidores e conjuntos de unidades da célula. Isto inclui configurar servidores e conjuntos de unidades, adicionar servidores a conjuntos de unidades, iniciar a parar servidores e conjuntos de unidades e, implementar módulos SCA.

Embora o gestor de implementação seja um tipo de servidor, não é possível implementar módulos no próprio gestor de implementação.

## Nós

Um *nó* é um agrupamento lógico de servidores geridos.

Um nó corresponde normalmente a um sistema informático lógico ou físico com um endereço de IP do sistema central distinto. Os nós não podem abranger múltiplos computadores. Os nomes dos nós são normalmente idênticos ao nome do sistema central do computador.

Os nós da topologia de implementação de rede podem ser geridos ou não geridos. Um nó gerido possui um processo de agente do nó que gera a configuração e os servidores. Os nós não geridos não possuem um agente do nó.

### **Nós geridos:**

Um *nó gerido* é um nó federado a um gestor de implementação, contém um agente de nó e pode conter servidores geridos. Num nó gerido, é possível configurar e executar servidores geridos.

Os servidores configurados num nó gerido constituem os recursos do ambiente de implementação. Estes servidores são criados, configurados, iniciados, interrompidos, geridos e eliminados utilizando a consola administrativa do gestor de implementação.

Um nó gerido possui um agente de nó que gere todos os servidores de um nó.

Quando um nó é federado, é criado automaticamente um processo de agente do nó. Este agente do nó deverá estar em execução para ser possível gerir a configuração do perfil. Por exemplo, quando são executadas as tarefas seguintes:

- Iniciar e parar processos no servidor.
- Sincronizar dados de configuração no gestor de implementação com a cópia no nó.

No entanto, o agente do nó não necessita de estar em execução para que as aplicações possam ser executadas ou para configurar recursos no nó.

Um nó gerido poderá conter um ou mais servidores, geridos por um gestor de implementação. É possível implementar soluções nos servidores de um nó gerido, mas o nó gerido não possui uma galeria de aplicações de exemplo. O nó gerido é definido por um perfil de nó gerido e possui uma consola Quick Start.

### **Nós não geridos:**

Um nó não gerido não possui um agente do nó para gerir os servidores.

Nós não gerido na topologia Implementação de rede pode possuir definições de servidor tais como servidores da Web, mas não definições de application server. Os nós não geridos nunca podem ser federados. Isto é, um agente do nó nunca pode ser adicionado a um nó não gerido. Um servidor autónomo é outro tipo de nó não gerido. O gestor de implementação não pode gerir este servidor autónomo porque é desconhecido para a célula. Um servidor autónomo pode ser federado. Quando é federado, é criado automaticamente um agente do nó. O nó torna-se um nó gerido na célula.

### **Agentes de nós**

Os agentes de nós são agentes administrativos que encaminham pedidos administrativos para os servidores.

Um agente do nó é um servidor que é executado em todos os sistemas centrais que participa na configuração da Implementação de Rede (Network Deployment). Trata-se de um agente puramente administrativo e não está envolvido nas funções das aplicações de servidor. Um agente do nó aloja também outras funções administrativas importantes, tais como serviços de transferência de ficheiros, sincronização da configuração e supervisão do desempenho.

### **Considerações de nome para perfis, nós, servidores, sistemas centrais e células**

Este tópico discute termos reservados e questões a considerar ao dar um nome a um perfil, nó, servidor, sistema central e célula (se tal for aplicável). Este tópico aplica-se a plataformas distribuídas.

## Considerações de nomes de perfil

O nome de perfil pode ser qualquer nome único com as seguintes restrições. Não utilizar qualquer dos seguintes caracteres ao dar um nome a um perfil:

- Espaços
- Caracteres especiais que não são permitidos no nome de um directório no sistema operativo, tal como \*, & ou ?.
- Barras (/) ou barras invertidas (\)

São permitidos caracteres de duplo byte.

**Windows** **Considerações sobre caminho do directório:** o caminho do directório de instalação deverá ser inferior ou igual a 60 caracteres. O número de caracteres do *caminho\_directório\_perfis\nome\_perfil* directório deverá ser inferior ou igual a 80 caracteres.

**Nota:** Utilize a convenção de nomenclatura de caminho abreviada quando criar um perfil num ambiente Windows para evitar a limitação do comprimento do caminho de 255 caracteres do Windows.

## Considerações de nomes de nó, servidor, sistema central e célula

**Nomes reservados:** evitar utilizar nomes reservados como valores de campos. A utilização de nomes reservados pode provocar resultados imprevisíveis. As palavras seguintes são reservadas:

- células
- nós
- servidores
- conjuntos
- aplicações
- implementações

**Descrição dos campos nas páginas Nós e Sistemas Centrais e Nó, Sistema Central e Nomes de Células:** Utilize as directrizes de nomenclatura adequadas quando cria perfis.

- Perfis do servidor autónomo
- Perfis do gestor de implementação
- Perfis do nó gerido

Tabela 20. directrizes de nomenclatura para perfis de servidores autónomos

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<b>Linux</b> <b>UNIX</b> <b>Windows</b> <i>shortHostName</i> Nó <i>NodeNumber</i> onde: <ul style="list-style-type: none"><li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li><li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li></ul>	Evitar a utilização dos nomes reservados.	Seleccionar qualquer nome pretendido. Se for planeada a criação de mais do que um servidor no sistema, deverá ser utilizado um nome único para ajudar a organizar a instalação.
Nome do servidor	<b>Linux</b> <b>UNIX</b> <b>Windows</b> servidor1	Utilize um nome único para o servidor.	Nome lógico para o servidor.

Tabela 20. directrizes de nomenclatura para perfis de servidores autónomos (continuação)

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows</p> <p>O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede.	Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.

Tabela 21. directrizes de nomenclatura para perfis de gestores de implementação

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<p>Linux      UNIX</p> <p>Windows</p> <p><i>shortHostName</i> Gestor de Célula<i>Node Number</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li> </ul>	Utilize um nome único para o gestor de implementação. Evitar a utilização dos nomes reservados.	O nome é utilizado para administração na célula do gestor de implementação.
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows</p> <p>O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede. Evitar a utilização dos nomes reservados.	Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.

Tabela 21. directrizes de nomenclatura para perfis de gestores de implementação (continuação)

Nome do campo	Valor predefinido	Restrições	Descrição
Nome da célula	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i></p> <p>Célula <i>CellNumber</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>CellNumber</i> é um número sequencial com início em 01.</li> </ul>	<p>Utilize um nome único para a célula do gestor de implementação. O nome da célula deverá ser único em todas as ocorrências em que o produto está em execução na mesma estação de trabalho física ou conjunto de unidades de estações de trabalho, tal como o Sysplex. Além disso, um nome de célula deverá ser único em qualquer circunstâncias em que seja necessária conectividade de rede entre entidades, seja entre as células ou a partir de um cliente que tenha de comunicar com cada uma das células. Os nomes das células também têm de ser únicos se os espaços de nomes correspondentes vão ser federados. Caso contrário, podem ocorrer sintomas de excepção, tal como o <code>javax.naming.NameNotFoundException</code>, em que será necessário criar células com nomes exclusivos.</p>	<p>Todos os nós federados tornam-se membros da célula do gestor de implementação, à qual foi dado o nome na página Nó, Sistema Central e Nomes de Célula da Ferramenta de Gestão de Perfis (Profile Management Tool).</p>

Tabela 22. Directrizes de nomenclatura para perfis de nós geridos

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i></p> <p>Nó <i>NodeNumber</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li> </ul>	<p>Evitar a utilização dos nomes reservados.</p> <p>Utilize um nome único na célula do gestor de implementação.</p>	<p>O nome é utilizado para administração na célula do gestor de implementação à qual é adicionado o perfil do nó gerido. Utilize um nome único na célula do gestor de implementação.</p>
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows      O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	<p>Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede.</p>	<p>Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.</p>

### Considerações sobre o nome do sistema central:

O nome do sistema central é o nome da rede da estação de trabalho física em que o nó está instalado. O nome do sistema central deverá ser resolvido para o nó de uma rede física no servidor. Quando existem múltiplos cartões de rede no servidor, o nome do sistema central ou o endereço de IP deverá ser resolvido para um dos cartões de rede. Os nós remotos utilizam o nome do sistema central para ligar e para comunicar com este nó.

IBM Business Process Manager é compatível com os Protocolos da Internet versão 4 (IPv4) e versão 6 (IPv6). Poderá utilizar qualquer dos formatos sempre que for necessário introduzir endereços de IP na consola administrativa ou em outro local. Observe que, se o IPv6 estiver implementado no sistema será necessário introduzir o endereço de IP no formato IPv6 e reciprocamente, se o IPv6 ainda não estiver disponível, deverá introduzir o endereço IP no formato IPv4. Para mais informações sobre IPv6 consulte a seguinte descrição: IPv6.

As directrizes seguintes podem ajudar a determinar o nome do sistema central apropriado para a estação de trabalho:

- Selecione um nome do sistema central que possam ser contactado por outras estações de trabalho na rede.
- Não utilizar o identificador genérico, sistema central local (localhost), para este valor.
- Não proceder à instalação de IBM Business Process Manager produtos num servidor com um nome do sistema central que utilize caracteres do conjunto de caracteres de duplo byte (DBCS). A utilização de caracteres DBCS no nome do sistema central, não é suportada.
- Evite utilizar o carácter de sublinhado ( \_ ) nos nomes dos servidores. Os padrões da Internet ditam que os nomes dos domínios estão em conformidade com os requisitos de nome do sistema central descritos no Internet Official Protocol Standards RFC 952 e RFC 1123. Os nomes do domínio devem conter apenas letras (maiúsculas ou minúsculas) e dígitos. Os nomes dos domínios podem também conter caracteres de traço ( - ) desde que os traços não estejam no fim do nome. Os caracteres de sublinhado ( \_ ) não são suportados no nome do sistema central. Se tiver instalado IBM Business Process Manager num servidor com um carácter de sublinhado no nome do servidor, terá de aceder ao servidor através do respectivo IP até que seja mudado o nome.

Caso sejam definidos nós coexistindo no mesmo computador com endereços de IP únicos, deverá ser definido cada um dos endereços de IP numa tabela de procura num servidor de nomes de domínio (DNS). Os ficheiros de configuração para servidores não fornecem a resolução do nome do domínio para múltiplos endereços IP numa estação de trabalho com um único endereço de rede.

O valor especificado para o nome do sistema central é utilizado como valor da propriedade hostName nos documentos de configuração. Especifique o valor do nome do sistema central num dos formatos seguintes:

- cadeia de nome do sistema central dos servidores de nomes de domínio (DNS), como xmachine.manhattan.ibm.com
- A cadeia predefinida do nome do sistema central do DNS, tal como xmachine
- Endereços de IP numéricos, como 127.1.255.3

O nome do sistema central do DNS totalmente qualificado tem a vantagem de ser totalmente inequívoco e flexível. Existe a flexibilidade de alterar o endereço de IP actual para o sistema central sem ter de alterar a configuração do servidor. Este valor para o nome do sistema central é particularmente útil se planear alterar o endereço de IP frequentemente se utilizar o protocolo DHCP para atribuir endereços de IP. Uma desvantagem deste formato é a dependência do DNS. Se o DNS não estiver disponível, então a conectividade fica comprometida.

O nome curto do sistema central também é resolvido de forma dinâmica. O formato de nome curto tem a capacidade acrescida de poder ser redefinido no ficheiro dos sistemas centrais locais para que o sistema possa executar o servidor mesmo quando está desligado da rede. Para ser possível executar desligado, deverá definir o nome curto como 127.0.0.1 (circuito fechado local) no ficheiro dos sistemas centrais. A



desvantagem do formato de nome curto e a dependência do DNS para o acesso remoto. Se o DNS não estiver disponível, então a conectividade fica comprometida.

Um endereço de IP numérico tem a vantagem de não necessitar da resolução de nomes através do DNS. Um nó remoto pode ligar-se ao nó nomeado com um endereço de IP numérico sem a necessidade do DNS estar disponível. A desvantagem deste formato é o facto do endereço de IP numérico ser fixo. A definição da propriedade hostName (nome do sistema central) terá de ser alterada nos documentos de configuração sempre que o endereço de IP da estação de trabalho for alterado. Por esta razão, não deverá utilizar um endereço de IP numérico se utilizar DHCP ou se alterar regularmente o endereço de IP. Outra desvantagem deste formato é não poder utilizar o nó se o sistema central estiver desligado da rede.

## **BPMN 2.0**

As definições de processo empresarial IBM Business Process Manager suportam a subclasse executável comum da classe de conformidade de modelação de processos BPMN 2.0, que lida com modelos que pode executar.

O BPMN (Business Process Model and Notation) é a norma fundamental dos processos no IBM Process Designer e IBM Process Center. Os diagramas do BPD (Business Process Definition) baseiam-se na especificação do BPMN. Este tópico apresenta alguns dos modos de aplicação do BPMN 2.0 no IBM Business Process Manager. Para obter informações detalhadas sobre o BPMN, consulte a página de Especificação do BPMN em <http://www.bpmn.org/>.

O IBM Business Process Manager suporta os seguintes tipos de tarefas do BPMN 2.0:

- Nenhuma (tarefa abstracta na especificação do BPMN 2.0)
- Tarefa do sistema (tarefa do serviço na especificação do BPMN 2.0)
- Tarefa de utilizador
- Script
- Tarefa de decisão (tarefa de regra de negócio na especificação do BPMN 2.0)

Os eventos de mensagens intermédias do IBM BPM fornecem funções semelhantes às tarefas de envio e recepção do BPMN.

## **Notação do BPMN 2.0**

Iniciar em V7.5.1, os ícones de tarefa Process Designer BPMN 2.0 nos diagramas BPD são recolhidos numa paleta simplificada e apresentados em diagramas em processamento. Os ícones mostram se a sua actividade é uma tarefa de sistema, uma tarefa de utilizador, uma tarefa de decisão, um script ou um processo de ligação. As actividades em modelos criados em versões anteriores também apresentam os tipos de tarefas BPMN 2.0 apropriados e os ícones de tarefa quando os visualiza na versão 7.5.1 ou posterior.

## **Actividades e tarefas**

Existem algumas alterações de terminologia das versões anteriores do Process Designer. Algumas dessas alterações envolvem tipos de actividades cujo nome foi alterado.

- As actividades do serviço (automáticas) são agora tarefas do sistema.
- As actividades do serviço (tarefas) numa pista sem ser do sistema são agora tarefas do utilizador.
- As actividades do serviço (tarefas) numa pista do sistema são agora tarefas de decisão se referenciarem um serviço de decisão.
- As actividades do serviço (tarefas) numa pista do sistema são agora tarefas do sistema se referenciarem qualquer tipo de serviço diferente de uma decisão.
- As actividades de Javascript são agora tarefas de script.
- As actividades de processos imbricados são agora processos ligados.



- As actividades externas de versões anteriores do Process Designer estão disponíveis como implementações externas de tarefas do utilizador ou do sistema.

## Portas de ligação

Não existem alterações de notação às portas de ligação de versões anteriores. No entanto, existem três alterações de terminologia. A porta de ligação de decisão é agora a *porta de ligação exclusiva*, a divisão simples ou a porta de ligação de junção é agora a *porta de ligação paralela* e a divisão condicional ou porta de ligação de junção é agora a *porta de ligação inclusiva*.

Também existe um novo tipo de porta de ligação, a *porta de ligação de eventos*. Uma porta de ligação de eventos representa um ponto de desvio num processo em que os caminhos alternativos que seguem a porta de ligação se baseiam nos eventos ocorridos, em vez de na avaliação de expressões que utilizam dados do processo (como numa porta de ligação exclusiva ou inclusiva). Um evento específico, normalmente a recepção de uma mensagem, determina o caminho que será seguido.

## Eventos de não interrupção

O BPMN 2.0 adicionou notação para eventos de não interrupção. Por predefinição, um evento limite interrompe a actividade anexada. Quando o evento é activado, a actividade pára e o símbolo continua o fluxo de sequência de saída do evento. Se o evento for definido como sendo de não interrupção, quando o evento é activado, a actividade anexa continua a ser executada em paralelo e é criado um novo símbolo que é passado pelo fluxo de sequência de saída do evento. O limite do evento é alterado para uma linha tracejada para eventos de não interrupção.

Os eventos intermédios anexados a actividades são eventos intermédios de interrupção se fecharem as actividades que lhes são anexas ou eventos intermédios de não interrupção se não fecharem as actividades que lhe são anexas.

## Evento inicial

A especificação do BPMN permite aos modelos de processo omitir símbolos de eventos de início e fim. O Process Designer requer que os modelos de processo utilizem eventos de início e paragem.

Existem vários tipos de eventos de início disponíveis no Process Designer:

### processos

- nenhum
- mensagem
- ad hoc

### subprocessos

- nenhum

### subprocessos do evento

- erro
- mensagem
- temporizador

Pode alterar o tipo de um evento de início editando as propriedades do evento. Pode possuir vários eventos de início de mensagem num processo, mas só pode utilizar um evento que não seja de início.

## Eventos de fim

Estão disponíveis quatro tipos de eventos de fim: *mensagem*, *término*, *erro* e *nenhum*. Pode alterar o tipo de um evento de fim.

Quando um processo ascendente chama um processo descendente e este último executa uma acção de evento terminar, o processo descendente pára e o processo ascendente continua para os passos seguintes.

## Subprocessos

A especificação do BPMN define dois tipos de subprocessos, incorporados e reutilizáveis. Pode reactivar ambos os tipos no Process Designer. Os subprocessos incorporados são denominados simplesmente *subprocessos* no Process Designer e são novos na versão 7.5.1. O subprocesso reutilizável do BPMN é denominado *processo ligado* no Process Designer.

O processo contém um subprocesso que é uma forma de agrupar passos de processo para reduzir a complexidade e os ecos parasitas do diagrama. Os subprocessos contraem vários passos numa actividade. O subprocesso só pode ser visualizado pelo processo no qual é definido. Um subprocesso existe no âmbito do respectivo chamador e tem acesso a todas as variáveis nesse ambiente. Não existe nenhum parâmetro a ser transmitido de e para o subprocesso incorporado.

Separado do subprocesso e do processo ligado, Process Designer possui um subprocesso de evento, que é um subprocesso especializado utilizado para tratamento de eventos. Não está ligado a outras actividades através do fluxo de sequência e ocorre apenas se o evento de início for activado.

## Processos ligados

Um subprocesso reutilizável do BPMN é denominado *processo ligado* no Process Designer. É um processo criado fora do processo actual que pode ser chamado pelo processo actual. É reutilizável, porque as outras definições do processo também podem chamar este processo. O processo ligado define os parâmetros de entrada e saída e não tem acesso ao âmbito ou ambiente do chamador. O processo ligado é semelhante ao processo imbricado disponível nas versões anteriores; não existe qualquer alteração no comportamento da actividade. Os processos imbricados anteriores passam a processos ligados. O processo ligado é semelhante a um subprocesso com um pequeno limite e está realçado na janela do Inspector.

## Ciclos

O BPMN fornece o conceito de uma actividade que pode ser repetida. A actividade pode ser atómica, o que significa que a actividade é repetida ou pode ser um subprocesso, encapsulando um conjunto de passos repetidos. Se expandir a actividade repetida, poderá ver as actividades contidas executadas repetidamente. A condição é sempre avaliada no início de cada iteração do ciclo. Não existe capacidade de avaliação no final de cada iteração do ciclo.

IBM Business Process Manager possui um *ciclo de várias ocorrências*, que é executado um determinado número de vezes com as actividades contidas executadas em sequência ou em paralelo.

## Importar processos não BPMN

É possível importar modelos criados no IBM WebSphere Business Modeler e utilizá-los no Process Designer. Para mais informações sobre a importação BPMN 2.0, consulte Mapeamento de IBM WebSphere Business Modeler elementos para IBM Business Process Manager construção. Também pode importar modelos BPMN 2.0 criados em WebSphere Business Compass da IBM, Rational Software Architect, ou outros ambientes de modelação.

## Definições de Processo Empresarial (BPDs)

Para modelar um processo no IBM Process Designer, tem de criar uma definição de processo empresarial (BPD). A definição de business process pode ser baseado num modelo BPMN importado.

Uma BPD é um modelo reutilizável de um processo, que define aquilo que é comum a todas as instâncias do tempo de execução desse modelo de processo. Uma BPD tem de conter um evento de

início, um evento de conclusão, pelo menos uma pista, e uma ou mais actividades. Consulte "Convenções de nomenclatura do Process Designer da IBM" nas ligações relacionadas para obter mais detalhes sobre as limitações de caracteres aplicáveis a BPDs.

Uma Definição de Processo Empresarial (BPD) tem de incluir uma pista para cada sistema ou grupo de utilizadores que participe num processo. Os controlos podem ser de participante ou de sistema. No entanto, é possível criar uma BPD que agrupe as actividades de um grupo e de um sistema numa única pista, se preferir. Consulte "Criar uma definição de processo empresarial (BPD)" nas ligações relacionadas para obter informações sobre como criar uma BPD.

É possível nomear qualquer pessoa ou grupo específico como responsável pelas actividades que ocorrem numa pista de participante. Cada pista que cria é atribuída ao grupo Todos os Utilizadores por predefinição. É possível utilizar este grupo predefinido para executar e testar a BPD no Inspector. O grupo Todos os utilizadores inclui todos os utilizadores que são membros do grupo de segurança **tw\_allusers**, que é um grupo de segurança especial que inclui automaticamente todos os utilizadores no sistema.

Uma pista de sistema contém actividades geridas por um sistema IBM Process Center específico. Cada actividade requer uma implementação, que irá definir a actividade e as propriedades da tarefa. Durante a implementação, um programador cria um serviço ou escreve o JavaScript necessário para concluir as actividades de uma pista de sistema. Consulte "Compreender os tipos de serviço" nas ligações relacionadas para obter mais informações sobre os serviços.

Para cada BPD que criar, terá de declarar variáveis para capturar dados de negócio que são passados de actividade para actividade no processo. Consulte "Gerir e mapear variáveis" nas ligações relacionadas para aprender mais sobre a implementação de variáveis.

É possível também adicionar eventos a uma BPD. Os eventos no IBM BPM podem ser activados pela ultrapassagem de um prazo, uma excepção ou a chegada de uma mensagem. O activador que pretende determina o tipo de evento que escolhe implementar. Para obter informações mais detalhadas sobre os tipos de eventos disponíveis e respectivos activadores, consulte "Eventos de modelação".

## Associações

No núcleo de uma arquitectura orientada para o serviço está o conceito de um *serviço*, uma unidade de funcionalidade realizada por uma interacção entre dispositivos de processamento. Uma *exportação* define a interface externa (ou ponto de acesso) de um módulo, de forma que os componentes da área de comunicações partilhada (SCA, Service Component Architecture) dentro do módulo possam oferecer os seus serviços a clientes externos. Uma *importação* define uma interface para serviços fora de um módulo, para que os serviços possam ser chamados dentro do módulo. Utiliza *associações* específicas do protocolo com importações e exportações para especificar os meios de transportar os dados para dentro ou para fora do módulo.

## Exportações

Os clientes externos podem chamar componentes SCA num módulo de integração através de uma variedade de protocolos (tais como HTTP, JMS, MQ e RMI/IIOP) com dados numa variedade de formatos (tais como XML, CSV, COBOL e JavaBeans). As exportações são componentes que recebem estes pedidos de fontes externas e que, em seguida, chamam IBM Business Process Manager componentes utilizando o modelo de programação SCA.

Por exemplo, na imagem seguinte, uma exportação recebe um pedido através do protocolo de HTTP de uma aplicação cliente. Os dados são transformados num objecto de negócio, o formato utilizado pelo componente SCA. O componente é, em seguida, chamado com o objecto de dados.

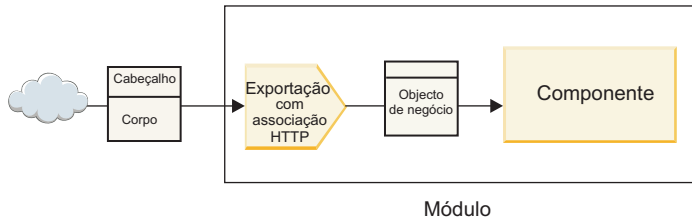


Figura 1. Uma exportação com associação HTTP

## Importações

Um componente SCA pode querer chamar um serviço externo não SCA que espera dados num formato diferente. Uma importação é utilizada pelo componente SCA para chamar o serviço externo, utilizando o modelo de programação SCA. A importação chama, em seguida, o serviço destino na forma esperada pelo serviço.

Por exemplo, na seguinte imagem é enviado um pedido de um componente SCA, pela importação, para um serviço externo. O objecto de negócio que é o formato utilizado pelo componente SCA, é transformado no formato esperado pelo serviço e o serviço é chamado.

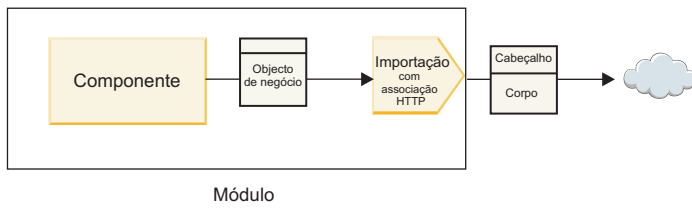


Figura 2. Uma importação com associação de HTTP

## Lista de associações

Utiliza Integration Designer para gerar uma associação para uma importação ou exportação e para configurar a associação. Os tipos de associações que estão disponíveis encontram-se na lista seguinte.

- SCA
 

A associação SCA, que é a predefinição, permite que o serviço comunique com serviços noutros módulos SCA. Utilize uma importação com uma associação SCA para aceder a um serviço noutro módulo SCA. Utilize uma exportação com uma associação SCA para oferecer um serviço a outros módulos SCA.
- Serviço da Web
 

Um serviço da web permite-lhe aceder a um serviço externo utilizando mensagens SOAP inter-operacionais e qualidades de serviço. Também pode utilizar associações do serviço da web para incluir anexos como parte da mensagem SOAP.

A associação do serviço da web pode utilizar um protocolo de transporte de SOAP/HTTP (SOAP acima de HTTP) ou SOAP/JMS (SOAP acima de JMS). Independentemente do transporte (HTTP ou JMS) utilizado para transmitir as mensagens SOAP, as associações do serviço da web processam interações de pedido/resposta de forma síncrona.
- HTTP
 

A associação de HTTP permite aceder a um serviço externo utilizando o protocolo HTTP, onde são utilizadas mensagens não SOAP ou onde é necessário o acesso directo de HTTP. Esta associação é utilizada quando está a trabalhar com serviços da web que se baseiam no modelo HTTP (ou seja, serviços que utilizam as conhecidas operações de interface HTTP, como GET, PUT, DELETE, etc.).
- Empresa JavaBeans (EJB)

As associações EJB permitem que os componentes SCA interajam com serviços fornecidos pela lógica de negócio Java EE a ser executada num servidor Java EE.

- EIS

A associação do Enterprise Information System (EIS), quando utilizada com um adaptador de recurso JCA, permite-lhe aceder a serviços num sistema de informação empresarial ou tornar os serviços disponíveis no EIS.

- Associações JMS

As associações de serviço de mensagens Java (JMS), JMS genérico e WebSphere MQ JMS (MQ JMS) são utilizadas para interações com sistemas de serviço de mensagens onde a comunicação assíncrona através de filas de mensagens é crítica quanto à fiabilidade.

Uma exportação com uma das associações JMS observa uma fila para a chegada de uma mensagem e envia de forma assíncrona a resposta, se existente, à fila de respostas. Uma importação com uma das associações JMS cria e envia uma mensagem à fila JMS e observa uma fila para a chegada da resposta, se existente.

- JMS

A associação JMS permite aceder ao fornecedor de JMS integrado no WebSphere.

- JMS genérico

A associação de JMS genérico permite aceder a um fornecedor de sistemas de serviço de mensagens que não pertence à IBM.

- MQ JMS

A associação MQ JMS permite aceder ao subconjunto JMS de um sistema de serviço de mensagens WebSphere MQ. Esta associação é utilizada quando o subconjunto JMS de funções for suficiente para a aplicação do utilizador.

- MQ

A associação WebSphere MQ permite comunicar com aplicações nativas MQ, trazendo-as ao contexto de arquitectura orientada para o serviço e fornecendo acesso a informação de cabeçalho específica de MQ. Utilize esta associação quando precisar de utilizar funções nativas de MQ.

## **Exportar e importar descrição geral de associação**

Uma exportação permite efectuar serviços num módulo de integração disponível a clientes externos e uma importação permite que os seus componentes da área de comunicações partilhada (SCA) num módulo de integração chamem serviços externos. A associação associada à exportação e importação especifica a relação entre mensagens de protocolo e objectos de negócio. Também especifica a forma como as operações e falhas são seleccionadas.

## **Fluxo de informações através de uma exportação**

Uma exportação recebe um pedido que se destina ao componente ao qual a exportação está ligada, através de um transporte específico determinado pela associação associada (por exemplo, HTTP).

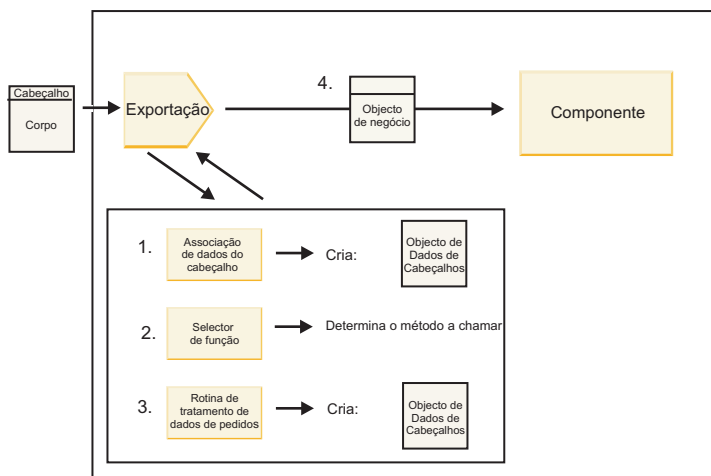


Figura 3. Fluxo de um pedido através da exportação para um componente

Quando a exportação recebe o pedido, aparece a seguinte sequência de eventos:

1. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho transforma o cabeçalho do protocolo num objecto de dados de cabeçalho.
2. O selector de função determina o nome do método nativo a partir da mensagem de protocolo. O nome do método nativo é mapeado pela configuração de exportação para o nome de uma operação na interface da exportação.
3. A rotina de tratamento de dados ou associação de dados de pedido no método transforma o pedido num objecto de negócio de pedido.
4. A exportação chama o método de componente com o objecto de negócio de pedido.
  - A associação de exportação HTTP, a associação de exportação do serviço da web, e a associação de exportação EJB chamam o componente da área de comunicações partilhada (SCA) de forma síncrona.
  - As associações de exportação JMS, Generic JMS, MQ JMS e WebSphere MQ chamam o componente SCA de forma síncrona.

Tenha em atenção que uma exportação pode propagar os cabeçalhos e as propriedades de utilizador que recebe através do protocolo, se a propagação de contexto estiver activada. Os componentes que estão ligados à exportação podem aceder a estes cabeçalhos e às propriedades de utilizador. Consulte o tópico “Propagation (Propagação)” no Information Center do WebSphere Integration Developer para obter mais informações.

Se esta for uma operação bidireccional, o componente devolve uma resposta.

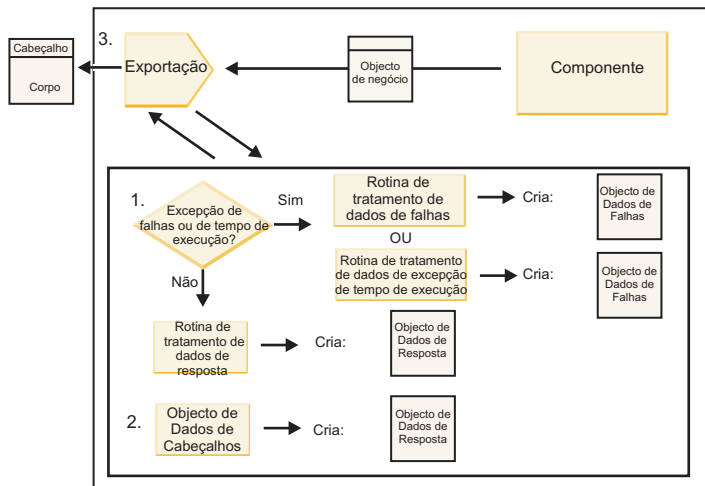


Figura 4. Fluxo de uma resposta de volta pela exportação

Aparece a seguinte sequência de passos:

1. Se uma mensagem de resposta normal for recebida pela associação de exportação, a rotina de tratamento de dados ou a associação de dados no método transforma o objecto de negócio numa resposta.

Se a resposta for uma falha, a rotina de tratamento de dados ou associação de dados de falha no método transforma a falha numa resposta de falha.

Apenas para associações de exportação HTTP, se a resposta for uma exceção de tempo de execução, a rotina de tratamento de dados de exceção de tempo de execução, se estiver configurada, é chamada.

2. Apenas para associações WebSphere, a associação de dados de cabeçalho transforma os objectos de dados de cabeçalho em cabeçalhos de protocolo.
3. A exportação envia a resposta de serviço através do transporte.

### Fluxo de informações através de uma importação

Os componentes enviam pedidos a serviços fora do módulo, utilizando uma importação. O pedido é enviado através de um transporte específico determinado pela associação associada.

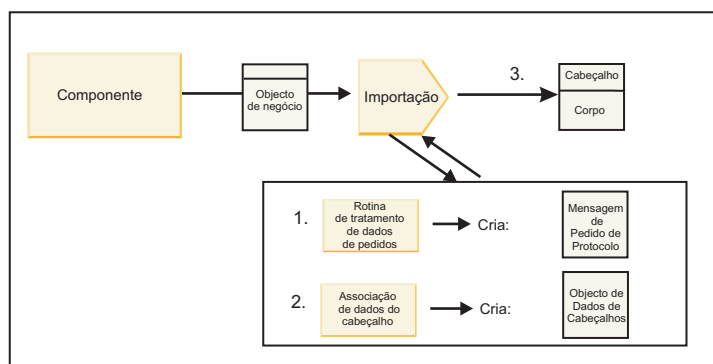


Figura 5. Fluxo de um componente através da importação para um serviço

O componente chama a importação com um objecto de negócio de pedido.

**Nota:**

- A associação de importação de HTTP, a associação de importação do serviço da web e a associação de importação de EJB devem ser chamadas de forma síncrona pelo componente de chamada.
- As associações de importação JMS, Generic JMS, MQ JMS e WebSphere MQ devem ser chamadas de forma assíncrona.

Depois do componente chamar a importação, aparece a seguinte sequência de eventos:

1. A rotina de tratamento de dados ou associação de dados de pedido no método transformam o objecto de negócio de pedido numa mensagem de pedido de protocolo.
2. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho no método define o objecto de negócio de cabeçalho no cabeçalho do protocolo.
3. A importação chama o serviço com o pedido de serviço através do transporte.

Se esta for uma operação bidireccional, o serviço devolve uma resposta e é apresentada a seguinte sequência de passos:

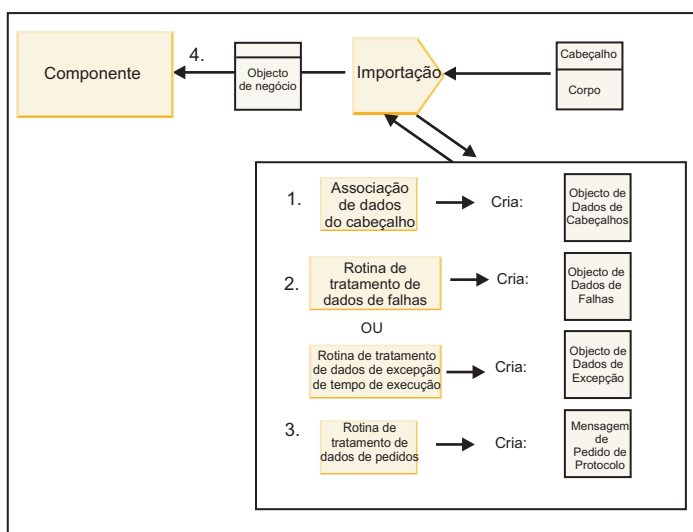


Figura 6. Fluxo de uma resposta de volta pela importação

1. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho transforma o cabeçalho do protocolo num objecto de dados de cabeçalho.
2. É determinado se a resposta é uma falha.
  - Se a resposta for uma falha, o selector de falhas inspeciona a falha para determinar em que falha WSDL é mapeada. A rotina de tratamento de dados de falhas no método transforma, assim, a falha numa resposta de falha.
  - Se a resposta for uma excepção de tempo de execução, a rotina de tratamento de dados da excepção de tempo de execução, se estiver configurada, é chamada.
3. A rotina de tratamento de dados de resposta ou associação no método transforma a resposta num objecto de negócio de resposta.
4. A importação devolve o objecto de negócio de resposta ao componente.

## Exportar e importar configuração de associação

Um dos aspectos fundamentais de exportar e importar associações é a transformação do formato de dados que indica de que forma os dados são mapeados (anular a serialização) de um formato de ligação nativo para um objecto de negócio ou de que forma é mapeado (serializado) de um objecto de negócio para um formato de ligação nativo. Para associações associadas a exportações, também pode especificar um selector de função para indicar qual a operação a executar nos dados. Para associações associadas a exportações e importações, pode indicar de que forma devem ser processadas as falhas que ocorrem durante o processamento.



Além disso, pode especificar informações sobre associações específicas de transporte. Por exemplo, para uma associação HTML específica o URL do ponto final. Para a associação HTTP, as informações específicas de transporte encontram-se descritas nos tópicos “Gerar uma associação de importação HTTP” e “Gerar uma associação de exportação HTTP”. Também é possível encontrar informações sobre outras associações no Information Center.

### **Transformação de formato de dados em importações e exportações:**

Quando uma associação de exportação ou importação é configurada em IBM Integration Designer, uma das propriedades de configuração que especifica é o formato de dados utilizado pela associação.

- Para associações de exportação, nas quais uma aplicação de cliente envia pedidos e recebe respostas de um componente SCA, indique o formato dos dados nativos. Dependendo do formato, o sistema selecciona a rotina de tratamento de dados apropriada para transformar os dados nativos num objecto de negócio (que é utilizado pelo componente SCA) e para transformar inversamente o objecto de negócio em dados nativos (que é a resposta à aplicação de cliente).
- Para associações de importação nas quais um componente SCA envia pedidos e recebe respostas de um serviço fora do módulo, indique o formato de dados dos dados nativos. Dependendo do formato, o sistema selecciona a rotina de tratamento de dados apropriada ou associação de dados para transformar o objecto de negócio em dados nativos ou vice-versa.

IBM Business Process Manager fornece um conjunto de formatos de dados predefinidos e rotinas de tratamento de dados correspondentes ou associações de dados que suportem os formatos. Também pode criar as suas próprias rotinas de tratamento de dados personalizadas e registar o formato de dados para essas rotinas de tratamento de dados. Para mais informações, consulte o tópico “desenvolver rotinas de tratamento de dados” no Information Center IBM Integration Designer.

- *Rotinas de tratamento de dados* são neutros de protocolo e transformam dados de um formato para outro. No IBM Business Process Manager, as rotinas de tratamento de dados transformam, normalmente, os dados nativos (como XML, CSV e COBOL) num objecto de negócio e um objecto de negócio em dados nativos. Pelo facto de serem neutras de protocolo, pode reutilizar a mesma rotina de tratamento de dados com uma variedade de associações de exportação e importação. Por exemplo, pode utilizar a mesma rotina de tratamento de dados XML com uma associação de exportação ou importação HTTP ou com uma associação de exportação ou importação JMS.
- *As associações de dados* também transformam dados nativos num objecto de negócio (e vice-versa), mas são específicas de protocolo. Por exemplo, uma associação de dados HTTP pode ser utilizada com uma associação de exportação ou importação HTTP apenas. Ao contrário das rotinas de tratamento de dados, uma associação de dados HTTP não pode ser reutilizada com uma associação de exportação ou importação MQ.

**Nota:** Três associações de dados HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão obsoletas a partir da IBM Business Process Manager versão 7.0. Utilize rotinas de tratamento de dados sempre que possível.

Como referido anteriormente, é possível criar rotinas de tratamento de dados personalizadas, se necessário. Também pode criar associações de dados personalizadas; no entanto, recomenda-se que crie rotinas de tratamento de dados personalizadas uma vez que podem ser utilizadas em múltiplas associações.

#### *Rotinas de tratamento de dados:*

As rotinas de tratamento de dados são configuradas contra associações de exportação e importação para transformar dados de um formato para outro numa forma neutra a protocolo. Várias rotinas de tratamento de dados são fornecidas como parte do produto, mas também pode criar a sua própria rotina de tratamento de dados, se necessário. Pode associar uma rotina de tratamento de dados a uma

associação de exportação ou importação em um de dois níveis: pode associá-la a todas as operações na interface da exportação ou importação ou pode associá-la a uma operação específica para o pedido ou resposta.

### Rotinas de tratamento de dados predefinidas

Utilize o IBM Integration Designer para especificar a rotina de tratamento de dados que pretende utilizar.

As rotinas de tratamento de dados que estão predefinidas para a sua utilização estão listadas na seguinte tabela que também descreve de que forma cada rotina de tratamento de dados transforma dados internos e externos.

**Nota:** Excepto onde notado, estas rotinas de tratamento de dados podem ser utilizadas com associações JMS, Generic JMS, MQ JMS, WebSphere MQ, e HTTP.

Consulte o tópico “Data handlers (Rotinas de tratamento de dados)” no Information Center do Integration Designer para obter informações mais detalhadas.

Tabela 23. Rotinas de tratamento de dados predefinidas

Rotina de tratamento de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
ATOM	Analisa ATOM feeds num objecto de negócio ATOM feed.	Dispõe em série um objecto de negócio ATOM feed para ATOM feeds.
Delimitado	Analisa dados delimitados num objecto de negócio.	Dispõe em série um objecto de negócio para dados delimitados, incluindo CSV.
Largura fixa	Analisa dados de largura fixa no objecto de negócio.	Dispõe em série um objecto de negócio para dados de largura fixa.
Processado por WTX	Delega a conversão do formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX deriva da rotina de tratamento de dados.	Delega a conversão do formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX deriva da rotina de tratamento de dados.
Processado por WTX Invoker	Delega a transformação de formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX é fornecido pelo utilizador.	Delega a transformação de formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX é fornecido pelo utilizador.
JAXB	Dispõe em série beans de Java para um objecto de negócio utilizando as regras de mapeamento definidas pela Arquitectura Java para especificação de associação XML (JAXB).	Anula a serialização de um objecto de negócio para beans de Java utilizando as regras de mapeamento definidas pela especificação JAXB.
JAXWS <b>Nota:</b> A rotina de tratamento de dados JAXWS pode ser utilizada apenas com a associação EJB.	Utilizada por uma associação EJB para transformar um objecto Java de resposta ou objecto Java de excepção para um objecto de negócio de resposta, utilizando as regras de mapeamento definidas pela especificação Java API para serviços da Web XML (JAX-WS).	Utilizada por uma associação EJB para transformar um objecto de negócio em parâmetros de método Java de saída, utilizando as regras de mapeamento definidas pela especificação JAX-WS.
JSON	Analisa dados JSON num objecto de negócio.	Dispõe em série um objecto de negócio para dados JSON.

Tabela 23. Rotinas de tratamento de dados predefinidas (continuação)

Rotina de tratamento de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Corpo nativo	Analisa os bytes nativos, texto, mapa, sequência ou objecto num dos cinco objectos de negócio (texto, bytes, mapa, sequência ou objecto).	Transforma os cinco objectos de negócio base em byte, texto, mapa, sequência ou objecto.
SOAP	Analisa a mensagem SOAP (e o cabeçalho) num objecto de negócio.	Dispõe em série um objecto de negócio para uma mensagem SOAP.
XML	Analisa dados XML num objecto de negócio.	Dispõe em série um objecto de negócio para dados XML.
UTF8XMLDataHandler	Analisa dados XML codificados em UTF-8 num objecto de negócio.	Dispõe em série um objecto de negócio em dados XML codificados em UTF-8 ao enviar uma mensagem.

### Criar uma rotina de tratamento de dados

Informações detalhadas sobre como criar uma rotina de tratamento de dados podem ser encontradas no tópico “Developing data handlers (Desenvolver rotinas de tratamento de dados)” no Information Center Integration Designer.

#### Associações de dados:

As associações de dados estão configuradas contra associações de exportação e importação para transformar dados de um formato para outro. As associações de dados são específicas para um protocolo. São fornecidas várias associações de dados como parte do produto, mas também pode criar a sua própria associação de dados, se necessário. Pode associar uma associação de dados com uma exportação ou importação a um de dois níveis-pode associá-la a todas as operações na interface da exportação ou importação ou pode associá-la com uma operação específica para o pedido ou resposta.

Utilize IBM Integration Designer para especificar qual a associação de dados que pretende utilizar ou para criar a sua própria associação de dados. Pode encontrar um debate sobre a criação de associações de dados na secção de “descrição geral de associações JMS, MQ JMS e JMS genérico” do Information Center IBM Integration Designer.

### Associações JMS

A seguinte tabela lista as associações de dados que podem ser utilizadas com:

- Associações JMS
- Associações genéricas JMS
- Associações JMS do WebSphere MQ

A tabela também inclui uma descrição das tarefas executadas pelas associações de dados.

Tabela 24. Associações de dados predefinidas para associações JMS

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Objecto Java serializado	Transforma o objecto serializado Java num objecto de negócio (que está correlacionado com o tipo de entrada ou saída no WSDL).	Dispõe em série um objecto de negócio para o objecto disposto em série Java na mensagem de objecto JMS.

Tabela 24. Associações de dados predefinidas para associações JMS (continuação)

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Bytes reiniciados ciclicamente	Extrai os bytes da mensagem de bytes JMS de chegada e reinicia-os ciclicamente no objecto de negócio JMSBytesBody.	Extrai os bytes do objecto de negócio JMSBytesBody e reinicia-os ciclicamente na mensagem de bytes JMS de saída.
Entrada de mapa reiniciada ciclicamente	Extrai o nome, valor e informação de tipo para cada entrada na mensagem de mapa JMS de entrada e cria uma lista de objectos de negócio MapEntry. Em seguida, reinicia ciclicamente a lista no objecto de negócio JMSTextBody.	Extrai o nome, valor e informação de tipo da lista MapEntry no objecto de negócio JMSMapBody e cria as entradas correspondentes na mensagem de correlação JMS de saída.
Objecto reiniciado ciclicamente	Extrai o objecto da mensagem de objecto JMS de entrada e reinicia-o ciclicamente no objecto de negócio JMSObjectBody.	Extrai o objecto do objecto de negócio JMSObjectBody e reinicia-o ciclicamente na mensagem de objecto JMS de saída.
Texto reiniciado ciclicamente	Extrai o texto da mensagem de texto JMS de entrada e reinicia-o ciclicamente no objecto de negócio JMSTextBody.	Extrai o texto do objecto de negócio JMSTextBody e reinicia-o ciclicamente na mensagem de texto JMS de saída.

## Associações do WebSphere MQ

A seguinte tabela lista as associações de dados que podem ser utilizadas com o WebSphere MQ e descreve as tarefas que as associações de dados executam.

Tabela 25. Associações de dados predefinidas para associações WebSphere MQ

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Objecto Java serializado	Transforma o objecto Java serializado da mensagem recebida num objecto de negócio (que está mapeado como o tipo de entrada ou saída no WSDL).	Transforma um objecto de negócio num objecto serializado Java na mensagem de saída.
Bytes reiniciados ciclicamente	Extrai os bytes da mensagem de bytes MQ não estruturada e reinicia-os ciclicamente no objecto de negócio JMSBytesBody.	Extrai os bytes de um objecto de negócio JMSBytesBody e reinicia ciclicamente os bytes na mensagem de bytes MQ não estruturada de saída.
Texto reiniciado ciclicamente	Extrai o texto da mensagem de texto MQ não estruturada e reinicia-o ciclicamente no objecto de negócio JMSTextBody.	Extrai o texto de um objecto de negócio JMSTextBody e reinicia-o ciclicamente numa mensagem de texto MQ não estruturada.
Entrada sequencial de reinício cíclico	Extrai o nome e informação de tipo para cada entrada na mensagem sequencial JMS de entrada e cria uma lista dos objectos de negócio StreamEntry. Em seguida, reinicia ciclicamente a lista no objecto de negócio JMSSStreamBody.	Extrai o nome e informação de tipo da lista StreamEntry no objecto de negócio JMSSStreamBody e cria entradas correspondentes na mensagem de saída JMSSStream.

Para além das associações de dados enunciadas na Tabela 25 na página 60, o WebSphere MQ também utiliza associações de dados de cabeçalho. Para mais informações, consulte o Information Center IBM Integration Designer.

### Associações HTTP

A seguinte tabela lista as associações de dados que podem ser utilizadas com HTTP e descreve as tarefas que as associações de dados podem executar.

Tabela 26. associações de dados predefinidas para associações HTTP

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Bytes reiniciados ciclicamente	Extrai os bytes do corpo da mensagem HTTP de entrada e reinicia-os ciclicamente no objecto de negócio HTTPBytes.	Extrai os bytes do objecto de negócio HTTPBytes e inclui-os no corpo da mensagem HTTP de saída.
Texto reiniciado ciclicamente	Extrai o texto do corpo da mensagem HTTP de entrada e reinicia-o ciclicamente no objecto de negócio HTTPText.	Extrai o texto do objecto de negócio HTTPText e inclui-o na mensagem HTTP de saída.

### Selectores de função em associações de exportação:

Um selector de função é utilizado para indicar qual a operação a executar nos dados para uma mensagem de pedido. Os selectores de função estão configurados como parte de uma associação de exportação.

Considere uma exportação SCA que expõe uma interface. A interface contém duas operações-Criar e Actualizar. A exportação tem uma associação JMS que lê de uma fila.

Quando chega uma mensagem à fila, a exportação é passada aos dados associados, mas que operação da interface da exportação deve ser chamada no componente ligado? A operação é determinada pelo selector de função e a configuração de associação de exportação.

O selector de função devolve o nome de função nativo (o nome de função no sistema cliente que enviou a mensagem). O nome de função nativo é depois mapeado para a operação ou nome de função na interface associada à exportação. Por exemplo, na imagem seguinte, o selector de função devolve o nome de função nativo (CRT) da mensagem recebida, o nome de função nativo é mapeado para a operação Criar e o objecto de negócio é enviado ao componente SCA com a operação Criar.

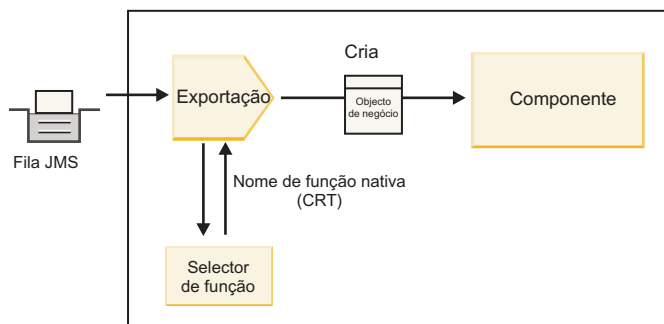


Figura 7. O selector de função

Se a interface tiver apenas uma operação, não há necessidade de especificar um selector de função.

Vários selectores de função pré-embalados encontram-se disponíveis e listados nas seguintes secções.

### Associações JMS

A seguinte tabela lista os selectores de função que podem ser utilizados com:

- Associações JMS
- Associações genéricas JMS
- Associações JMS do WebSphere MQ

*Tabela 27. Selectores de função predefinidos para associações JMS*

Selector de função	Descrição
Selector de função JMS para associações de dados JMS simples	Utiliza a propriedade JMSType da mensagem para seleccionar o nome de operação.
Selector de função de propriedade de cabeçalho JMS	Devolve o valor da propriedade de cadeia JMS TargetFunctionName do cabeçalho.
Selector de função de porta de ligação de serviço JMS	Determina se o pedido é uma operação unidireccional ou bidireccional ao examinar a propriedade JMSReplyTo definida pelo cliente.

### Associações WebSphere MQ

A seguinte tabela lista os selectores de função que podem ser utilizados com associações WebSphere MQ.

*Tabela 28. Selectores de função predefinidos para associações WebSphere MQ*

Selector de função	Descrição
Selector de função MQ handleMessage	Devolve handleMessage como um valor que é mapeado utilizando as associações de método de exportação para o nome de uma operação na interface.
MQ utiliza o selector de função predefinido JMS	Lê a operação nativa a partir da propriedade TargetFunctionName da pasta de um cabeçalho MQRFH2.
MQ utiliza o formato de corpo de mensagem como função nativa	Encontra o campo Formatar do último cabeçalho e devolve esse campo como cadeia.
Selector de função do tipo MQ	Cria um método na associação de exportação ao obter um URL contendo as propriedades Msd, Set, Type (Tipo) e Format (Formatar) no cabeçalho MQRFH2.
Selector de função de porta de ligação de serviço MQ	Utiliza a propriedade MsgType no cabeçalho MQMD para determinar o nome da operação.

### Associações HTTP

A seguinte tabela lista os selectores de função que podem ser utilizados com associações HTTP.

*Tabela 29. Selectores de função predefinidos para associações HTTP*

Selector de função	Descrição
Selector de função HTTP baseado no cabeçalho TargetFunctionName	Utiliza a propriedade de cabeçalho HTTP TargetFunctionName do cliente para determinar qual a operação a chamar no tempo de execução da exportação.
Selector de função HTTP baseado no método URL e HTTP	Utiliza o caminho relativo do URL juntamente com o método HTTP do cliente para determinar a operação nativa definida na exportação.

Tabela 29. Selectores de função predefinidos para associações HTTP (continuação)

Selector de função	Descrição
O selector de função de porta de ligação de serviço HTTP baseado no URL com um nome de operação	Determina o método para chamar com base no URL se "operationMode = oneWay" foi anexado ao pedido URL.

**Nota:** Também pode criar o seu próprio selector de função, utilizando o IBM Integration Designer. São fornecidas informações sobre como criar um selector de funções no Information Center do IBM Integration Designer. Por exemplo, uma descrição da criação de um selector de função para associações WebSphere MQ podem ser encontradas em “Descrição geral dos selectores de função MQ”.

### Processamento de falhas:

Pode configurar as associações de importação e exportação para processar falhas (por exemplo, exceções de negócio) que ocorrem durante o processamento ao especificar rotinas de tratamento de dados de falha. Pode definir uma rotina de tratamento de dados em três níveis-pode associar uma rotina de tratamento de dados de falha com uma falha, com uma operação ou para todas as operações com uma associação.

Uma rotina de tratamento de dados de falha processa dados de falha e transforma-os no formato correcto para serem enviados pela associação de exportação ou importação.

- Para uma associação de exportação, a rotina de tratamento de dados de falha transforma o objecto de negócio enviado pelo componente numa mensagem de resposta que pode ser utilizada pela aplicação do cliente.
- Para uma associação de importação, a rotina de tratamento de dados de falha transforma os dados de falha ou mensagem de resposta enviados por um serviço num objecto de negócio de excepção pode ser utilizada pelo componente SCA.

Para associações de importação, a associação chama o selector de falha, que determina se a mensagem de resposta é uma resposta normal, uma falha de negócio ou uma excepção de tempo de execução.

Pode especificar uma rotina de tratamento de dados de falha para uma determinada falha, para uma operação e para todas as operações com uma associação.

- Se a rotina de tratamento de dados de falha estiver definida em todos os três níveis, a rotina de tratamento de dados associada a uma determinada falha é chamada.
- Se as rotinas de tratamento de dados de falha estiverem definidas nos níveis de operação e associação, a rotina de tratamento de dados associada à operação é chamada.

São utilizados editores no IBM Integration Designer para especificar processamento de falhas. O editor de interface é utilizado para indicar se vai haver uma falha na operação. Depois de ser gerada uma associação com esta interface, o editor na vista de propriedades permite configurar a forma como a falha será processada. Para mais informações, consulte o tópico “Selector de falhas” nas informações do Information Center do IBM Integration Designer.

*De que forma são processadas as falhas nas associações de exportação:*

Quando há ocorrência de uma falha durante o processamento do pedido de uma aplicação de cliente, a associação de exportação pode devolver a informação de falha ao cliente. Pode configurar a associação de exportação para especificar de que forma a falha deve ser processada e devolvida ao cliente.

Configure a associação de exportação utilizando IBM Integration Designer.

Durante o processamento do pedido, o cliente chama uma exportação com um pedido e a exportação chama o componente SCA. Durante o processamento do pedido, o componente SCA pode devolver uma resposta de negócio ou pode lançar uma excepção de negócio de serviço ou uma excepção de tempo de execução de serviço. Quando isto ocorre, a associação de exportação transforma a excepção numa



mensagem de falha e envia-a ao cliente, como apresentado na seguinte imagem e descrito nas secções que se seguem.

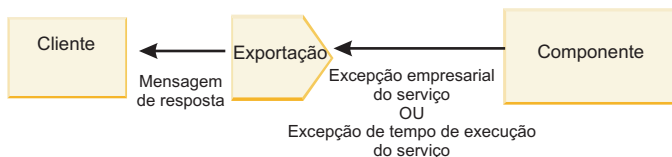


Figura 8. De que forma a informação de falha é enviada do componente através da associação de exportação ao cliente

Pode criar uma rotina de tratamento de dados personalizada ou associação de dados para processar falhas.

### Falhas de negócio

Falhas de negócio são falhas de negócio ou excepções que ocorrem durante o processamento.

Considere a seguinte interface que tem uma operação createCustomer. Esta operação tem duas falhas de negócio definidas: CustomerAlreadyExists e MissingCustomerId.

Operações

Operações e os respectivos parâmetros

	Nome	Tipo
▼ createCustomer		
Entrada(s)	entrada	CustomerInfo
Saída(s)	Saída	CustomerInfo
Falha	Cliente já existe	Customer Already ExistsBO
Falha	MissingCustomerId	MissingCustomerIdBO

Figura 9. Interface com duas falhas

Neste exemplo, se um cliente envia um pedido para criar um cliente (para este componente SCA) e esse cliente já existir, o componente lança uma falha CustomerAlreadyExists para a exportação. A exportação precisa de propagar esta falha de negócio de volta para o cliente de chamada. Para esse efeito, utiliza a rotina de tratamento de dados que está definida na associação de exportação.

Quando uma falha de negócio é recebida pela associação de exportação, ocorre o seguinte processamento:

1. A associação determina qual a rotina de tratamento de dados a chamar para processar a falha. Se a excepção de negócio de serviço contiver o nome da falha, é chamada a rotina de tratamento de dados que está definida na falha. Se a excepção de negócio de serviço não contiver o nome da falha, o nome da falha é derivado ao corresponder os tipos de falha.
2. A associação chama a rotina de tratamento de dados com o objecto de dados da excepção de negócio de serviço.
3. A rotina de tratamento de dados falhada transforma o objecto de dados de falha numa mensagem de resposta e devolve-a à associação de exportação.
4. A exportação devolve a mensagem de resposta ao cliente.

Se a excepção de negócio de serviço contiver o nome da falha, é chamada a rotina de tratamento de dados que está definida na falha. Se a excepção de negócio de serviço não contiver o nome da falha, o nome da falha é derivado ao corresponder os tipos de falha.



## Excepções de tempo de execução

Uma excepção de tempo de execução é uma excepção que ocorre na aplicação SCA durante o processamento de um pedido que não corresponde à falha de negócio. Ao contrário das falhas de negócio, as excepções de tempo de execução não estão definidas na interface.

Em determinados cenários pode querer propagar estas excepções de tempo de execução pela aplicação de cliente de forma que a aplicação do cliente possa tomar a acção adequada.

Por exemplo, se um cliente enviar um pedido (ao componente SCA) para criar um cliente e um erro de autorização ocorrer durante o processamento do pedido, o componente lança uma excepção de tempo de execução. Esta excepção de tempo de execução tem de ser propagada de volta ao cliente de chamada para que possa tomar a acção adequada quanto à autorização. Isto é alcançado através da rotina de tratamento de dados de excepção de tempo de execução configurada na associação de exportação.

**Nota:** Pode configurar uma rotina de tratamento de dados de excepção de tempo de execução apenas em associações HTTP.

O processamento de uma excepção de tempo de execução é semelhante ao processamento de uma falha de negócio. Se tiver sido definida uma rotina de tratamento de dados de excepção de tempo de execução, ocorre o seguinte processamento:

1. A associação de exportação chama a rotina de tratamento de dados adequada com a excepção de tempo de execução de serviço.
2. A rotina de tratamento de dados transforma o objecto de dados de falha numa mensagem de resposta e devolve-a à associação de exportação.
3. A exportação devolve a mensagem de resposta ao cliente.

O processamento de falhas e o processamento de excepções de tempo de execução são opcionais. Se não pretender propagar falhas ou excepções de tempo de execução pelo cliente de chamada, não configure a rotina de tratamento de dados de falha ou rotina de tratamento de dados de excepção de tempo de execução.

*Como são processadas falhas em associações de importação:*

Um componente utiliza uma importação para enviar um pedido a um serviço fora do módulo. Quando ocorre uma falha durante o processamento do pedido, o serviço devolve a falha à associação de importação. Pode configurar a associação de importação para especificar de que forma a falha deve ser processada e devolvida ao componente.

Pode configurar a associação de importação utilizando IBM Integration Designer. É possível especificar uma rotina de tratamento de dados de falha (ou associação de dados) e especifica também um selector de falhas.

## Rotinas de tratamento de dados de falhas

O serviço que processa o pedido envia à associação de importação informações de falha em forma de uma excepção ou mensagem de resposta que contém os dados de falha.

A associação de importação transforma a excepção de serviço ou mensagem de resposta numa excepção de negócio de serviço ou excepção de tempo de execução de serviço, como indicado na imagem seguinte e descrito nas secções seguintes.

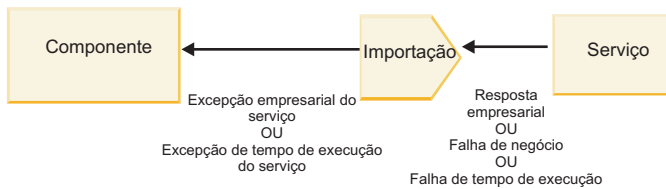


Figura 10. De que forma são enviadas as informações de falha do serviço através da importação para o componente

Pode criar uma rotina de tratamento de dados personalizada ou associação de dados para processar falhas.

### Selectores de falha

Quando configura uma associação de importação, pode especificar um selector de falhas. O selector de falhas determina se a resposta de importação é uma resposta efectiva, uma excepção de negócio ou uma falha de tempo de execução. Determina também, do corpo de resposta ou cabeçalho, o nome de falha nativo que é mapeado pela configuração de associação para o nome de uma falha na interface associada.

Dois tipos de selectores de falhas pré-emballados estão disponíveis para utilização com JMS, MQ JMS, JMS Genérico JMS, WebSphere MQ e importações HTTP:

Tabela 30. Selectores de falhas pré-emballados

Tipo de selector de falhas	Descrição
Com base no cabeçalho	Determina se uma mensagem de resposta é uma falha de negócio, uma excepção de tempo de execução ou uma mensagem normal com base nos cabeçalhos na mensagem de resposta recebida.
SOAP	Determina se a mensagem SOAP de resposta é uma resposta normal, uma falha de negócio ou uma excepção de tempo de execução.

O que se segue mostra exemplos de selectores de falhas com base no cabeçalho e do selector de falhas SOAP.

- Selector de falhas com base no cabeçalho

Se uma aplicação pretender indicar que a mensagem recebida é uma falha de negócio, têm de existir dois cabeçalhos na mensagem recebida para falhas de negócio, que são apresentados da seguinte forma:

Nome do cabeçalho = FaultType, Valor do cabeçalho = Business  
 Nome do cabeçalho = FaultName, Valor do cabeçalho = <nome de falha nativo definido pelo utilizador>

Se uma aplicação pretender indicar que a mensagem de resposta recebida é uma excepção de tempo de execução, tem de existir um cabeçalho na mensagem de entrada, que é mostrada da seguinte forma:

Header name = FaultType, Header value = Runtime

- Selector de falhas SOAP

Uma falha de negócio pode ser enviada como parte da mensagem SOAP com o seguinte cabeçalho SOAP personalizado. "CustomerAlreadyExists" é o nome da falha neste caso.

```
<ibmSoap:BusinessFaultName
xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists
</ibmSoap:BusinessFaultName>
```

O selector de falhas é opcional. Se não especificar um selector de falhas, a associação de importação não pode determinar o tipo de resposta. Assim, a associação processa-o como uma resposta de negócio e chama a rotina de tratamento de dados ou associação de dados.

É possível criar um selector de falhas personalizado. Os passos para criar um selector de falhas personalizado são fornecidos no tópico “Developing a custom fault selector (Desenvolver um selector de falhas personalizado)” do Information Center do IBM Integration Designer.

### Falhas de negócio

Uma falha de negócio pode ocorrer quando existe um erro no processamento de um pedido. Por exemplo, se enviar um pedido para criar um cliente e esse cliente já existe, o serviço envia uma excepção de negócio à associação de importação.

Quando uma excepção de negócio é recebida pela associação, os passos de processamento dependem de um selector de falhas ter sido configurado para a associação.

- Se não tiver sido configurado nenhum selector de falhas, a associação chama a rotina de tratamento de dados de resposta ou associação de dados.
- Se tiver sido configurado um selector de falhas, ocorre o seguinte processamento:
  1. A associação de importação chama o selector de falhas para determinar se a resposta é uma falha de negócio, uma resposta de negócio ou uma falha de tempo de execução.
  2. Se a resposta for uma falha de negócio, a associação de importação chama o selector de falhas para fornecer o nome de falha nativo.
  3. A associação de importação determina a falha WSDL correspondente ao nome de falha nativo devolvido pelo selector de falhas.
  4. A associação de importação determina a rotina de tratamento de dados de falha que está configurada para esta falha WSDL.
  5. A associação de importação chama esta rotina de tratamento de dados de falha com os dados de falha.
  6. A rotina de tratamento de dados de falha transforma os dados de falha num objecto de dados de devolve-o à associação de importação.
  7. A associação de importação constrói um objecto de excepção de negócio de serviço com o objecto de dados e o nome da falha.
  8. A importação devolve a excepção de negócio de serviço ao componente.

### Excepções de tempo de execução

Uma excepção de tempo de execução pode ocorrer quando existe um problema em comunicar com o serviço. O processamento de uma exception de tempo de execução é semelhante ao processamento de uma excepção de negócio. Se tiver sido configurado um selector de falhas, ocorre o seguinte processamento:

1. A associação de importação chama a rotina de tratamento de dados de excepção de tempo de execução apropriada com os dados de excepção.
2. A rotina de tratamento de dados de excepção de tempo de execução transforma os dados de excepção num objecto de excepção de tempo de execução de serviço e devolve-o à associação de importação.
3. A importação devolve o objecto de excepção de tempo de execução ao componente.

### Interoperacionalidade entre módulos SCA e serviços Open SCA

O IBM WebSphere Application Server V7.0 Feature Pack for Service Component Architecture (SCA) fornece um modelo de programação simples mas poderoso para construir aplicações baseadas nas especificações Open SCA. Os módulos SCA de IBM Business Process Manager utilizam associações de importação e exportação para interoperar com serviços Open SCA desenvolvidos num ambiente Rational Application Developer e alojados pelo WebSphere Application Server Feature Pack for Service Component Architecture.

Uma aplicação SCA chama uma aplicação Open SCA através de uma associação de importação. Uma aplicação SCA recebe uma aplicação Open SCA através de uma associação de exportação. Uma lista de associações suportadas é apresentada na “Chamar serviços através de associações interoperáveis” na página 69.

## Chamar Open SCA a partir de módulos SCA

As aplicações SCA desenvolvidas com o IBM Integration Designer podem chamar aplicações Open SCA desenvolvidas num ambiente Rational Application Developer. Esta secção fornece um exemplo de chamada de um serviço Open SCA a partir de um módulo SCA utilizando uma associação de importação SCA.

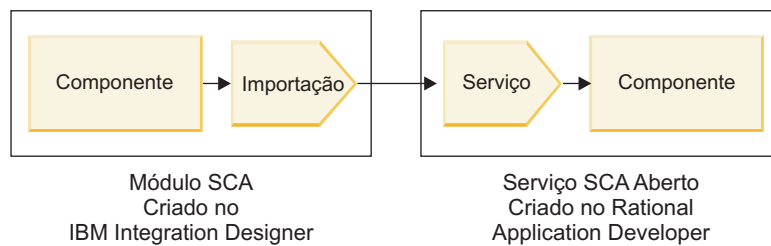


Figura 11. Componente em módulo SCA a chamar serviço Open SCA

Não é necessária qualquer configuração especial para chamar um serviço Open SCA.

Para ligar a um serviço Open SCA através de uma associação de importação SCA, fornece o nome do componente e o nome de serviço do serviço Open SCA na associação de importação.

1. Para obter o nome do componente de destino e serviço do composto Open SCA, execute os seguintes passos:
  - a. Certifique-se que o separador **Properties (Propriedades)** está aberto ao fazer clique em **Window (Janela) > Show View (Mostrar vista) > Properties (Propriedades)**.
  - b. Abra o editor de composto através de duplo clique no diagrama que contém o componente e serviço. Por exemplo, para um componente chamado **customer (cliente)**, o diagrama composto é **customer.composite\_diagram**.
  - c. Faça clique no componente de destino.
  - d. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome do componente de destino.
  - e. Faça clique no ícone de serviço associado ao componente.
  - f. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome do serviço.
2. Para configurar a importação IBM Business Process Manager para ligar ao serviço Open SCA, efectue os seguintes passos:
  - a. No IBM Integration Designer, navegue para o separador **Properties (Propriedades)** da importação SCA que pretende ligar ao serviço Open SCA.
  - b. No campo **Module name (Nome do módulo)**, introduza o nome do componente do passo 1d.
  - c. No campo **Export name (Nome da exportação)**, introduza o nome do serviço do passo 1f.
  - d. Guarde o seu trabalho ao premir Ctrl+S.

## Chamar módulos SCA de serviços Open SCA

Aplicações SCA abertas desenvolvidas num ambiente Rational Application Developer podem chamar aplicações SCA desenvolvidas com o IBM Integration Designer. Esta secção fornece um exemplo de

chamada de um módulo SCA (através de uma associação de exportação SCA) de um serviço Open SCA.

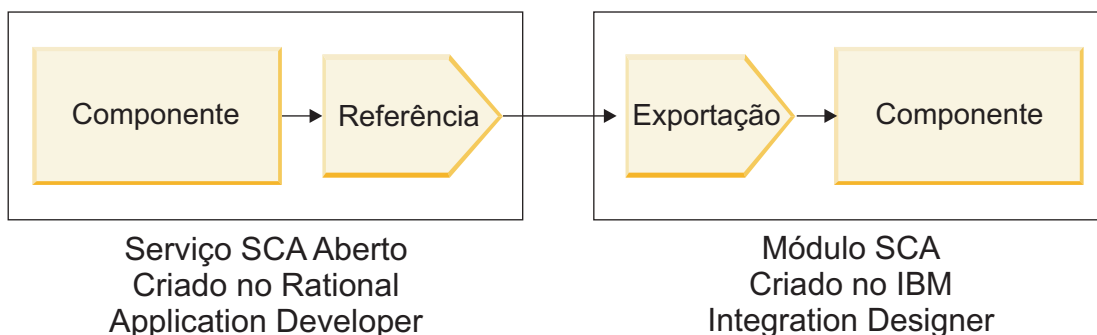


Figura 12. Componente de chamada de serviço Open SCA em módulo SCA

Para ligar a um componente SCA através de uma associação de referência Open SCA, fornece o nome do módulo e nome de exportação.

1. Para obter o nome do módulo e exportação de destino, execute os seguintes passos:
  - a. No IBM Integration Designer, abra o módulo no editor de montagem através de duplo clique no módulo.
  - b. Faça clique na exportação.
  - c. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome da exportação.
2. Configure a referência Open SCA que pretende ligar ao módulo IBM Business Process Manager e exporte:
  - a. No Rational Application Developer, abra o editor de composto através de duplo clique no diagrama de composto que contém o componente e serviço.
  - b. Faça clique no ícone de referência da referência do componente para apresentar as propriedades de referência no separador **Properties (Propriedades)**.
  - c. Faça clique no separador **Binding (Associação)** no lado esquerdo da página.
  - d. Faça clique em **Bindings (Associação)** e, em seguida, faça clique em **Add (Adicionar)**.
  - e. Selecciona a associação **SCA**.
  - f. No campo **Uri**, introduza o nome do módulo IBM Business Process Manager, seguido de uma barra ("/"), seguida do nome de exportação (que determinou no passo 1c).
  - g. Faça clique em **OK**.
  - h. Guarde o seu trabalho ao premir Ctrl+S.

## Chamar serviços através de associações interoperáveis

As seguintes associações são suportadas para interoperacionalidade com um serviço Open SCA.

- Associação SCA

No IBM Business Process Manager, quando um módulo SCA chama um serviço Open SCA, são suportados os seguintes estilos de chamada para execução:

- Assíncrono (unidireccional)
- Síncrono (pedido/resposta)

A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).

Note que a associação SCA suporta propagação de contexto de transacção e segurança.

- Associação de serviço da Web (JAX-WS - Web service) com o protocolo SOAP1.1/HTTP ou o protocolo SOAP1.2/HTTP

A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).

Para além disso, são suportadas as seguintes qualidades de serviço:

- Web Services Atomic Transaction
- Web Services Security
- Associação EJB  
Uma interface Java é utilizada para definir a interacção entre um módulo SCA e um serviço Open SCA quando a associação EJB é utilizada.  
Note que a associação EJB suporta propagação de contexto de transacção e segurança.
- Associações JMS  
A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).  
São suportados os seguintes fornecedores JMS:
  - WebSphere Platform Messaging (Associação JMS)
  - WebSphere MQ (Associação JMS MQ)

**Nota:** Os gráficos comerciais não são interoperáveis em quaisquer associações SCA e, por esse motivo, não são suportados em interfaces utilizadas para interoperar com o WebSphere Application Server Feature Pack for Service Component Architecture.

## Tipos de associação

Utilize *associações* específicas de protocolo com importações e exportações para especificar os meios de transportar dados para dentro ou para fora de um módulo.

### Selecionar associações apropriadas:

Quando está a criar uma aplicação, necessita de saber como seleccionar a associação mais adequada às necessidades da aplicação.

As associações disponíveis em IBM Integration Designer proporcionam uma gama de opções. Nesta lista, pode determinar que tipo de associação poderá ser mais adequada às necessidades da aplicação.

Tenha em consideração uma associação de *Arquitetura do Componente de Serviço (SCA, Service Component Architecture)* quando estes factores forem aplicáveis:

- Todos os serviços estão contidos em módulos, ou seja, não existem serviços externos.
- Pretende separar a função em módulos SCA diferentes que interajam directamente uns com os outros.
- Os módulos estão estreitamente agrupados.

Tenha em consideração uma associação de *Serviço da web* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço externo através da Internet ou de fornecer um serviço através da Internet.
- Os serviços estão vagamente agrupados.
- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.
- O protocolo do serviço externo a que está a aceder ou do serviço que pretende fornecer é o SOAP/HTTP ou SOAP/JMS.

Tenha em consideração uma associação de *HTTP* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço externo através da Internet ou disponibilizar um serviço através da Internet e está a trabalhar com outros serviços da Web, tal como OBTER, COLOCAR, e ELIMINAR.
- Os serviços estão vagamente agrupados.

- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.

Tenha em consideração uma associação de *Enterprise JavaBeans (EJB)* quando estes factores forem aplicáveis:

- A associação destina-se a um serviço importado, que é um EJB ou que necessita de ser acedido por clientes EJB.
- O serviço importado é vagamente agrupado.
- As interações EJB declaráveis não são necessárias.
- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.

Tenha em consideração uma associação de *Enterprise Information Systems (EIS)* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço num sistema EIS utilizando um adaptador de recurso.
- Prefere-se a transmissão de dados síncrona à assíncrona.

Tenha em consideração uma associação de *Java Message Service (JMS)* quando estes factores forem aplicáveis:

**Importante:** Existem vários tipos de associações JMS. Se pretender trocar mensagens SOAP com JMS, tenha em consideração a associação de serviço da web com o protocolo SOAP/JMS. Consulte “Associações do serviço da Web” na página 72.

- Necessita de aceder a um sistema de serviço de mensagens.
- Os serviços estão vagamente agrupados.
- Prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *Serviço de Mensagem Java (JMS) Genérico* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens que de um fornecedor que não pertença à IBM.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *Fila de Mensagens (MQ, Message Queue)* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens do WebSphere MQ e necessita de utilizar as funções nativas de MQ.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *MQ JMS* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens WebSphere MQ mas pode fazê-lo num contexto JMS, ou seja, o subconjunto de funções JMS é suficiente para a sua aplicação.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.



## Associações SCA:

Uma associação da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) permite que um serviço comunique com outros serviços noutros módulos. Uma importação com associação SCA permite-lhe aceder a um serviço noutro módulo SCA. Uma exportação com uma associação SCA permite-lhe oferecer um serviço a outros módulos.

Utilize IBM Integration Designer para criar e configurar associações SACA para importações e exportações nos módulos SCA.

Se os módulos estiverem a ser executados no mesmo servidor ou implementados no mesmo conjunto de unidades, uma associação SCA é a associação mais fácil e rápida a utilizar.

Depois do módulo que contém a associação SCA ser implementado no servidor, pode utilizar a consola administrativa para ver informações sobre a associação ou, no caso de uma associação de importação, alterar as propriedades seleccionadas da associação.

## Associações do serviço da Web:

Uma associação do serviço da Web é o meio de transmitir mensagens desde um componente de Arquitectura do Componente de Serviço (SCA) para um serviço da web (e vice-versa).

Descrição geral de associações de serviço da Web: **Advanced**

Uma importação de serviço da web permite-lhe contactar um serviço da web externo através dos seus componentes de Arquitectura do Componente de Serviço (SCA). Uma associação de exportação de serviço da web permite-lhe expor os seus componentes SCA a clientes como serviços da web.

Com uma associação de serviço da web, acede a serviços externos utilizando mensagens SOAP interoperáveis e qualidades de serviço (QoS).

É possível utilizar o Integration Designer para criar e configurar associações de serviços da Web em importações e exportações em módulos SCA. Estão disponíveis os seguintes tipos de associações de serviços da Web:

- SOAP1.2/HTTP e SOAP1.1/HTTP

Estas associações são baseadas em Java API para Serviços da Web XML (JAX-WS), uma programação Java API para criar serviços da web.

- Utilize SOAP1.2/HTTP se o seu serviço da web for compatível com a especificação SOAP 1.2.
- Utilize SOAP1.1/HTTP se o seu serviço da web for compatível com a especificação SOAP 1.1.

**Importante:** Quando implementa uma aplicação com uma associação de serviço da web (JAX-WS), o servidor de destino não pode ter a opção **Start components as needed (Iniciar componentes consoante for necessário)** seleccionada. Consulte “Verificar a configuração do servidor” na página 81 para obter mais detalhes.

Quando selecciona uma destas associações, é possível enviar anexos com as suas mensagens SOAP.

As associações de serviço da web funcionam com mensagens SOAP padrão. Porém, utilizando uma das associações de serviço da web, é possível personalizar o modo como as mensagens SOAP são analisadas ou escritas. Por exemplo, é possível processar elementos não padronizados em mensagens SOAP ou aplicar processamento adicional à mensagem SOAP. Quando configura a associação, especifica uma rotina de tratamento de dados que executa este processamento na mensagem SOAP.

É possível utilizar conjuntos de políticas com uma associação de serviço da web (JAX-WS). Um conjunto de políticas é um conjunto de tipos de políticas, com cada um a fornecer uma qualidade de serviço (QoS). Por exemplo, o conjunto de políticas WSAddressing fornece um modo neutro de

transporte para tratar, uniformemente, serviços da web e mensagens. Utiliza Integration Designer para seleccionar o conjunto de políticas para a associação.

**Nota:** Se deseja utilizar um conjunto de políticas de Linguagem de Marcação para Asserções de Segurança (SAML), deve executar algumas configurações adicionais, como descrito em “Importar definições de políticas SAML” na página 78.

- SOAP1.1/HTTP

Utilize esta associação se desejar criar serviços da web que utilizem uma mensagem de codificação SOAP baseada em Java API para RPC com base em XML (JAX-RPC).

- SOAP1.1/JMS

utilize esta associação para enviar ou receber mensagens SOAP utilizando um destino de Serviço de Mensagem (JMS) Java.

Independentemente do transporte (HTTP ou JMS) utilizado para transmitir a mensagem SOAP, as associações de serviço da web processam sempre interações de pedido/resposta de modo síncrono. O módulo que estabelece a chamada para execução no fornecedor de serviço é bloqueado até ser recebida uma resposta do fornecedor. Consulte “Chamada para execução síncrona” para obter mais informações sobre este estilo de chamada para execução.

**Importante:** As seguintes combinações de associações de serviço da web não podem ser utilizadas em exportações no mesmo módulo. Se necessitar de expor componentes utilizando mais do que uma destas associações de exportação, necessita de ter cada uma num módulo separado e, de seguida, ligar esses módulos aos seus componentes utilizando a associação SCA:

- SOAP 1.1/JMS e SOAP 1.1/HTTP utilizando JAX-RPC
- SOAP 1.1/HTTP utilizando JAX-RPC e SOAP 1.1/HTTP utilizando JAX-WS
- SOAP 1.1/HTTP utilizando JAX-RPC e SOAP 1.2/HTTP utilizando JAX-WS

Após o módulo SCA que contém a associação de serviço da web ser implementado no servidor, é possível utilizar a consola administrativa para ver informação sobre a associação ou para alterar as propriedades seleccionadas da associação.

**Nota:** Os serviços da Web permitem às aplicações interoperar utilizando descrições padrão de serviços e formatos padrão para as mensagens que trocam. Por exemplo, as associações de serviço da web de importação e exportação podem interoperar com serviços implementados utilizando Web Services Enhancements (WSE) Versão 3.5 e Windows Communication Foundation (WCF) Versão 3.5 para Microsoft .NET. Quando interoperar com tais serviços, deve assegurar que:

- O ficheiro Web Services Description Language (WSDL) utilizado para aceder à exportação de serviço da web inclui um valor de acção SOAP não vazio para cada operação na interface.
- O cliente do serviço da web define o cabeçalho SOAPAction ou o cabeçalho wsa:Action quando envia mensagens para uma exportação de serviço da web.

*Propagação de cabeçalho SOAP:* **Advanced**

Quando processar mensagens SOAP, poderá necessitar de aceder a informações de certos cabeçalhos SOAP nas mensagens que são recebidas, assegurar que as mensagens com cabeçalhos SOAP são enviadas com valores específicos ou permitir aos cabeçalhos SOAP atravessarem um módulo.

Quando configura uma associação de serviço da web no Integration Designer, é possível indicar que deseja transportar cabeçalhos SOAP para serem pagados.

- Quando são recebidos pedidos numa exportação ou quando são recebidas respostas numa importação, a informação de cabeçalho SOAP pode ser acedida, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.

- Quando os pedidos são enviados de uma exportação ou quando as respostas são enviadas de uma importação, os cabeçalhos SOAP podem ser incluídos nessas mensagens.

O formato e presença dos cabeçalhos SOAP propagados podem ser afectados pelos conjuntos de políticas configuradas para a importação ou exportação, como explicado na Tabela 31 na página 75.

Para configurar a propagação de cabeçalhos SOAP para uma importação ou exportação, seleccione (da vista de Propriedades do Integration Designer) o separador **Propagate Protocol Header (Propagar cabeçalho de protocolo)** e seleccione as opções de que necessitar.

### Cabeçalho de endereçamento WS

O cabeçalho de endereçamento WS pode ser propagado pela associação de serviço da web (JAX-WS).

Quando propagar o cabeçalho de endereçamento WS, tenha em atenção a seguinte informação:

- Se activar a propagação para o cabeçalho de endereçamento WS, o cabeçalho será propagado para o módulo nas seguintes circunstâncias:
  - Quando os pedidos são recebidos numa exportação
  - Quando as respostas são recebidas numa importação
- O cabeçalho de endereçamento WS não é propagado para as mensagens de saída de IBM Business Process Manager (isto é, o cabeçalho não é propagado quando os pedidos são enviados de uma importação ou quando as respostas são enviadas da exportação).

### Cabeçalho de segurança WS

O cabeçalho de segurança WS pode ser propagado pela associação de serviço da web (JAX-WS) e pela associação de serviço da web (JAX-RPC).

A especificação da Segurança WS dos serviços da web descreve melhorias ao serviço de mensagens SOAP para proporcionar qualidade de protecção através de integridade de mensagem, confidencialidade de mensagem e autenticação de mensagem única. Estes mecanismos podem ser utilizados para acomodar uma ampla variedade de modelos de segurança e tecnologias de encriptação.

Quando propagar o cabeçalho Segurança WS, tenha em atenção a seguinte informação:

- Se activar a propagação para o cabeçalho de segurança WS, o cabeçalho será propagado através do módulo nas seguintes circunstâncias:
  - Quando os pedidos são recebidos numa exportação
  - Quando os pedidos são enviados de uma importação
  - Quando as respostas são recebidas numa importação
- O cabeçalho *não* será, por predefinição, propagado quando as respostas são enviadas da exportação. Porém, se definir a propriedade JVM **WSSECURITY.ECHO.ENABLED** para **true**, o cabeçalho será propagado quando as respostas forem enviadas da exportação. Neste caso, se o cabeçalho de segurança WS no caminho de pedido não for modificado, os cabeçalhos de segurança WS poderão ser automaticamente passados dos pedidos para as respostas.
- O formato exacto da mensagem SOAP enviada de uma importação para um pedido ou de uma exportação para uma resposta poderá não corresponder exactamente à mensagem SOAP que foi originalmente recebida. Por este motivo, assume-se que qualquer assinatura digital se torna inválida. Se for necessária uma assinatura digital nas mensagens que são enviadas, deve ser estabelecida utilizando o conjunto de políticas de segurança apropriado e os cabeçalhos de segurança WS relacionados com a assinatura digital nas mensagens recebidas devem ser removidos do módulo.

Para propagar o cabeçalho de segurança WS, deve incluir o esquema de segurança WS com o módulo de aplicação. Consulte “Incluindo o esquema de segurança WS num módulo de aplicação” na página 75 para obter o procedimento para incluir o esquema.

## Como os cabeçalhos são propagados

O modo como os cabeçalhos são propagados varia de acordo com a definição da política de segurança sobre associação de importação e exportação, como mostrado na Tabela 31:

Tabela 31. Como são passados os cabeçalhos de segurança

	Associação de exportação sem política de segurança	Associação de exportação com política de segurança
Associação de importação sem política de segurança	<p>Os cabeçalhos de segurança são passados tal como estão pelo módulo. Não são descodificados.</p> <p>Os cabeçalhos são enviados para o exterior no mesmo formato em que são recebidos.</p> <p>A assinatura digital pode tornar-se inválida.</p>	<p>Os cabeçalhos de segurança são descodificados e passados através do módulo com verificação e autenticação de assinatura.</p> <p>Os cabeçalhos descodificados são enviados para o exterior.</p> <p>A assinatura digital pode tornar-se inválida.</p>
Associação de importação com política de segurança	<p>Os cabeçalhos de segurança são passados tal como estão pelo módulo. Não são descodificados.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>	<p>Os cabeçalhos de segurança são descodificados e passados através do módulo com verificação e autenticação de assinatura.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>

Configure os conjuntos de políticas apropriados nas associações de exportação e importação, pois isto isola o autor do pedido de serviço de alterações à configuração ou os requisitos de QoS do fornecedor de serviços. Ter cabeçalhos SOAP padrão visíveis num módulo pode, depois, ser utilizado para influenciar o processamento (por exemplo, registo e rastreio) no módulo. Propagar cabeçalhos SOAP através de um módulo a partir de uma mensagem recebida para uma mensagem enviada significa que as vantagens de isolamento do módulo são reduzidas.

Os cabeçalhos padrão, como cabeçalhos de segurança WS, não devem ser propagados num pedido para uma importação ou num pedido para uma exportação quando a importação ou exportação tiver um conjunto de políticas associado que normalmente resultaria na criação de tais cabeçalhos. Caso contrário, ocorre um erro devido à duplicação dos cabeçalhos. Em alternativa, os cabeçalhos devem ser explicitamente removidos ou a associação da importação ou exportação deve ser configurada para impedir a propagação de cabeçalhos de protocolo.

### Aceder a cabeçalhos SOAP

Quando é recebida uma mensagem, que contém cabeçalhos SOAP, de uma importação ou exportação de serviço da web, os cabeçalhos são colocados na secção de cabeçalhos do objecto de mensagem de serviço (SMO). É possível aceder à informação do cabeçalho, como descrito em "Aceder à informação do cabeçalho SOAP no SMO".

### Incluindo o esquema de segurança WS num módulo de aplicação

O seguinte procedimento destaca os passos para incluir o esquema no módulo de aplicação:

- Se o computador no qual o Integration Designer estiver em execução tiver acesso à Internet, execute os passos seguintes:
  1. Na perspectiva de Business Integration, seleccione **Dependencies (Dependências)** para o seu projecto.

2. Expanda **Predefined Resources (Recursos predefinidos)** e seleccione **WS-Security 1.0 schema files (ficheiros do esquema de segurança WS 1.0)** ou **WS-Security 1.1 schema files (ficheiros do esquema de segurança 1.1)** para importar o esquema para o seu módulo.
  3. Limpe e reconstrua o projecto.
- Se um computador no qual o Integration Designer estiver em execução não tiver acesso à Internet, pode transferir o esquema para um segundo computador que tenha acesso à Internet. Pode, de seguida, copiá-lo para o computador em que Integration Designer está a ser executado.
    1. A partir do computador com acesso à Internet, transfira o esquema remoto:
      - a. Faça clique em **File (Ficheiro) > Import (Importação) > Business Integration > WSDL and XSD (WSDL e XSD)**.
      - b. Seleccione **Remote WSDL (WSDL remota)** ou **XSD File (Ficheiro XSD)**.
      - c. Importe os seguintes esquemas:
        - `http://www.w3.org/2003/05/soap-envelope/`
        - `http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd`
        - `http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd`
    2. Copie os esquemas para o computador sem acesso à Internet.
    3. A partir do computador sem acesso à Internet, importe o esquema:
      - a. Faça clique em **File (Ficheiro) > Import (Importação) > Business Integration > WSDL and XSD (WSDL e XSD)**.
      - b. Seleccione **Local WSDL (WSDL local)** ou **XSD File (Ficheiro XSD)**.
    4. Altere as localizações do esquema para `oasis-wss-wssecurity_secext-1.1.xsd`:
      - a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd`.
      - b. Altere:
 

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='http://www.w3.org/2003/05/soap-envelope'/>
```

 para:
 

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='../w3/_2003/_05/soap_envelope.xsd'/>
```
      - c. Altere:
 

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd'/>
```

 para:
 

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd'/>
```
    5. Altere a localização do esquema para `oasis-200401-wss-wssecurity-secext-1.0.xsd`:
      - a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-200401-wss-wssecurity-secext-1.0.xsd`.
      - b. Altere:
 

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd"/>
```

 para:
 

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="../w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd"/>
```
    6. Limpe e reconstrua o projecto.

Propagação de cabeçalho de transporte: **Advanced**

Quando processar mensagens SOAP, poderá necessitar de aceder a informações de certos cabeçalhos de transporte nas mensagens que são recebidas, assegurar que as mensagens com cabeçalhos de transporte são enviadas com valores específicos ou permitir aos cabeçalhos de transporte atravessarem um módulo.

Quando configura uma associação de serviço da web no Integration Designer, é possível indicar que deseja transportar cabeçalhos para serem propagados.

- Quando são recebidos pedidos numa exportação ou quando são recebidas respostas numa importação, a informação de cabeçalho de transporte pode ser acedida, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que estes cabeçalhos sejam modificados.
- Quando as respostas são enviadas de uma exportação ou quando os pedidos são enviados de uma importação, os cabeçalhos de transporte podem ser incluídos nessas mensagens.

### Especificar a propagação de cabeçalhos

Para configurar a propagação de cabeçalhos de transporte para uma importação ou exportação, execute os seguintes passos:

1. A partir da vista de Propriedades do Integration Designer, seleccione **Binding (Associação) > Propagation (Propagação)**.
2. Defina a opção de propagação de cabeçalho de transporte de que necessita.

**Nota:** A propagação de cabeçalho de transporte está desactivada por predefinição e apenas pode ser implementada num ambiente em tempo de execução de Versão 7.0.0.3 (ou posterior). Tenha também em atenção que, para a Versão 7.0.0.3, a propagação de cabeçalho de transporte está limitada a cabeçalhos de transporte de HTTP apenas.

Se activar a propagação de cabeçalhos de transporte, os cabeçalhos serão propagados através de um módulo de mensagens recebidas e, se não remover, explicitamente, os cabeçalhos, estes serão utilizados em chamadas para execução subsequentes na mesma sessão.

**Nota:** Os cabeçalhos de transporte não podem ser propagados quando estiver a utilizar a associação de serviço da web (JAX-RPC).

### Aceder à informação de cabeçalho

Quando a propagação de cabeçalho de transporte é activada para mensagens recebidas, todos os cabeçalhos de transporte (incluindo cabeçalhos definidos por cliente) são visíveis no objecto de mensagem de serviço (SMO). Pode definir os cabeçalhos para valores diferentes ou criar novos. Repare, porém, que não existe verificação ou validação dos valores que definir e quaisquer cabeçalhos incorrectos poderão causar problemas ao tempo de execução do serviço da web.

Tenha em conta a seguinte informação sobre a definição de cabeçalhos de HTTP:

- Quaisquer alterações aos cabeçalhos que estejam reservados para o motor do serviço da web não serão respeitadas na mensagem de saída. Por exemplo, os cabeçalhos de versão ou método HTTP, tipo de conteúdo, comprimento de conteúdo e SOAPAction estão reservados para o motor de serviço da web.
- Se o valor de cabeçalho for um número, o número (em vez de cadeia) deve ser directamente definido. Por exemplo, utilize **Max-Forwards = 5** (em vez de **Max-Forwards = Max-Forwards: 5**) e **Age = 300** (em vez de **Age = Age: 300**).
- Se a mensagem de pedido for inferior a 32 KB de tamanho, o motor de serviço da web retira o cabeçalho Transferência-Codificação e, em alternativa, define o cabeçalho Conteúdo-Comprimento para o tamanho fixo da mensagem.
- O idioma do conteúdo é reposto por WAS.channel.http no caminho de resposta.



- Uma definição inválida para Actualização resulta num erro 500.
- Os seguintes cabeçalhos acrescentam o valor reservado pelo motor de serviço da web às definições do cliente:
  - Agente do utilizador
  - Controlo de Cache
  - Pragma
  - Aceitar
  - Ligação

É possível aceder à informação do cabeçalho de uma das seguintes maneiras:

- Utilizar uma primitiva de mediação para aceder a estruturas SMO  
Consulte as ligações “Informações Relacionadas” para encontrar informação sobre como utilizar primitivas de mediação.
- Utilizar o SPI de serviço de contexto

O seguinte código exemplo lê os cabeçalhos de transporte de HTTP do serviço de contexto:

```
HeadersType headerType = ContextService.INSTANCE.getHeaders();
HTTPHeaderType httpHeaderType = headerType.getHTTPHeader();
List HTTPHeader httpHeaders = httpHeaderType.getHeader();
if(httpHeaders!=null){
    for(HTTPHeader httpHeader: httpHeaders){
        String httpHeadername = httpHeader.getName();
        String httpHeaderValue = httpHeader.getValue();
    }
}
List PropertyType properties = headerType.getProperties();
if(properties!=null){
    for(PropertyType property: properties){
        String propertyName = property.getName();
        String propertyValue = property.getValue().toString();
    }
}
```

## Resolução de problemas

Se tiver problemas ao enviar os cabeçalhos revistos, pode interceptar a mensagem TCP/IP ao utilizar ferramentas como o Monitor TCP/IP no Integration Designer. Acede-se ao Monitor TCP/IP ao seleccionar **Run/Debug (Executar/Depurar) > TCP/IP Monitor (Monitor TCP/IP)** da página de Preferências.

Também pode ver os valores do cabeçalho usando o rastreio do motor JAX-WS: **org.apache.axis2.\*=all: com.ibm.ws.websvcs.\*=all:**

*Trabalhar com associações do serviço da web (JAX-WS):* **Advanced**

Quando utilizar associações do serviço da web (JAX-WS) com as suas aplicações, pode adicionar uma qualidade de serviço (QoS) de Linguagem de Marcação para Asserções de Segurança (SAML) à associação. Deve, primeiro, utilizar a consola administrativa para importar o conjunto de políticas. Também pode utilizar a consola administrativa para se certificar de que o servidor está devidamente configurado para utilização com a associação do serviço da web (JAX-WS).

*Importar definições de políticas SAML:* **Advanced**

A Security Assertion Markup Language (SAML) é um OASIS padrão, com base XML, para permutar informação relativa a identidade do utilizador e atributos de segurança. Quando configura uma ligação de serviço da Web (JAX-WS) no Integration Designer, pode especificar uma definição de políticas SAML.



Primeiro, utilizar a consola administrativa de IBM Business Process Manager para disponibilizar as definições de política SAML, de modo a que possam ser importadas para Integration Designer.

As definições de política SAML estão, geralmente, localizadas no directório de configuração de perfis:

*profile\_root/config/templates/PolicySets*

Antes de iniciar este procedimento, verificar se os seguintes directórios (que contêm as definições de políticas) estão localizados no directório de configuração de perfis:

- predefinição SAML11 Bearer WSHTTPS
- predefinição SAML20 Bearer WSHTTPS
- predefinição SAML11 Bearer WSSecurity
- predefinição SAML20 Bearer WSSecurity
- predefinição SAML11 HoK Public WSSecurity
- predefinição SAML20 HoK Public WSSecurity
- predefinição SAML11 HoK Symmetric WSSecurity
- predefinição SAML20 HoK Symmetric WSSecurity
- predefinição Username WSHTTPS

Caso os directórios não estejam no directório de configuração de perfis, deverão ser copiados para esse directório, a partir da seguinte localização:

*app\_server\_root/profileTemplates/default/documents/config/templates/PolicySets*

Para importar as definições de política para a consola administrativa, seleccione as que pretende tornar disponíveis no Integration Designer e, em seguida, guarde um ficheiro .zip para cada uma dessas definições de política numa localização que seja acessível pelo Integration Designer.

1. Importar as definições de política, seguindo os passos que se seguem:
  - a. A partir da consola administrativa, fazer clique em **Services (Serviços) > Policy Sets (Definições de Política) > Application policy sets (Definições de política de Aplicação)**.
  - b. Fazer clique em **Import (Importar) > from Default Repository (do Repositório Predefinido)**.
  - c. Seleccionar as definições de política SAML predefinidas, e fazer clique em **OK (OK)**.
2. Exportar as definições de política para que possam ser utilizadas por Integration Designer:
  - a. A partir da página de definições de política de Aplicação, seleccionar a definição de política SAML para exportar, e fazer clique em **Export (Exportar)**.

**Nota:** Caso a página de definições de política de Aplicação não esteja a ser exibida, fazer clique em **Services (Serviços) > Policy Sets (Definições de Política) > Application Policy Sets (Definições de Política de Aplicação)** a partir da consola administrativa.

- b. Na página seguinte, fazer clique na ligação do ficheiro .ZIP para a definição de política.
- c. Na janela File Download (Transferir Ficheiro), fazer clique em **Save (Guardar)**, e indicar a localização que é acessível a partir de Integration Designer.
- d. Fazer clique em **Back (Retroceder)**.
- e. Concluir os passos 2a concluídos 2d para cada definição de política que deseja exportar.

As definições de política SAML são guardadas nos ficheiros .ZIP e estão prontas para serem importadas para Integration Designer.

Importar as definições de política para Integration Designer, conforme descrito no tópico “Definições de política”.

Chamar serviços da Web que exigem autenticação base de HTTP: **Advanced**

A autenticação base de HTTP utiliza um nome de utilizador e uma palavra-passe para autenticar um cliente do serviço para um destino final seguro. Poderá definir uma autenticação base de HTTP, enviando ou recebendo pedidos de serviços da Web.

O utilizador define a autenticação básica de HTTP para receber pedidos de serviço da Web através da configuração de associações de exportação de Serviços da Web Java API para XML (JAX-WS), como descrito em Criar e atribuir funções de segurança a exportações de serviços da Web.

A autenticação base de HTTP pode ser activada para pedidos de serviço da Web, que são enviados por uma associação de importações JWX-WS, num destes dois modos:

- Quando configurar a associação de importações de um módulo SCA, poderá seleccionar o conjunto de política de autenticação HTTP que foi fornecida, chamada BBPMHTTPBasicAuthentication (que é fornecida com a associação de importação do serviço da Web (JAX-WS) ou qualquer outro conjunto de política que inclua a política HTTPTransport.
- Aquando da construção do módulo SCA, poderá utilizar capacidades de mediação de fluxo para criar, de forma dinâmica, um novo cabeçalho de autenticação HTTP, e especificar a informação relativa ao nome de utilizador e à palavra-passe no cabeçalho.

**Nota:** O conjunto de políticas tem preferência sobre o valor especificado no cabeçalho. Caso deseje utilizar o conjunto de valores no cabeçalho de autenticação HTTP no tempo de execução, não deverá anexar um conjunto de políticas que inclua a política HTTPTransport. Não utilizar, especificamente, o conjunto de políticas BPMHTTPBasicAuthentication predefinido e, se definiu um conjunto de políticas, certifique-se de que ele não contém a política HTTPTransport.

Para obter mais informações sobre definições e associações de políticas de serviços da Web e sobre como estas são utilizadas, consulte Definições de políticas de serviços da Web do Information Center do WebSphere Application Server.

- Para utilizar o conjunto de políticas fornecidas, executar os passos que se seguem:
  1. Opcional: Na consola administrativa, crie uma associação de políticas gerais do cliente ou edite uma associação existente que inclui a política HTTPTransport com os valores de ID do utilizador e palavra-passe necessários.
  2. Em IBM Integration Designer, criar uma associação de importações de um serviço da Web (JAX-WS), e anexar o conjunto de políticas BPMHTTPBasicAuthentication.
  3. Executar *um* dos seguintes passos:
    - Em IBM Integration Designer, nas propriedades de associações de importações dos serviços da Web (JAX-WS), especificar o nome de uma associação de políticas de clientes gerais, já existente, que inclua a política HTTPTransport.
    - Após a implementação do módulo SCA, utilize a consola administrativa para seleccionar uma associação de políticas de clientes existente ou para criar uma associação de políticas de clientes nova e, em seguida, associe-a à associação de importação.
  4. Opcional: Na consola administrativa do Process Server, editar a associação de definições das políticas seleccionadas, para especificar o ID e palavra-passe exigidos.
- Para especificar o nome de utilizador e a palavra-passe no cabeçalho da autenticação HTTP, executar um dos seguintes passos:
  - Utilizar a primitiva de mediação HTTP Header Setter IBM Integration Designer para criar o cabeçalho de autenticação HTTP, e especifique o nome de utilizador e a palavra-passe.
  - Caso seja exigida uma lógica adicional, utilizar um código Java numa primitiva de mediação, personalizada (conforme exibido no exemplo seguinte) para:
    1. Criar um cabeçalho de autenticação HTTP.
    2. Especificar a informação relativa ao nome de utilizador e à palavra-passe.

3. Adicionar o novo cabeçalho de autenticação HTTP ao HTTPControl.
4. Retroceder o HTTPControl actualizado para o Context service (Serviço de contexto).

```
//Get the HeaderInfoType from contextService
ContextService contextService = (ContextService) ServiceManager.INSTANCE
.locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
//Get the HTTP header and HTTP Control from HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
//Create new HTTPAuthentication and set the HTTPCredentials
HTTPAuthentication authorization = factory.createHTTPAuthentication();
HTTPCredentials credentials = factory.createHTTPCredentials();
authorization.setAuthenticationType(HTTPAuthenticationType.BASIC_LITERAL);
credentials.setUserId("USERNAME");
credentials.setPassword("PASSWORD");
authorization.setCredentials(credentials);
cp.setAuthentication(authorization);
httpHeaderType.setControl(cp);
// Set header info back to the current execution context.
contextService.setHeaderInfo(headers);
```

Verificar a configuração do servidor: **Advanced**

Quando implementa uma aplicação com uma associação de serviço da Web (JAX-WS), tem de se certificar que o servidor no qual a aplicação está implementada não tem a opção **Start components as needed (Iniciar componentes como necessário)** seleccionada.

Poderá verificar se esta opção está seleccionada, executando os seguintes passos, a partir da consola administrativa:

1. Fazer clique em **Servers (Servidores) > Server types (Tipos de servidor) > WebSphere application servers (Servidores WebSphere Application)**.
2. Fazer clique no nome do servidor.
3. A partir do separador Configuração, determinar se foram seleccionados os **Start components as needed (Componentes de início necessários)**.
4. Executar um dos seguintes passos:
  - Caso estejam seleccionados os **Start components as needed (Componentes de início necessários)**, remover a marca, e fazer clique em **Apply (Aplicar)**.
  - Caso não estejam seleccionados os **Start components as needed (Componentes de início necessários)** fazer clique em **Cancel (Cancelar)**.

Anexos em mensagens SOAP: **Advanced**

É possível enviar e receber mensagens SOAP que incluam dados binários (como ficheiros PDF ou imagens JPEG) como anexos. Os anexos podem ser *referenciados* (ou seja, explicitamente representados como componentes da mensagem na interface de serviço) ou *não referenciados* (situação na qual ou números e tipos arbitrários dos anexos podem ser incluídos).

Um anexo referenciado pode ser representado numa das seguintes maneiras:

- Os anexos MTOM utilizam a codificação especificada do Mecanismo de Optimização de Transmissão de Mensagem SOAP (<http://www.w3.org/TR/soap12-mtom/>). Os anexos MTOM são activados através de uma opção de configuração nas associações de importação e exportação e são o modo recomendado para codificar anexos para novas aplicações.
- Como um elemento wsi:swaRef-typed no esquema de mensagem  
Os anexos definidos utilizando wsi:swaRef type estão em conformidade com a Web Services Interoperability Organization (WS-I) *Perfil de Anexos Versão 1.0* (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como os elementos da mensagem se relacionam com os componentes MIME.
- Como um componente da mensagem de nível superior, utilizando um tipo de esquema binário  
Os anexos representados como componentes de nível superior estão em conformidade com a especificação das *Mensagens SOAP com Anexos* (<http://www.w3.org/TR/SOAP-attachments> ).  
Os anexos representados como componentes da mensagem de nível superior também podem ser configurados para assegurar que as mensagens e documento WSDL produzidos pela associação estão em conformidade com a WS-I *Perfil de Anexos Versão 1.0* e WS-I *Perfil Básico Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Um anexo não referenciado é transportado numa mensagem SOAP sem qualquer representação no esquema da mensagem.

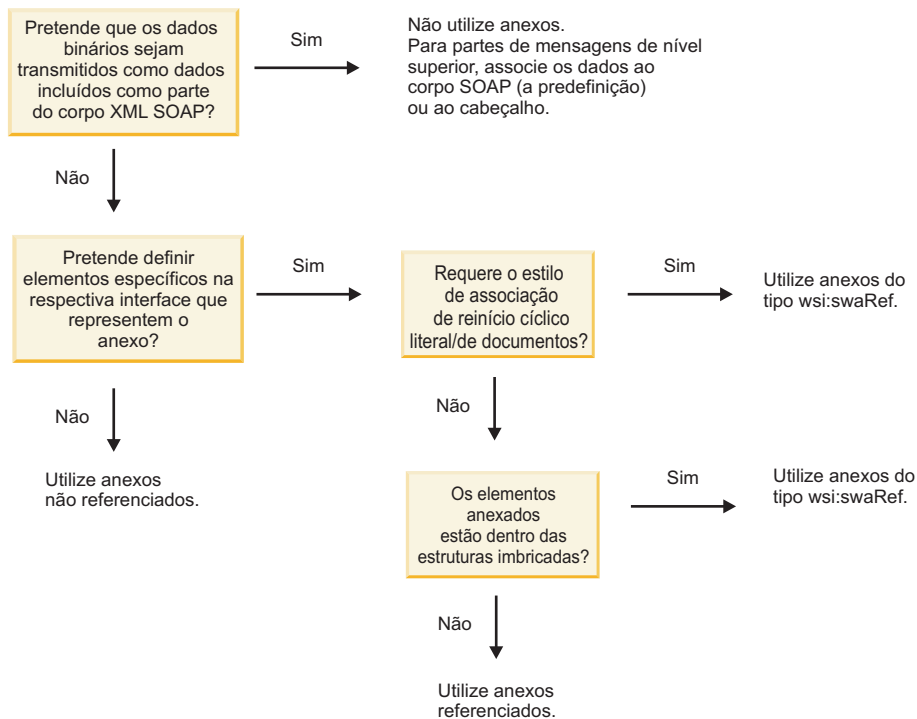
Em todos os casos, excepto anexos MTOM, a associação SOAP WSDL deve incluir uma associação MIME para anexos a serem utilizados, e o tamanho máximo dos anexos não deve ultrapassar 20 MB.

**Nota:** Para enviar ou receber mensagens SOAP com anexos, deve utilizar uma das associações de serviço da web baseadas em Java API para Serviços da Web XML (JAX-WS).

Como seleccionar o estilo de anexo apropriado:  **Advanced**

Quando conceber um serviço que inclua dados binários, tenha em consideração o modo como esses dados binários são transportados nas mensagens SOAP que são enviadas e recebidas pelo serviço.

O Mecanismo de Optimização de Transmissão de Mensagens (MTOM) deve ser utilizado para anexos, caso a aplicação do serviço da Web ligada o suporte. Caso contrário, o seguinte diagrama mostra como são seleccionados outros estilos de anexo. Utilize o seguinte conjunto de questões para determinar o estilo de anexo apropriado:



*Anexos MTOM: partes de mensagens de nível superior:*

É possível enviar e receber mensagens de serviço da Web que incluem anexos Mecanismo de Optimização de Transmissão de Mensagens (MTOM, Message Transmission Optimization Mechanism) SOAP. Numa mensagem SOAP de várias partes MIME, o corpo SOAP é a primeira parte da mensagem, vindo o anexo ou anexos nas partes seguintes.

Ao enviar ou receber um anexo referenciado numa mensagem SOAP, os dados binários que compõem o anexo (que são, muitas vezes, extensos) são retidos separadamente do corpo da mensagem SOAP, para não terem de ser analisados como XML. Isto resulta num processamento mais eficiente do que se os dados binários fossem mantidos num elemento XML.

Segue-se um exemplo de uma mensagem SOAP MTOM:

```

... other transport headers (outros cabeçalhos de transporte)...
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812;
type="application/xop+xml"; start="
<0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>; start-info="text/xml"; charset=UTF-8

--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: application/xop+xml; charset=UTF-8; type="text/xml";
content-transfer-encoding: binary
content-id:
  <0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sendImage xmlns="http://org.apache.axis2/jaxws/sample/mtom">
      <input>
        <imageData><xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
          href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/></imageData>
      </input>
    </sendImage>
  </soapenv:Body>
</soapenv:Envelope>
  
```

```
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: text/plain
content-transfer-encoding: binary
content-id:
    <1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org>

... binary data goes here ...
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--
```

Repare que no exemplo MTOM, o tipo conteúdo para o envelope SOAP é **application/xop+xml** e que os dados binários são substituídos por um elemento **xop:Include** como se demonstra abaixo:

```
<xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/>
```

### Processamento interno de anexos referenciados

Quando um cliente passa uma mensagem SOAP com um anexo ao componente Service Component Architecture (SCA), a associação da exportação do serviço da web (JAX-WS) remove o anexo em primeiro lugar. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por fim, a associação define o binário do anexo no objecto empresarial.

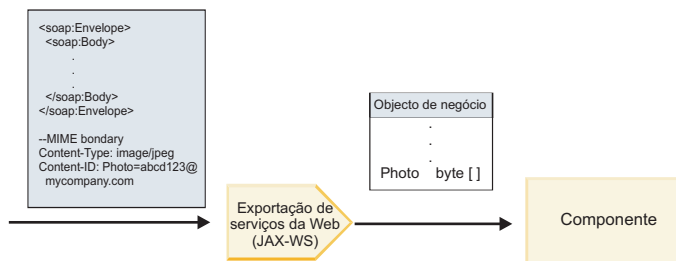


Figura 13. Forma como a associação da exportação do serviço da web (JAX-WS) processa uma mensagem SOAP com um anexo referenciado

### Atributos do anexo MTOM

- O MTOM suporta elementos de anexo dentro de estruturas imbricadas.
- O MTOM só está disponível para o tipo base64Binary.
- O MTOM suporta elementos de anexo dentro de estruturas imbricadas, o que significa que os **bodyPath** para anexos MTOM são as localizações **xpath** do elemento onde o anexo MTOM é retido. A lógica computacional do **bodyPath** segue estritamente o esquema para gerar a localização **xpath**, tal como demonstrado nos exemplos abaixo:
  - Para um tipo não matricial (**maxOccurs** é 1): /sendImage/input/imageData
  - Para um tipo matricial (**maxOccurs** > 1): /sendImage/input/imageData[1]
- Os tipos de anexo misturados não são suportados, o que significa que, se o MTOM for activado na associação de importação, será gerado o anexo MTOM. Se o MTOM for desactivado ou se o valor de configuração MTOM for mantido no valor predefinido na associação de exportação, a mensagem MTOM de entrada não será suportada.

Anexos referenciados: elementos do tipo swaRef: **Advanced**

É possível enviar e receber mensagens SOAP que incluam anexos representados na interface de serviço como elementos do tipo swaRef.

É definido um elemento do tipo swaRef no *Perfil de Anexos Versão 1.0* de Web Services Interoperability Organization (WS-I) (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como os elementos da mensagem estão relacionados com componentes MIME.

Na mensagem SOAP, o corpo SOAP contém um elemento do tipo swaRef que identifica o identificador de conteúdo do anexo.

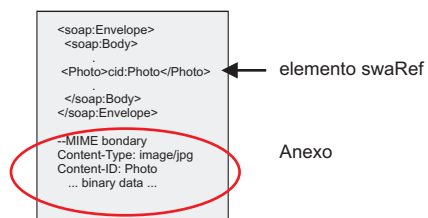


Figura 14. Uma mensagem SOAP com um elemento swaRef

A WSDL para esta mensagem SOAP contém um elemento do tipo swaRef, dentro de um componente da mensagem, que identifica o anexo.

```
<element name="sendPhoto">
  <complexType>
    <sequence>
      <element name="Photo" type="wsi:swaRef"/>
    </sequence>
  </complexType>
</element>
```

A WSDL também deve conter uma associação MIME que indica quais as mensagens multi-parte MIME que devem ser utilizadas.

**Nota:** A WSDL *não* inclui uma associação MIME para o elemento da mensagem do tipo swaRef específico, pois as associações MIME apenas se aplicam a componentes da mensagem de nível superior.

Os anexos representados como elementos do tipo swaRef apenas podem ser propagados através de componentes de fluxo de mediação. Caso um anexo deva ser acedido por ou propagado para outro tipo de componente, utilize um componente de fluxo de mediação para mover o anexo para uma localização que seja acessível por esse componente.

### Processamento interno de anexos

Utiliza Integration Designer para configurar uma associação de exportação para receber o anexo. Cria um módulo e a respectiva interface associada e operações, incluindo um elemento do tipo swaRef. Pode, em seguida, criar uma associação de serviço da web (JAX-WS).

**Nota:** Consulte o tópico “Working with attachments (Trabalhar com anexos)” no Information Center do Integration Designer para obter informações mais detalhadas.

Quando um cliente passa uma mensagem SOAP com um anexo SwaRef para um componente de Arquitetura do Componente de Serviço (SCA), a associação de exportação do serviço da web (JAX-WS) remove, primeiro, o anexo. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por último, a associação coloca o identificador de conteúdo no anexo no objecto de negócio.



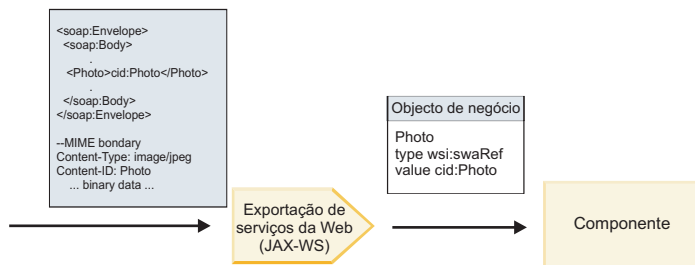


Figura 15. Como a associação de exportação do serviço da web (JAX-WS) processa uma mensagem SOAP com um anexo swaRef

### Aceder a metadados do anexo num componente de fluxo de mediação

Como apresentado na Figura 16, quando os anexos swaRef são acedidos por componentes, o identificador do conteúdo do anexo aparece como um elemento do tipo swaRef.

Cada anexo de uma mensagem SOAP também tem um elemento de **anexos** correspondente no SMO. Quando utilizar o tipo swaRef de WS-I, o elemento dos **anexos** inclui o tipo de conteúdo do anexo e identificador do conteúdo e também os dados binários concretos do anexo.

Para obter o valor de um anexo swaRef, é, assim, necessário obter o valor do elemento do tipo swaRef e, de seguida, localizar o elemento dos **anexos** com o valor correspondente do **identificador de conteúdo**. Tenha em consideração que o valor do **identificador de conteúdo** tem, geralmente, o prefixo **cid:** removido do valor de swaRef.

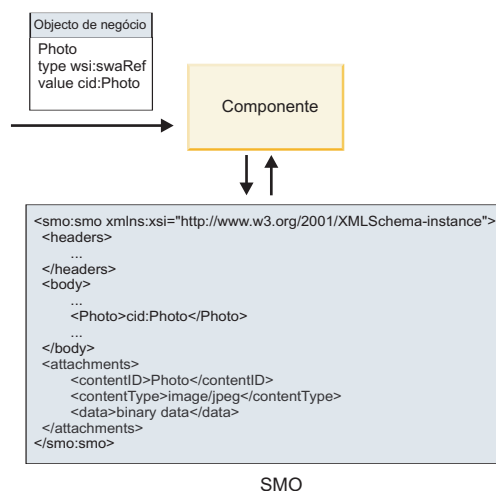


Figura 16. Como os anexos swaRef aparecem no SMO

### Processamento de saída

Utiliza Integration Designer para configurar uma associação de importação de serviço da web (JAX-WS) para chamar um serviço da web externo. A associação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define o anexo que será passado para o serviço da web.

Quando uma mensagem SCA é recebida por uma associação de importação de serviço da web (JAX-WS), os elementos de tipo swaRef são enviados como anexos caso a importação esteja ligada a um componente de fluxo de mediação e o elemento de tipo swaRef tenha um elemento do **anexo** correspondente.

Para processamento de saída, os elementos de tipo swaRef são sempre enviados com os respectivos valores do identificador de conteúdo; porém, o módulo de mediação deve assegurar que existe um elemento dos **anexos** correspondente em concordância com o valor do **identificador de conteúdo**.

**Nota:** Para ser compatível com o Perfil de Anexos de WS-I, o valor do **identificador de conteúdo** deve acompanhar a "codificação do componente do identificador de conteúdo", como descrito na secção 3.8 do *Perfil de Anexos 1.0 de WS-I*.

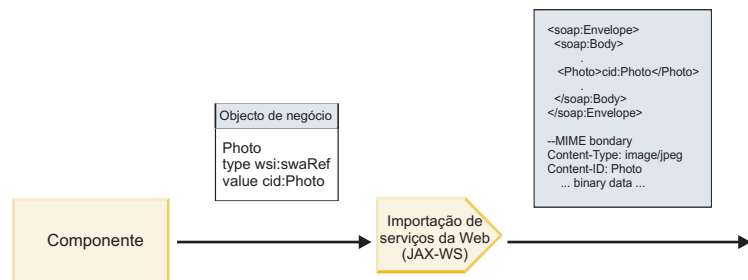


Figura 17. Como a associação de importação do serviço da web (JAX-WS) cria uma mensagem SOAP com um anexo swaRef

### Definir metadados de anexo num componente de fluxo de mediação

Se, no SMO, existir um valor de elemento do tipo swaRef e um elemento dos **anexos**, a associação prepara a mensagem SOAP (com o anexo) e envia-a para um destinatário.

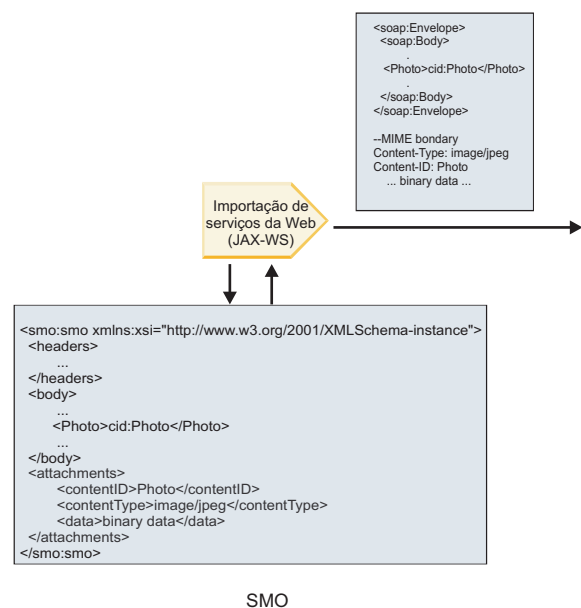


Figura 18. Como um anexo swaRef no SMO é acedido para criar a mensagem SOAP

O elemento dos **anexos** está presente no SMO apenas se um componente de fluxo de mediação estiver directamente ligado à importação ou exportação; não é passado por outros tipos de componente. Se os valores forem necessários num módulo que contenha outros tipos de componente, deve ser utilizado um componente de fluxo de mediação para copiar os valores para uma localização onde possam ser acedidos no módulo e utilizado um componente de fluxo de mediação para definir os valores correctos antes de uma chamada para execução através de uma importação de serviço da web.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.
- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

Anexos referenciados: componentes de mensagem de nível superior: 

É possível enviar e receber mensagens SOAP que incluam anexos binários que sejam declarados como componentes na sua interface de serviço.

Numa mensagem SOAP de multi-parte MIME, o corpo SOAP é o primeiro componente da mensagem e o anexo ou anexos estão em componentes subsequentes.

Qual a vantagem de enviar ou receber um anexo referenciado numa mensagem SOAP? Os dados binários que constituem o anexo (que é, frequentemente, bastante grande) são mantidos em separado do corpo da mensagem SOAP para que não tenham de ser analisados como XML. Isto resulta num processamento mais eficiente do que se os dados binários fossem mantidos num elemento XML.

### Tipos de mensagens SOAP com anexos referenciados

A partir da Versão 7.0.0.3 de IBM Business Process Manager, pode seleccionar o modo como a mensagem SOAP é criada:

- **Mensagens compatíveis com WS-I**

O tempo de execução pode criar mensagens SOAP compatíveis com *Perfil de Anexos Versão 1.0* de WS-I e *Perfil Básico Versão 1.1* de WS-I. Numa mensagem SOAP compatível com estes perfis, apenas um componente de mensagem está associado ao corpo SOAP; para os associados como anexos, a codificação do componente do identificador de conteúdo (como descrito no *Perfil de Anexos Versão 1.0* de WS-I) é utilizada para relacionar o anexo com o componente da mensagem.

- **Mensagens não compatíveis com WS-I**

O tempo de execução pode criar mensagens SOAP não compatíveis com os perfis de WS-I mas que sejam compatíveis com as mensagens criadas na Versão 7.0 ou 7.0.0.2 de IBM Business Process Manager. As mensagens SOAP utilizam elementos de nível superior com o nome do componente da mensagem com um atributo **href** que retém o **content-id** do anexo, mas a codificação do elemento do identificador de conteúdo (como descrito no *Perfil de Anexos Versão 1.0* de WS-I) não é utilizada.

### Seleccionar compatibilidade de WS-I para exportações de serviço da web

Utiliza Integration Designer para configurar uma associação de exportação. Cria um módulo e respectiva interface e operações associadas. Pode, em seguida, criar uma associação de serviço da web (JAX-WS). A página Anexos referenciados apresenta todos os elementos binários da operação criada, seleccionando-se quais os componentes que serão anexos. Especifica, de seguida, na página Especificar a compatibilidade de WS-I AP 1.0 de Integration Designer, uma das seguintes escolhas:

- **Use WS-I AP 1.0 compliant SOAP message (Utilizar mensagem SOAP compatível com WS-I AP 1.0)**

Se seleccionar esta opção, também pode especificar qual o componente da mensagem que deve ser associado ao corpo SOAP.

**Nota:** Esta opção apenas pode ser utilizada quando o ficheiro WSDL correspondente também é compatível com WS-I.

Um ficheiro WSDL que seja criado por Integration Designer de Versão 7.0.0.3 é compatível com WS-I. Porém, se importar um ficheiro WSDL que não seja compatível com WS-I, não é possível seleccionar esta opção.

- **Use non WS-I AP 1.0 compliant SOAP message (Utilize mensagem SOAP não compatível com WS-I AP 1.0)**

Se seleccionar esta opção, a qual é a predefinição, o primeiro componente da mensagem é associado ao corpo SOAP.

**Nota:** Apenas os componentes de mensagem de nível superior (ou seja, elementos definidos no tipo de porta WSDL como componentes na mensagem de entrada ou de saída) que têm um tipo binário (base64Binary ou hexBinary) podem ser enviados ou recebidos como anexos referenciados. Consulte o tópico “Working with attachments (Trabalhar com anexos)” no Information Center do Integration Designer para obter informações mais detalhadas.

Para mensagens compatíveis com WS-I, o identificador de conteúdo que é criado na mensagem SOAP é uma concatenação dos seguintes elementos:

- O valor do atributo de **nome** do elemento **wSDL:part** referenciado por **mime:content**
- O carácter =
- Um valor globalmente único, como um UUID
- O carácter @
- Um nome de domínio válido

### Processamento interno de anexos referenciados

Quando um cliente passa uma mensagem SOAP com um anexo ao componente Service Component Architecture (SCA), a associação da exportação do serviço da web (JAX-WS) remove o anexo em primeiro lugar. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por fim, a associação define o binário do anexo no objecto empresarial.

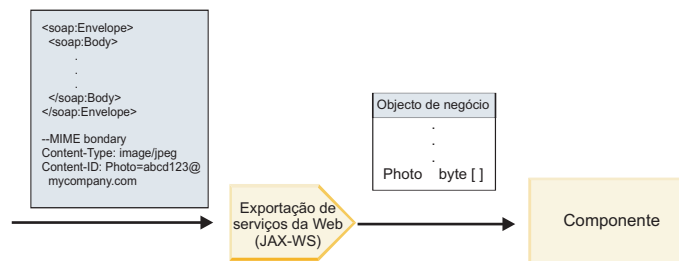


Figura 19. Como a associação de exportação do serviço da web (JAX-WS) processa uma mensagem SOAP compatível com WS-I com um anexo referenciado

### Aceder a metadados do anexo num componente de fluxo de mediação

Como mostrado na Figura 19, quando os anexos referenciados são acedidos por componentes, os dados do anexo aparecem como uma matriz de bytes.

Cada anexo referenciado de uma mensagem SOAP também tem um elemento de **anexos** correspondente no SMO. O elemento dos **anexos** inclui o tipo de conteúdo do anexo e o caminho para o elemento do corpo da mensagem, onde se encontra o anexo.

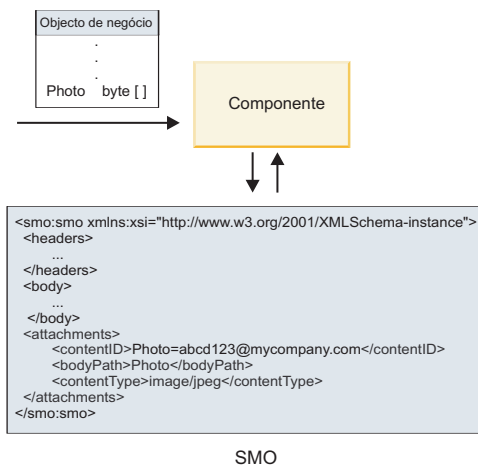


Figura 20. Como os anexos referenciados aparecem no SMO

**Importante:** O caminho para o elemento do corpo da mensagem não é automaticamente actualizado caso a mensagem seja transformada e o anexo deslocado. É possível utilizar um fluxo de mediação para actualizar o elemento dos **anexos** com um novo caminho (por exemplo, como parte da transformação ou utilizando um definidor de elemento da mensagem separado).

### Como são construídas as mensagens SOAP

Utiliza Integration Designer para configurar uma associação de importação de serviço da web (JAX-WS) para chamar um serviço da web externo. A associação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define quais os componentes da mensagem que devem ser passados como anexos. É possível indicar, na página de Especificar a compatibilidade de WS-I AP 1.0 do Integration Designer, uma das seguintes opções:

- **Use WS-I AP 1.0 compliant SOAP message (Utilizar mensagem SOAP compatível com WS-I AP 1.0)**  
Se seleccionar esta opção, também pode especificar qual o componente da mensagem que deve ser associado ao corpo SOAP; todos os outros são associados a anexos ou a cabeçalhos. As mensagens enviadas pela associação não incluem elementos no corpo SOAP que se refiram aos anexos; a relação é expressa através do identificador de conteúdo do anexo, incluindo o nome do componente da mensagem.
- **Use non WS-I AP 1.0 compliant SOAP message (Utilize mensagem SOAP não compatível com WS-I AP 1.0)**  
Se seleccionar esta opção, a qual é a predefinição, o primeiro componente da mensagem é associado ao corpo SOAP; todos os outros são associados a anexos ou cabeçalhos. As mensagens enviadas pela associação incluem um ou mais elementos no corpo SOAP que se referem aos anexos através de um atributo **href**.

**Nota:** O componente que representa um anexo, como definido na WSDL, deve ser de tipo simples (base64Binary ou hexBinary). Se um componente for definido por um complexType, esse componente não pode ser associado como anexo.

### Processamento externo de anexos referenciados

A associação de importação utiliza informação no SMO para determinar como os componentes de mensagem binária de nível superior são enviados como anexos.

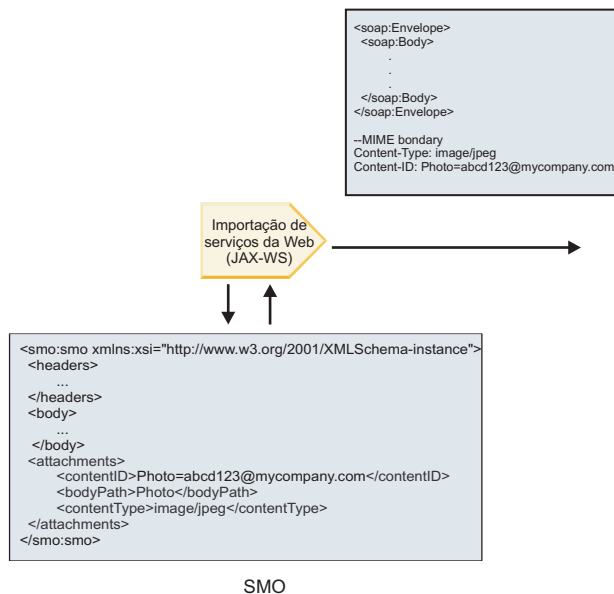


Figura 21. Como o anexo referenciado no SMO é acessado para criar a mensagem SOAP

O elemento dos **anexos** está presente no SMO apenas se um componente de fluxo de mediação estiver directamente ligado à importação ou exportação; não é passado por outros tipos de componente. Se os valores forem necessários num módulo que contenha outros tipos de componente, deve ser utilizado um componente de fluxo de mediação para copiar os valores para uma localização onde possam ser acessados no módulo e utilizado um componente de fluxo de mediação para definir os valores correctos antes de uma chamada para execução através de uma importação de serviço da web.

A associação utiliza uma combinação das seguintes condições para determinar como (ou se) a mensagem é enviada:

- Se existe uma associação de WSDL MIME para o componente binário da mensagem de nível superior e, caso exista, como é definido o tipo de conteúdo
- Se existe um elemento dos **anexos** no SMO cujo valor de **bodyPath** faça referência a um componente binário de nível superior

### Como os anexos são criados quando um elemento do anexo existe no SMO

A tabela seguinte mostra como um anexo é criado e enviado caso o SMO contenha um elemento do **anexo** com um **bodyPath** que corresponda a um componente do nome da mensagem:

Tabela 32. Como é criado o anexo

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> <li>• Nenhum tipo de conteúdo definido para o componente da mensagem</li> <li>• Tipos múltiplos de conteúdo definidos</li> <li>• Tipo de conteúdo de carácter global definido</li> </ul>	Componente da mensagem enviado como um anexo.  O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.  O <b>Content-Type</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para <b>application/octet-stream</b> .

Tabela 32. Como é criado o anexo (continuação)

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com conteúdo individual, sem carácter global, para o componente da mensagem	<p>Componente da mensagem enviado como um anexo.</p> <p>O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.</p> <p>O <b>tipo de conteúdo</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para o tipo definido no elemento do conteúdo de WSDL MIME.</p>
Não presente	<p>Componente da mensagem enviado como um anexo.</p> <p>O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.</p> <p>O <b>Content-Type</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para <b>application/octet-stream</b>.</p> <p><b>Nota:</b> Enviar componentes da mensagem como anexos quando não assim definido na WSDL pode interromper a compatibilidade com o Perfil de Anexos de WS-I 1.0 e isto deve ser evitado, se possível.</p>

### Como os anexos são criados quando não existe qualquer elemento do anexo no SMO

A tabela seguinte mostra como um anexo é criado e enviado caso o SMO não contenha um elemento do **anexo** com um **bodyPath** que corresponda a um componente do nome da mensagem:

Tabela 33. Como é criado o anexo

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> <li>• Nenhum tipo de conteúdo definido para o componente da mensagem</li> <li>• Tipos múltiplos de conteúdo definidos</li> <li>• Tipo de conteúdo de carácter global definido</li> </ul>	<p>Componente da mensagem enviado como um anexo.</p> <p>É criada o <b>identificador de conteúdo</b>.</p> <p>O <b>tipo de conteúdo</b> é definido para <b>application/octet-stream</b>.</p>
Presente com conteúdo individual, sem carácter global, para o componente da mensagem	<p>Componente da mensagem enviado como um anexo.</p> <p>É criada o <b>identificador de conteúdo</b>.</p> <p>O <b>tipo de conteúdo</b> é definido para o tipo definido no elemento do conteúdo de WSDL MIME.</p>
Não presente	Componente da mensagem não enviado como anexo.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.



- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

Anexos não referenciados: **Advanced**

Pode enviar e receber anexos *não referenciados* que não estejam declarados na interface de serviço.

Numa mensagem SOAP de multi-parte MIME, o corpo SOAP é o primeiro componente da mensagem e os anexos estão nos componentes subsequentes. Não é incluída qualquer referência ao anexo no corpo SOAP.

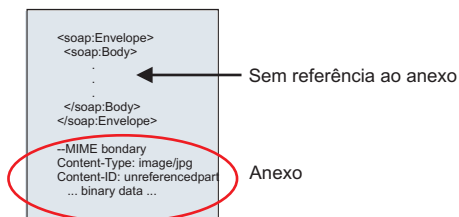


Figura 22. Uma mensagem SOAP com um anexo não referenciado

Pode enviar uma mensagem SOAP com um anexo não referenciado através de uma exportação de serviço da web para uma importação de serviço da web. A mensagem de saída, que é enviada para o serviço da web de destino, contém o anexo.

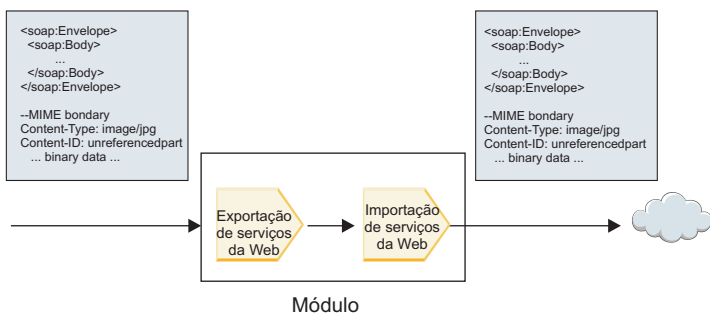


Figura 23. Um anexo a passar através de um módulo SCA

Na Figura 23, a mensagem SOAP, com o anexo, passa sem modificação.

Também pode modificar a mensagem SOAP utilizando um componente de fluxo de mediação. Por exemplo, pode utilizar o componente de fluxo de mediação para extrair dados da mensagem SOAP (dados binários no corpo da mensagem, neste caso) e criar uma mensagem com anexos SOAP. Os dados são processados como componente do elemento dos anexos de um objecto de mensagem de serviço (SMO).

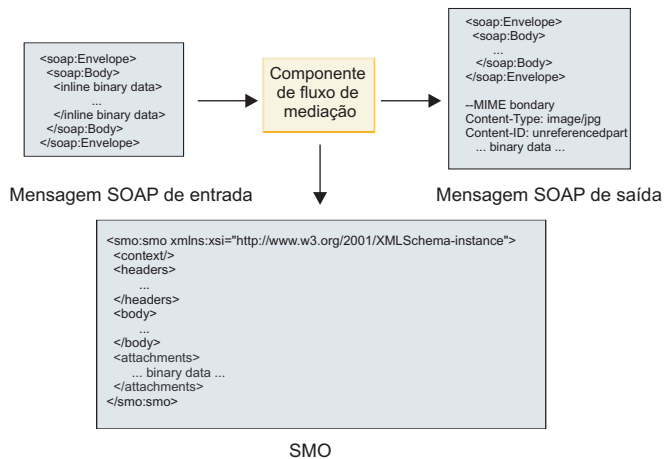


Figura 24. Uma mensagem processada por um componente de fluxo de mediação

Da mesma forma, o componente de fluxo de mediação pode transformar mensagens recebidas ao extrair e ao codificar o anexo e, de seguida, ao transmitir a mensagem sem anexos.

Em vez de extrair dados de uma mensagem SOAP recebida para formar uma mensagem com anexos SOAP, é possível obter os dados do anexo através de uma fonte externa, tal como uma base de dados.

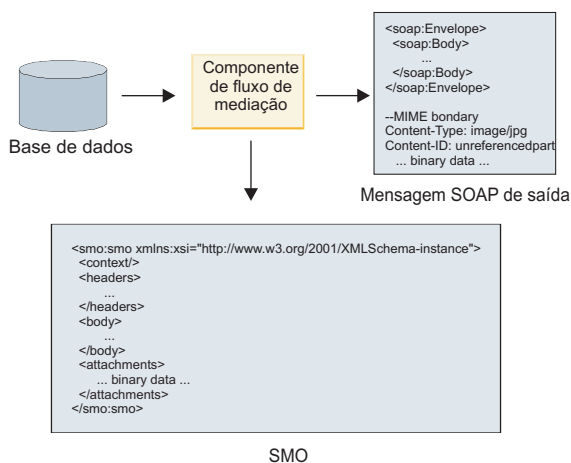


Figura 25. Um anexo obtido de uma base de dados e adicionado à mensagem SOAP

Da mesma forma, o componente de fluxo de mediação pode extrair o anexo de uma mensagem SOAP recebida e processar a mensagem (por exemplo, armazenar o anexo numa base de dados).

Os anexos não referenciados apenas podem ser propagados por componentes de fluxo de mediação. Caso um anexo deva ser acedido por ou propagado para outro tipo de componente, utilize um componente de fluxo de mediação para mover o anexo para uma localização que seja acessível por esse componente.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.
- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

Utilização de associação de estilo de documento WSDL com mensagens multi-parte: **Advanced**

A organização de Web Services Interoperability Organization (WS-I) definiu um conjunto de regras relativamente a como os serviços da web devem ser descritos através de uma WSDL e a como as mensagens SOAP correspondentes devem ser formadas, de modo a assegurar a interoperabilidade.

Estas regras estão especificadas no *Perfil Básico Versão 1.1* de WS-I (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). Em específico, o Perfil Básico de WS-I 1.1 R2712 declara: "Uma associação documento-literal TEM DE ser convertida num ENVELOPE com um soap:Corpo cujo elemento descendente é uma ocorrência da declaração do elemento global referenciado pelo componente wsdl:componente da mensagem."

Isto significa que, quando utilizar uma associação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) definida com componentes múltiplos, apenas um desses componentes deve ser associado ao corpo SOAP de modo a ser compatível com o Perfil Básico 1.1 de WS-I.

Em acréscimo, o Perfil de Anexos 1.0 R2941 de WS-I declara: "Uma wsdl:binding numa DESCRIÇÃO DEVE associar todos os wsdl:part de uma wsdl:message no wsdl:portType, a que se refere, a um de soapbind:body, soapbind:header, soapbind:fault soapbind:headerfault ou mime:content."

Isto significa que, quando utilizar uma associação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) definida com componentes múltiplos, todos os componentes que não aquele seleccionado para ser associado ao corpo SOAP devem ser associados como anexos ou cabeçalhos.

A seguinte abordagem é utilizada quando as descrições de WSDL são criadas para exportações com associações de serviço da web (JAX-WS e JAX-RPC) neste caso:

- É possível seleccionar qual o componente da mensagem que é associado ao corpo SOAP caso exista mais do que um elemento de tipo não binário. Se apenas existir um elemento de tipo não binário, esse elemento é automaticamente associado ao corpo SOAP.
- Para a associação JAX-WS, todos os outros componentes da mensagem de tipo "hexBinary" ou "base64Binary" são associados como anexos referenciados. Consulte “Anexos referenciados: componentes de mensagem de nível superior” na página 88.
- Todos os outros componentes são associados como cabeçalhos SOAP.

As associações de importação JAX-RPC e JAX-WS respeitam a associação SOAP num documento WSDL existente com mensagens de estilo de documento multi-parte, mesmo que associem os componentes múltiplos ao corpo SOAP; porém, não é possível criar clientes de serviço da web para tais documentos WSDL no Rational Application Developer.

**Nota:** A associação JAX-RPC não suporta anexos.

O padrão recomendado quando utilizar mensagens multi-parte com uma operação com associação SOAP de estilo de documento é, assim:

1. Utilizar o estilo moldado de documento/literal. Neste caso, as mensagens têm sempre um componente individual; porém, os anexos não podem ser referenciados (como descrito em “Anexos não referenciados” na página 93) ou do tipo swaRef (como descrito em “Anexos referenciados: elementos do tipo swaRef” na página 84) neste caso.

2. Utilizar o estilo RPC/literal. Neste caso, não existem restrições na associação WSDL em termos de número de componentes associados ao corpo SOAP; a mensagem SOAP resultante tem sempre um descendente único que representa a operação a ser chamada, com os componentes da mensagem a serem descendentes desse elemento.
3. Para a associação JAX-WS, tenha, no máximo, um componente da mensagem que não seja do tipo "hexBinary" ou "base64Binary", excepto se for aceitável associar os outros componentes não binários aos cabeçalhos SOAP.
4. Quaisquer outros casos estão sujeitos ao comportamento descrito.

**Nota:** Existem restrições adicionais quando utiliza mensagens SOAP que não são compatíveis com o *Perfil Básico Versão 1.1* de WS-I.

- O primeiro componente da mensagem deve ser não binário.
- Quando recebe mensagens SOAP de estilo de documento multi-parte com anexos referenciados, a associação JAX-WS espera que cada anexo referenciado seja representado por um elemento descendente do corpo SOAP com um valor de atributo href que identifica o anexo pelo respectivo identificador de conteúdo. A associação JAX-WS envia anexos referenciados para tais mensagens do mesmo modo. Este comportamento não é compatível com o Perfil Básico de WS-I.

Para assegurar que as suas mensagens são compatíveis com o Perfil Básico, siga a abordagem 1 na página 95 ou 2 na lista anterior ou evite a utilização de anexos referenciados para tais mensagens e utilize anexos não referenciados ou do tipo swaRef.

#### **Associações HTTP:**

A associação HTTP está concebida para fornecer a conectividade da área de comunicações partilhada (SCA) ao HTTP. Como consequência, aplicações HTTP existentes ou desenvolvidas recentemente podem participar em ambientes de Arquitectura orientada para o serviço (SOA, Service Oriented Architecture).

HTTP é um protocolo amplamente utilizado para transferir informações na web. Quando está a trabalhar com uma aplicação externa que utiliza o protocolo HTTP, é necessária uma associação HTTP. A associação HTTP transforma os dados que são transferidos para uma mensagem num formato nativo para um objecto de negócio numa aplicação SCA. A associação HTTP também pode transformar dados que são transferidos como um objecto de negócio para um formato nativo esperado pela aplicação externa.

**Nota:** Se pretender interagir com clientes e serviços que utilizam o protocolo SOAP/HTTP dos serviços da Web, considere utilizar uma das associações de serviço da web que oferecem funcionalidade adicional referente ao processamento de qualidades de serviço padrão de serviços da web.

Alguns cenários comuns para utilizar a associação HTTP são descritos na lista seguinte:

- Serviços alojados em SCA podem chamar aplicações HTTP utilizando uma importação HTTP.
- Serviços alojados em SCA podem expor-se como aplicações activadas por HTTP, para que possam ser utilizadas por clientes HTTP, utilizando uma exportação HTTP.
- IBM Business Process Manager e Process Server podem comunicar entre eles em toda a infra-estrutura HTTP, consequentemente, os utilizadores conseguem gerir as suas comunicações de acordo com padrões empresariais.
- IBM Business Process Manager e Process Server podem agir como mediadores de comunicações HTTP, transformando e encaminhando mensagens, o que melhora a integração de aplicações utilizando uma rede HTTP.
- IBM Business Process Manager e Process Server podem ser utilizados para criar uma ponte entre HTTP e outros protocolos, tais como serviços da Web SOAP/HTTP, adaptadores de recurso baseados em Java Connector Architecture (JCA), JMS, etc.

Pode encontrar informações detalhadas sobre como criar associações de importação e exportação HTTP no Information Center do Integration Designer. Consulte os tópicos **Developing integration applications (Desenvolver aplicações de integração) > Accessing external services with HTTP (Aceder a serviços externos com HTTP)**.

*Descrição geral das associações HTTP:*

A associação HTTP fornece conectividade às aplicações alojadas em HTTP. Ela media a comunicação entre aplicações HTTP e permite aplicações existentes baseadas em HTTP a serem chamadas de um módulo.

### **Associações de importação HTTP**

A associação de importação HTTP fornece conectividade externa de aplicações da área de comunicações partilhada (SCA, Service Component Architecture) para um servidor HTTP ou aplicações.

A importação chama um URL de ponto de terminação HTTP. O URL pode ser especificado de uma das três formas:

- O URL pode ser definido dinamicamente nos cabeçalhos HTTP através do URL de substituição dinâmico.
- O URL pode ser definido dinamicamente no elemento de endereço destino SMO.
- O URL pode ser especificado como uma propriedade de configuração na importação.

Esta chamada é sempre síncrona por natureza.

Apesar das chamadas HTTP serem sempre pedido-resposta, a importação HTTP suporta tanto operações unidireccionais como bidireccionais e ignora a resposta no caso de uma operação unidireccional.

### **Associações de exportação HTTP**

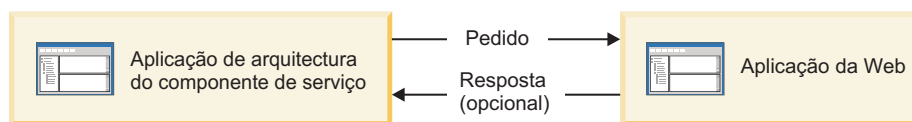
A associação de exportação HTTP fornece conectividade interna de aplicações HTTP para uma aplicação SCA.

Um URL é definido na exportação HTTP. As aplicações HTTP que pretendem enviar mensagens de pedido à exportação utilizam este URL para chamar a exportação.

A exportação HTTP também suporta (comando) ping.

### **Associações HTTP em tempo de execução**

Uma importação com uma associação HTTP em tempo de execução envia um pedido com ou sem dados no corpo da mensagem da aplicação SCA para o serviço externo da web. O pedido é feito pela aplicação SCA para o serviço externo da web, como apresentado na Figura 26.



*Figura 26. Fluxo de um pedido da aplicação SCA para a aplicação web*

Opcionalmente, a importação com a associação HTTP pode receber dados de volta da aplicação da web numa resposta ao pedido.

Com uma exportação, o pedido é feito por uma aplicação de cliente para um serviço da web, como apresentado na Figura 27 na página 98.

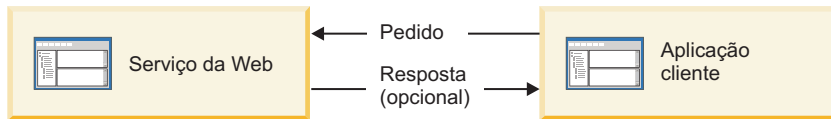


Figura 27. Fluxo de um pedido a partir da aplicação cliente para o serviço da Web.

O serviço da web é uma aplicação da web em execução no servidor. A exportação é implementada nessa aplicação da web como um servlet, para que o cliente envie o seu pedido para um endereço URL. O servlet passa o pedido à aplicação SCA em tempo de execução.

Opcionalmente, a exportação pode enviar dados à aplicação de cliente em resposta ao pedido.

#### Cabeçalhos HTTP:

As associações de importação e exportação HTTP permitem a configuração de cabeçalhos HTTP e que os seus valores sejam utilizados para mensagens de saída. A importação HTTP utiliza estes cabeçalhos para pedidos e a exportação HTTP utiliza-os para respostas.

Cabeçalhos configurados estaticamente e informações de controlo têm precedência sobre valores definidos dinamicamente no tempo de execução. No entanto, os valores de controlo de URL, versão e método de substituição dinâmica substituem os valores estáticos, que, de outra forma, são considerados predefinições.

A associação suporta a natureza dinâmica do URL de importação HTTP ao determinar o valor do URL, versão e método destino de HTTP em tempo de execução. Estes valores são determinados ao extrair o valor da Referência do ponto final, URL de substituição dinâmica, versão e método.

- Para a Referência de ponto final utilize `com.ibm.websphere.sca.addressing.EndpointReference` APIs ou defina o campo `/headers/SMOHeader/Target/address` no cabeçalho SMO.
- Para URL de substituição dinâmica, versão e método utilize a secção de parâmetros de controlo HTTP da área de comunicações partilhada (SCA). Tenha em atenção que o URL de substituição dinâmica tem precedência sobre a Referência do ponto final destino; no entanto, a referência do ponto final aplica-se em associações, sendo a abordagem preferida e devendo ser utilizada, se possível.

As informações sobre controlo e cabeçalho sobre mensagens de saída nas associações de exportação e importação HTTP são processadas na seguinte ordem:

1. As informações de cabeçalho e controlo, excluindo URL de substituição dinâmica HTTP, versão e método da mensagem SCA (prioridade mais baixa)
2. Alterações da consola administrativa ao nível de exportação/importação
3. Alterações da consola administrativa ao nível de método da exportação ou importação
4. Endereço destino especificado através da Referência do ponto final ou do cabeçalho SMO
5. Substituição dinâmica URL, versão e método da mensagem SCA
6. Cabeçalhos e informação de controlo da rotina de tratamento de dados ou associação de dados (prioridade alta)

A exportação e importação HTTP só preenche os cabeçalhos de direcção de entrada e os parâmetros de controlo com dados da mensagem recebida (`HTTPExportRequest` e `HTTPImportResponse`) se a propagação de cabeçalhos de protocolo estiver definida para **True**. De forma inversa, a exportação e importação HTTP só lê e processa cabeçalhos externos e parâmetros de controlo (`HTTPExportResponse` e `HTTPImportRequest`) se a propagação de cabeçalhos de protocolo estiver definida para **True**.

**Nota:** Alterações na rotina de tratamento de dados ou associação de dados para cabeçalhos ou parâmetros de controlo na resposta de importação ou no pedido de exportação não alteram as instruções

de processamento da mensagem dentro da associação de importação ou exportação e só devem ser utilizadas para propagar valores modificados para componentes SCA na direção do fluxo.

O serviço de contexto é responsável pela propagação do contexto (incluindo os cabeçalhos de protocolo, como o cabeçalho HTTP e o contexto de utilizado, como o ID de conta) ao longo de um caminho de chamada SCA. Durante o desenvolvimento no IBM Integration Designer, pode controlar a propagação do contexto através das propriedades de importação e exportação. Para obter mais detalhes, consulte as informações sobre associações de importação e exportação no Information Center do IBM Integration Designer.

### Estruturas de cabeçalho HTTP fornecidas e suporte

A Tabela 34 lista os parâmetros de pedido/resposta para pedidos e respostas de importação HTTP e exportação HTTP.

*Tabela 34. Informações de cabeçalho HTTP fornecidas*

Nome de controle	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
URL	Ignorado	Não definido	Ler a partir da mensagem de pedido. <b>Nota:</b> A cadeia de consulta também faz parte do parâmetro de controle URL.	Ignorado
Versão (valores possíveis: 1.0, 1.1; predefinição é 1.1)	Ignorado	Não definido	Ler a partir da mensagem de pedido	Ignorado
Método	Ignorado	Não definido	Ler a partir da mensagem de pedido	Ignorado
URL de substituição dinâmica	Se definido na rotina de tratamento de dados ou associação de dados, substitui o URL de importação HTTP. Escrito para a mensagem na linha de pedido. <b>Nota:</b> A cadeia de consulta também faz parte do parâmetro de controle URL.	Não definido	Não definido	Ignorado
Versão de substituição dinâmica	Se definido, substitui a versão de importação HTTP. Escrito para a mensagem na linha de pedido.	Não definido	Não definido	Ignorado
Método de substituição dinâmica	Se definido, substitui o Método de importação HTTP. Escrito para a mensagem na linha de pedido.	Não definido	Não definido	Ignorado



Tabela 34. Informações de cabeçalho HTTP fornecidas (continuação)

Nome de controlo	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
Tipo de meio (este parâmetro de controlo traz parte do valor do cabeçalho HTTP do tipo de conteúdo.)	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela rotina de tratamento de dados ou associação de dados.	Ler a partir da mensagem de pedido, cabeçalho do tipo de conteúdo	Ler a partir da mensagem de resposta, cabeçalho do tipo de conteúdo	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela rotina de tratamento de dados ou associação de dados.
Conjunto de caracteres (predefinição: UTF-8)	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela associação de dados.	Ler a partir da mensagem de pedido, cabeçalho do tipo de conteúdo	Ler a partir da mensagem de resposta, cabeçalho do tipo de conteúdo	Suportado, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela associação de dados.
Codificação de transferência (valores possíveis: segmentado, identidade; predefinição é identidade)	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a transformação de mensagem.	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a transformação de mensagem.
Codificação de conteúdo (valores possíveis: gzip, x-gzip, deflacionar, identidade; predefinição é identidade)	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a carga útil.	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a carga útil.
Comprimento do conteúdo	Ignorado	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Ignorado
StatusCode (predefinição: 200)	Não suportado	Ler a partir da mensagem de resposta	Não suportado	Se presente, escrito para a mensagem na linha de resposta.
ReasonPhrase (predefinição: OK)	Não suportado	Ler a partir da mensagem de resposta	Não suportado	Valor de controlo ignorado. O valor da linha de resposta da mensagem é gerado a partir do StatusCode.

Tabela 34. Informações de cabeçalho HTTP fornecidas (continuação)

Nome de controle	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
Autenticação (contém várias propriedades)	Se presente, utilizado para construir o cabeçalho de Autenticação básica. <b>Nota:</b> O valor para este cabeçalho só é codificado no protocolo HTTP. No SCA, é decodificado e passado como texto limpo.	Não aplicável	Ler a partir do cabeçalho de Autenticação básica da mensagem de pedido. A presença deste cabeçalho não indica que o utilizador tenha sido autenticado. A autenticação deve ser controlada na configuração servlet. <b>Nota:</b> O valor para este cabeçalho só é codificado no protocolo HTTP. No SCA, é decodificado e passado como texto limpo.	Não aplicável
Proxy (contém várias propriedades: sistema central, porta, autenticação)	Se presente, utilizado para estabelecer ligação através do proxy.	Não aplicável	Não aplicável	Não aplicável
SSL (contém várias propriedades: arquivo de chaves, palavra-passe do arquivo de chaves, Trustore, palavra-passe do Trustore, ClientAuth)	Se estiver preenchido e o URL de destino for HTTPS, é utilizado para estabelecer uma ligação através do SSL.	Não aplicável	Não aplicável	Não aplicável

#### Associações de dados HTTP:

Para cada mapeamento diferente de dados entre uma mensagem de Arquitetura de Componentes de Serviço (SCA, Service Component Architecture) e uma mensagem de protocolo HTTP, tem de ser configurada uma rotina de tratamento de dados ou uma associação de dados HTTP. As rotinas de tratamento de dados fornecem uma interface neutra de associação que permite a reutilização em todas as associações de transporte e representa a abordagem recomendada; as associações de dados são específicas para uma determinada associação de transporte. As classes de associação de dados específicas de HTTP são fornecidas; também pode gravar rotinas de tratamento de dados personalizadas ou associações de dados.

**Nota:** As três classes de associação de dados HTTP descritas neste tópico (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão obsoletas desde o IBM Business Process Manager Versão 7.0. Em vez de utilizar as associações de dados descritas neste tópico, considere as seguintes rotinas de tratamento de dados:

- Utilize SOAPDataHandler em vez de HTTPStreamDataBindingSOAP.
- Utilize UTF8XMLDataHandler em vez de HTTPStreamDataBindingXML
- Utilize GatewayTextDataHandler em vez de HTTPServiceGatewayDataBinding

As associações de dados são fornecidas para a utilização com importações HTTP e exportações HTTP: associação de dados binários, associação de dados XML e associação de dados SOAP. Não há necessidade de uma associação de dados de resposta para operações unidireccionais. Uma associação de dados é representada pelo nome de uma classe Java cujas instâncias podem ser convertidas de HTTP para ServiceDataObject e vice-versa. Um selector de funções tem de ser utilizado numa exportação que, juntamente com associações de métodos, pode determinar qual é a associação de dados utilizada e qual a operação chamada. As associações de dados fornecidas são:

- Associações de dados binários, que tratam o corpo como dados binários não estruturados. A implementação do esquema XSD de associação de dados binários é como segue:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- Associações de dados XML que suportam o corpo como dados XML. A implementação da associação de dados XML é semelhante à associação de dados JMS XML e não tem restrições no esquema de interface.
- Associações de dados SOAP que suportam o corpo como dados SOAP. A implementação da associação de dados SOAP não tem restrições no esquema de interface.

### Implementar associações de dados HTTP personalizadas

Esta secção descreve como implementar uma associação de dados HTTP personalizada.

**Nota:** A abordagem recomendada é implementar uma rotina de tratamento de dados personalizada porque pode ser reutilizada em todas as associações de transporte.

HTTPStreamDataBinding é a interface principal para processar mensagens HTTP personalizadas. A interface está concebida para permitir o processamento de grandes cargas úteis. No entanto, para tal implementação funcionar, esta associação de dados tem de devolver as informações de controlo e cabeçalhos antes de gravar a mensagem na sequência.

Os métodos e a sua ordem de execução, listados abaixo, têm de ser implementados pela associação de dados personalizados.

Para personalizar uma associação de dados, grave uma classe que implemente o HTTPStreamDataBinding. A associação de dados deve ter quatro propriedades privadas:

- DataObject pDataObject privado
- HTTPControl pCtrl privado
- HTTPHeaders pHeaders privado
- yourNativeDataType nativeData privado

A associação HTTP chama a associação de dados personalizada na seguinte ordem:

- Processamento externo (DataObject para formato nativo):

1. setDataObject(...)
  2. setHeaders(...)
  3. setControlParameters(...)
  4. setBusinessException(...)
  5. convertToNativeData()
  6. getControlParameters()
  7. getHeaders()
  8. write(...)
- Processamento interno (formato nativo para DataObject):
    1. setControlParameters(...)
    2. setHeaders(...)
    3. convertFromNativeData(...)
    4. isBusinessException()
    5. getDataObject()
    6. getControlParameters()
    7. getHeaders()

Necessita de chamar setDataObject(...) em convertFromNativeData(...) para definir o valor de dataObject, que é convertido de dados nativos para a propriedade privada "pDataObject".

```
public void setDataObject(DataObject dataObject)
    throws DataBindingException {
    pDataObject = dataObject;
}

public void setControlParameters(HTTPControl arg0) {
    this.pCtrl = arg0;
}

public void setHeaders(HTTPHeaders arg0) {
    this.pHeaders = arg0;
}
/*
 * Adicione cabeçalho http "IsBusinessException" em pHeaders.
 * Dois passos:
 * 1.Remova primeiro todo o cabeçalho com o nome IsBusinessException
 * (não sensível a maiúsculas e minúsculas)
 * Isto serve para garantir que só está presente um cabeçalho.
 * 2.Adicione o novo cabeçalho "IsBusinessException"
 */
public void setBusinessException(boolean isBusinessException) {
    //remova primeiro todos os cabeçalhos com o nome IsBusinessException
    (não sensível a maiúsculas e minúsculas).
    //Isto serve para garantir que só está presente um cabeçalho.
    //Adicione o novo cabeçalho "IsBusinessException", exemplo de código:
    HTTPHeader header=HeadersFactory.eINSTANCE.createHTTPHeader();
    header.setName("IsBusinessException");
    header.setValue(Boolean.toString(isBusinessException));
    this.pHeaders.getHeader().add(header);
}

public HTTPControl getControlParameters() {
    return pCtrl;
}

public HTTPHeaders getHeaders() {
    return pHeaders;
}

public DataObject getDataObject() throws DataBindingException {
    return pDataObject;
}
}
```

```

/*
 * Obter o cabeçalho "IsBusinessException" de pHeaders, devolve o respectivo valor booleano
 */
public boolean isBusinessException() {
    String headerValue = getHeaderValue(pHeaders,"IsBusinessException");
    boolean result=Boolean.parseBoolean(headerValue);
    return result;
}
public void convertToNativeData() throws DataBindingException {
    DataObject dataObject = getDataObject();
    this.nativeData=realConvertWorkFromSD0ToNativeData(dataObject);
}
public void convertFromNativeData(HTTPInputStream arg0){
    //Customer-developed method to
    //Read data from HTTPInputStream
    //Convert it to DataObject
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}
public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}

```

### Associações EJB:

As associações de importação Enterprise JavaBeans (EJB) permitem que componentes da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) chamem serviços fornecidos pela lógica empresarial Java EE a correr num servidor Java EE. As associações de exportação EJB permitem que componentes SCA sejam expostos como Enterprise JavaBeans, de forma que a lógica empresarial Java EE possa chamar componentes SCA de outra forma não disponíveis para.

#### *Associações de importação EJB:*

As associações de importação EJB permitem que um módulo SCA invoque implementações EJB através da especificação da forma como o módulo de consumo está ligado ao EJB externo. Importar serviços de uma implementação EJB externa permite aos utilizadores ligarem as suas lógicas empresariais ao ambiente IBM Business Process Manager e participarem num processo empresarial.

Utiliza o Integration Designer para criar associações de importação EJB. Pode utilizar qualquer um dos seguintes procedimentos para gerar associações:

- Criar importação EJB utilizando o assistente do serviço externo  
Pode utilizar o assistente do serviço externo no Integration Designer para criar uma importação EJB baseada numa implementação existente. O assistente do serviço externo cria serviços com base em critérios fornecidos pelo utilizador. Em seguida, cria objectos de negócio, interfaces e ficheiros de importação baseados nos serviços descobertos.
- Criar importação EJB utilizando o editor de montagem  
Pode criar uma importação EJB dentro de um diagrama de montagem utilizando o editor de montagem do Integration Designer. A partir da paleta, pode utilizar uma Importação ou utilizar uma classe de Java para criar a associação EJB.

A importação gerada tem associações de dados que criam a ligação Java-WSDL em vez de necessitarem de um componente de ponte Java. Pode ligar directamente um componente com uma referência Web Services Description Language (WSDL) à importação EJB que comunica com um serviço baseado em EJB utilizando uma interface Java.

A importação EJB pode interagir com uma lógica empresarial Java EE utilizando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada para execução para a lógica empresarial Java EE pode ser local (apenas para EJB 3.0) ou remota.

- A chamada para execução local é utilizada quando pretende invocar a lógica empresarial Java EE que reside no mesmo servidor que a importação.

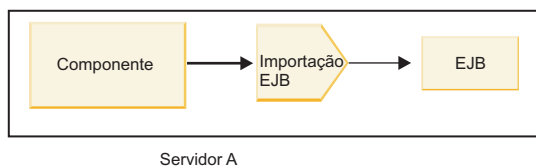


Figura 28. A chamada para execução local de um EJB (apenas EJB 3.0)

- A chamada para execução remota é utilizada quando pretende invocar a lógica empresarial Java EE que não reside no mesmo servidor que a importação.  
Por exemplo, na seguinte figura, uma importação EJB utiliza a Remote Method Invocation pelo Internet InterORB Protocol (RMI/IIOP) para chamar um método EJB noutro servidor.

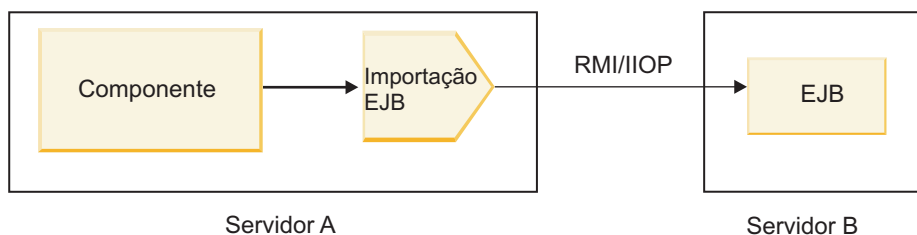


Figura 29. Chamada para execução remota de um EJB

Quando configura a associação EJB, o Integration Designer utiliza o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada para execução (local ou remota).

Associações de importação EJB contêm os seguintes componentes principais:

- rotina de tratamento de dados JAX-WS
- selector de falha EJB
- selector de função de importação EJB

Se o cenário de utilizador não for baseado no mapeamento JAX-WS, poderá necessitar de uma rotina de tratamento de dados, selector de função e selector de falha para executar as tarefas que são concluídas de outro modo pelos componentes que fazem parte das associações de importação EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento personalizado.

*Associações de exportação EJB:*

Aplicações externas Java EE podem chamar um componente SCA através de uma associação de exportação de um EJB. Através da utilização de uma exportação EJB é possível expor componentes SCA para que as aplicações externas Java EE possam chamar esses componentes para execução através da utilização do modelo de programação EJB.

**Nota:** A exportação EJB é um bean sem registo.

Utiliza o Integration Designer para criar associações EJB. Pode utilizar qualquer um dos seguintes procedimentos para gerar associações:

- Criar associações de exportação EJB utilizando o assistente do serviço externo

Pode utilizar o assistente do serviço externo no Integration Designer para criar um serviço de exportação EJB com base numa implementação existente. O assistente do serviço externo cria serviços com base em critérios fornecidos pelo utilizador. Em seguida, gera objectos de negócio, interfaces e ficheiros de exportação com base no serviços descobertos.

- Criar associações de exportação EJB utilizando o editor de montagem

Pode criar uma exportação EJB utilizando o editor de montagem do Integration Designer.

**Importante:** Um cliente Java 2 Platform, Standard Edition (J2SE) não consegue chamar o cliente de exportação EJB que é gerado no Integration Designer.

Pode gerar a associação a partir de um componente SCA existente ou pode gerar uma exportação com uma associação EJB para uma interface Java.

- Ao gerar uma exportação para um componente SCA existente que tenha uma interface WSDL existente, à exportação é atribuída uma interface Java interface.
- Ao gerar uma exportação para uma interface Java, pode seleccionar uma interface WSDL ou uma interface Java para a exportação.

**Nota:** Uma interface Java utilizada para criar uma exportação EJB tem as seguintes limitações no que diz respeito a objectos (parâmetros de entrada e de saída e excepções) transferidas como parâmetros numa chamada remota:

- Têm de ser de um tipo concreto (em vez de uma interface ou tipo abstracto).
- Têm de estar em conformidade com a especificação Enterprise JavaBeans. Têm de poder ser colocados em série e terem o construtor sem argumento predefinido, para que todas as propriedades possam estar acessíveis através de métodos de obtenção e definição.

Consulte o sítio da Web da Sun Microsystems, Inc., em <http://java.sun.com> para mais informações sobre a especificação JavaBeans.

Para além disso, a excepção tem de ser uma excepção verificada, herdada de `java.lang.Exception`, e tem de ser singular (ou seja, não suporta emitir vários tipos de excepção verificada).

Note também que a interface empresarial de um Java EnterpriseBean e uma interface Java simples e não deve estender-se a `javax.ejb.EJBObject` ou `javax.ejb.EJBLocalObject`. Os métodos de interface empresarial não devem apresentar uma excepção `java.rmi.RemoteException`.

As associações de exportação EJB podem interagir com a lógica empresarial Java EE através da utilização do modelo de programação EJB 2.1 ou do modelo de programação EJB 3.0.

A chamada para execução pode ser local (para EJB 3.0 apenas) ou remota.

- A chamada para execução local é utilizada quando a lógica empresarial Java EE chama um componente SCA residente no mesmo servidor que a exportação.
- A chamada para execução remota é utilizada quando a lógica empresarial Java EE business não reside no mesmo servidor que a exportação.

Por exemplo, na seguinte figura, um EJB utiliza RMI/IIOP para chamar um componente SCA num servidor diferente.



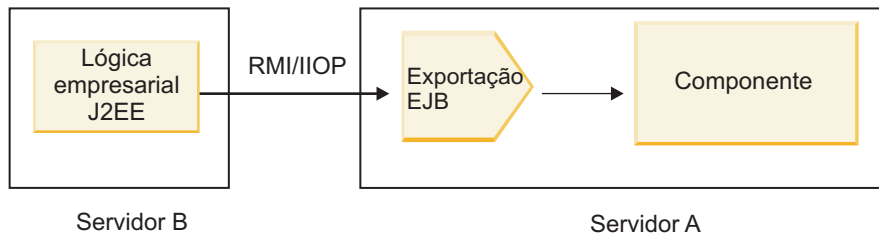


Figura 30. Chamada remota de um cliente para um componente SCA através de uma exportação EJB

Quando configura a associação EJB, o Integration Designer utiliza o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada para execução (local ou remota).

As associações de exportação EJB contêm os seguintes componentes principais:

- rotina de tratamento de dados JAX-WS
- selector de função de exportação EJB

Se o cenário de utilizador não for baseado no mapeamento JAX-WS, poderá necessitar de uma rotina de tratamento de dados e selector de função para executar as tarefas que são concluídas de outro modo pelos componentes que fazem parte das associações de exportação EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento personalizado.

*Propriedades de associação EJB:*

As associações de importação EJB utilizam os seus nomes JNDI configurados para determinar o nível de modelo de programação EJB e tipo de chamada para execução (local ou remota). As associações de importação e exportação EJB utilizam a rotina de tratamento de dados JAX-WS para conversão de dados. A associação de importação EJB utiliza um selector de função EJB e um selector de falha EJB, e a associação de exportação EJB utiliza um selector de função de exportação EJB.

*Nomes JNDI e associações de importação EJB:*

Ao configurar a associação EJB numa importação, o Integration Designer utiliza o nome JNDI para determinar o nível de programação EJB e tipo de chamada para execução (local ou remota).

Se não for especificado nenhum nome JNDI, é utilizada a associação de interface EJB predefinida. Os nomes predefinidos que são criados dependem de ser chamado o EJB 2.1 JavaBeans ou EJB 3.0 JavaBeans.

**Nota:** Consulte o tópico "descrição geral das associações da aplicação EJB 3.0" no Information Center WebSphere Application Server para obter informações mais detalhadas sobre as convenções de nomeação.

- EJB 2.1 JavaBeans

O nome JNDI predefinido seleccionado previamente pelo Integration Designer é a associação EJB 2.1 predefinida, que inclui o formato **ejb/** e acrescenta a interface da página inicial, separados por barras.

Por exemplo, para a interface da página inicial do EJB 2.1 JavaBeans para `com.mycompany.myremotebusinesshome`, a associação predefinida é:

```
ejb/com/mycompany/myremotebusinesshome
```

Para EJB 2.1 apenas é suportada a chamada para execução EJB remota.

- EJB 3.0 JavaBeans

O nome JNDI predefinido seleccionado previamente pelo Integration Designer para o JNDI local é o nome de classe completo da interface local precedido por **ejblocal:**. Por exemplo, para a interface completa da interface local `com.mycompany.mylocalbusiness`, o JNDI EJB 3.0 seleccionado previamente é:

```
ejblocal:com.mycompany.mylocalbusiness
```

Para a interface remota com.mycompany.myremotebusiness, o JNDI EJB 3.0 seleccionado previamente é a interface completa:

```
com.mycompany.myremotebusiness
```

As associações predefinidas EJB 3.0 são descritas na seguinte localização: Descrição geral das associações de aplicação EJB 3.0.

Integration Designer utilizará o nome "curto" como a localização JNDI predefinida para EJBs que utilizem o modelo de programação versão 3.0.

**Nota:** Se a referência JNDI implementada do EJB de destino for diferente da localização de associação JNDI predefinida, porque foi utilizado ou configurado um mapeamento personalizado, o nome JNDI de destino tem de ser especificado adequadamente. Pode especificar o nome no Integration Designer antes da implementação, ou , para a associação de importação, pode alterar o nome na consola administrativa (depois da implementação) para corresponder ao nome JNDI do EJB de destino.

Para obter mais informações sobre como criar associações EJB, consulte a secção dedicada a trabalhar com associações EJB no Information Center do Integration Designer.

#### *Rotina de tratamento de dados JAX-WS:*

A associação de importação do Enterprise JavaBeans (EJB) utiliza a rotina de tratamento de dados JAX-WS para transformar objectos de negócio de pedido em parâmetros de objecto Java e para transformar o valor de devolução do objecto Java no objecto de negócio de resposta. A associação de exportação EJB utiliza a rotina de tratamento de dados JAX-WS para transformar EJBs de pedido em objectos de negócio de pedido e para transformar o objecto empresarial de resposta num valor de devolução.

Esta rotina de tratamento de dados mapeia dados a partir da interface WSDL especificada por SCA para a interface Java EJB (e vice-versa) utilizando a especificação Java API for XML Web Services (JAX-WS) e a especificação Java Architecture for XML Binding (JAXB).

**Nota:** O suporte actual está restrito às especificações JAX-WS 2.1.1 e JAXB 2.1.3.

A rotina de tratamento de dados especificada no nível de associação EJB é utilizada para executar processamento de pedido, resposta, falha e excepção do tempo de execução.

**Nota:** Para falhas pode ser especificada uma rotina de tratamento de dados específica para cada falha através da especificação da propriedade de configuração `faultBindingType`. Isto substitui o valor especificado no nível de associação EJB.

A rotina de tratamento de dados JAX-WS é utilizada por predefinição quando a associação EJB tem uma interface WSDL. Esta rotina de tratamento de dados não pode ser utilizada para converter uma mensagem SOAP a representar uma chamada para execução JAX-WS num objecto de dados.

A associação de importação EJB utiliza uma rotina de tratamento de dados para converter um objecto de dados numa matriz Java Object (`Object[]`). Durante as comunicações externas, ocorre o seguinte processamento:

1. A associação EJB define o tipo esperado, elemento esperado e nome do método de destino no `BindingContext` (contexto da associação) para corresponder ao especificado no WSDL.
2. A associação EJB chama o método de conversão para o objecto de dados que requer a conversão de dados.
3. A rotina de tratamento de dados devolve um `Object[]` (objecto) que representa os parâmetros do método (na ordem da sua definição dentro do método).
4. A associação EJB utiliza o `Object[]` para chamar o método na interface EJB de destino.

A associação também prepara um `Object[]` para processar a resposta da chamada para execução EJB.

- O primeiro elemento no `Object[]` é o valor de devolução da chamada para execução de método Java.
- Os valores subsequentes representam os parâmetros de entrada para o método.

Isto é necessário para suportar os tipos de parâmetros In/Out e Out.

Para parâmetros do tipo Out, os valores têm de ser devolvidos no objecto de dados de resposta.

A rotina de tratamento de dados processa e converte valores encontrados no `Object[]` e depois devolve uma resposta ao objecto de dados.

A rotina de tratamento de dados suporta `xs:AnyType`, `xs:AnySimpleType` e `xs:Any`, juntamente com outros tipos de dados XSD. Para permitir suporte para `xs:Any`, utilize o `@XmlAnyElement (lax=true)` para a propriedade JavaBeans no código Java, tal como mostrado no exemplo seguinte:

```
public class TestType {
    private Object[] object;

    @XmlAnyElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Isto transforma o objecto da propriedade em `TestType` num campo `xs:any`. O valor da classe de Java utilizado no campo `xs:any` deve ter a anotação `@XmlAnyElement`. Por exemplo, se `Address` (endereço) for a classe de Java a ser utilizada para preencher a matriz do objecto, a classe `Address` deve ter a anotação `@XmlRootElement`.

**Nota:** Para personalizar o mapeamento dos tipos de Java XSD definido pela especificação JAX-WS, altere as anotações JAXB para se adequarem às necessidades do negócio. A rotina de tratamento de dados JAX-WS suporta `xs:any`, `xs:anyType` e `xs:anySimpleType`.

As seguintes restrições são aplicáveis à rotina de tratamento de dados JAX-WS:

- A rotina de tratamento de dados não inclui suporte para a anotação de atributo de cabeçalho `@WebParam`.
- O espaço de nomes para ficheiros de esquema de objectos de negócio (ficheiros XSD) não inclui mapeamento por predefinição a partir do nome do conjunto de programas Java. A anotação `@XMLSchema` em `package-info.java` também não funciona. A única forma de criar um XSD com um espaço de nomes é utilizar as anotações `@XmlType` e `@XmlRootElement`. `@XmlRootElement` define o espaço de nomes de destino para o elemento global em tipos de JavaBeans.
- O assistente de importação EJB não cria ficheiros XSD para classes não relacionadas. A versão 2.0 não suporta a anotação `@XmlSeeAlso`, por isso, se a classe descendente não estiver referenciada directamente a partir da classe ascendente, não é criado um XSD. A solução para este problema é executar `SchemaGen` para essas classes descendentes.

`SchemaGen` é um utilitário de linha de comandos (localizado no directório `WPS_Install_Home/bin`) fornecido para criar ficheiros XSD para um determinado bean. Esses XSDs têm de ser copiados manualmente para o módulo para que a solução funcione.

*Selector de falha EJB:*

O selector de falha EJB determina se uma chamada para execução EJB resultou numa falha, excepção de tempo de execução ou numa resposta com êxito.

Se for detectada uma falha, o selector de falhas EJB devolve o nome da nativo ao tempo de execução da associação, para que a rotina de tratamento de dados JAX-WS possa converter o objecto de excepção num objecto de negócio.

Em de resposta com êxito (não-avaria), a associação de importação EJB monta uma matriz de objecto Java (Object[]) para devolver os valores.

- O primeiro elemento no Object[] é o valor de devolução da chamada para execução de método Java.
- Os valores subsequentes representam os parâmetros de entrada para o método.

Isto é necessário para suportar os tipos de parâmetros In/Out e Out.

Para cenários de excepção a associação monta um Object[] e o primeiro elemento representa a excepção detectada pelo método.

O selector de falha pode devolver qualquer um dos seguintes valores:

*Tabela 35. Valores de devolução*

Tipo	Valor de devolução	Descrição
Falha	<b>ResponseType.FAULT</b>	Devolvido quando o objecto Object[] transferido contém um objecto de excepção.
Excepção do tempo de execução	<b>ResponseType.RUNTIME</b>	Devolvido se o objecto de excepção não corresponder a qualquer um dos tipos de excepção declarados no método.
Resposta normal	<b>ResponseType.RESPONSE</b>	Devolvido em todos os outros casos.

Se o selector de falha devolver um valor de **ResponseType.FAULT**, o nome de falha nativo é devolvido. Este nome de falha é utilizado pela associação para determinar o nome de falha WSDL correspondente do modelo e para chamar a rotina de tratamento de dados correcta.

*Selector de função EJB:*

As associações EJB utilizam um selector de função de importação (para processamento externo) ou um selector de função de exportação (para processamento interno) para determinar o método EJB a invocar.

### **Selector de função de importação**

Para processamento externo, o selector de função de importação deriva o tipo de método EJB baseado no nome da operação chamada pelo componente SCA que está ligado à importação EJB. O selector de função procura a anotação @WebMethod na classe Java anotada no JAX-WS gerado no Integration Designer para determinar o nome da operação de destino associada.

- Se a anotação @WebMethod estiver presente, o selector de função utiliza a anotação @WebMethod para determinar o mapeamento de método de Java correcto para o método WSDL.
- Se a anotação @WebMethod estiver em falta, o selector de função assume que o nome do método de Java é o mesmo do nome da operação chamado.

**Nota:** Este selector de função é válido apenas para uma interface do tipo WSDL numa importação EJB, não é válido para uma interface do tipo de Java numa importação EJB.

O selector de função devolve um objecto java.lang.reflect.Method que representa o método da interface EJB.

O selector de função utiliza uma matriz de Java Object (Object[]) para conter a resposta do método de destino. O primeiro elemento no Object[] é um método de Java com o nome do WSDL, e o segundo elemento no Object[] é o objecto de negócio de entrada.

## Selector de função de exportação

Para processamento interno, o selector de função de exportação deriva o método de destino a ser chamado a partir do método de Java.

O selector de função de exportação mapeia o nome da operação de Java chamado pelo cliente EJB para o nome da operação na interface do componente de destino. O nome do método é devolvido como uma cadeia e resolvido pelo tempo de execução de SCA dependendo do tipo de interface do componente de destino.

### Associações EIS:

As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é conseguida utilizando exportações EIS e importações EIS que suportem adaptadores de recurso JCA 1.5 e Websphere Adapters.

Os componentes SCA podem requerer que os dados sejam transferidos para ou de um EIS externo. Quando cria um módulo SCA que requer essa conectividade, incluirá (para além do componente SCA) uma importação ou exportação com uma associação EIS para comunicação com um EIS externo específico.

Os adaptadores de recurso no IBM Integration Designer são utilizados dentro do contexto de uma importação ou uma exportação. Desenvolve uma importação ou uma exportação com o assistente do serviço externo e, ao desenvolver, inclui o adaptador de recurso. Uma importação EIS que lhe permite chamar um serviço num sistema EIS, ou uma exportação que permite que uma aplicação num sistema EIS chame um serviço desenvolvido no IBM Integration Designer, são criadas com um adaptador de recurso. Por exemplo, criaria uma importação com o adaptador JD Edwards para chamar um serviço no sistema JD Edwards.

Quando utiliza o assistente do serviço externo, a informação da associação EIS é criada. Também pode utilizar outra ferramenta, o editor de montagem, para adicionar ou alterar a informação da associação. Consulte o tópico *Accessing external services with adapters* (Aceder a serviços externos com adaptadores) para mais informações.

Depois do módulo SCA que contém a associação EIS ser implementado no servidor, pode utilizar a consola administrativa para ver informações sobre a associação ou para configurar a associação.

### *Descrição geral das associações EI:*

A associação do Enterprise Information System (EIS), quando utilizada com um adaptador de recurso JCA, permite-lhe aceder a serviços num sistema de informação empresarial ou tornar os serviços disponíveis no EIS.

O exemplo seguinte mostra de que forma um módulo SCA chamado ContactSyncModule sincroniza informações de contacto entre um sistema Siebel e um sistema SAP.

1. O componente SCA chamado ContactSync escuta (através de uma exportação de aplicação EIS chamada Siebel Contact) por alterações para contactos Siebel.
2. O componente SCA ContactSync em si faz uso de uma aplicação SAP (através de uma importação de aplicação EIS) para actualizar respectivamente a informação de contacto SAP de forma.

Como as estruturas de dados utilizadas para armazenar contactos são diferentes nos sistemas Siebel e SAP, o componente SCA ContactSync tem de fornecer um mapeamento.

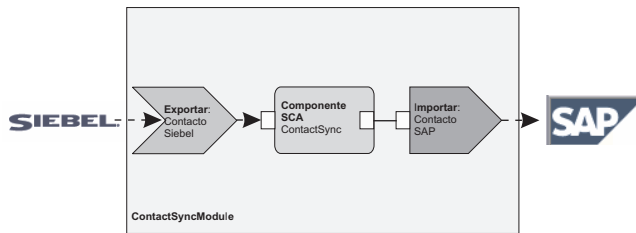


Figura 31. Fluxo de um sistema Siebel para um sistema SAP

A exportação de contactos Siebel e a importação de contactos SAP têm os adaptadores de recurso apropriados configurados.

*Funções-chave de associações ao EIS:*

A importação EIS é uma importação do módulo Arquitectura do Componente de Serviço (SCA, Service Component Architecture) que permite que componentes desse mesmo módulo utilizem aplicações do EIS definidas fora do módulo SCA. É utilizada uma importação EIS para transferir dados do componente SCA para um EIS externo; é utilizada uma exportação EIS para transferir dados de um EIS externo para um módulo SCA.

## Importações

A função da importação EIS é estabelecer a comunicação entre componentes SCA e sistemas de EIS externo. As aplicações externas podem ser vistas como uma importação EIS. Neste caso, a importação EIS envia dados para o EIS externo e possui a opção de receber dados como resposta.

A importação EIS fornece componentes SCA com uma apresentação uniforme das aplicações exteriores ao módulo. Isto permite que os componentes comuniquem com um EIS externo, tal como o SAP, Siebel ou PeopleSoft, utilizando um modelo SCA consistente.

No lado da importação relativo ao cliente, existe uma interface, exposta pela aplicação da importação EIS, com um ou mais métodos, cada um deles considerando os objectos de dados como argumentos e valores devolvidos. No lado da implementação, existe uma Interface de Cliente Comum (CCI, Common Client Interface CCI) implementada por um adaptador de recurso.

A implementação do tempo de execução de uma importação EIS liga a interface do lado do cliente à CCI. A importação mapeia a chamada para execução do método pertencente à interface para a chamada para execução no CCI.

As associações são criadas em três níveis: a associação da interface, que depois utiliza as associações de métodos contidas, que, por sua vez, utilizam associações de dados.

A associação da interface relaciona a interface da importação com a ligação ao sistema EIS que fornece a aplicação. Isto reflecte o facto de o conjunto de aplicações, representado pela interface, ser fornecido pela ocorrência específica do EIS e de a ligação proporcionar o acesso a esta ocorrência. O elemento de associação contém propriedades com informação suficiente para criar a ligação (estas propriedades fazem parte da ocorrência `javax.resource.spi.ManagedConnectionFactory`).

A associação de métodos associa o método que contém a interacção específica com o sistema EIS. Para a Arquitectura de Criptografia Java (JCA, Java Cryptography Architecture), a interacção é caracterizada pelo conjunto de propriedades da implementação da interface `javax.resource.cci.InteractionSpec`. O elemento de interacção da associação de métodos contém estas propriedades, juntamente com o nome da classe, proporcionando assim informação suficiente para efectuar a interacção. A associação de métodos utiliza associações de dados que descrevem à representação EIS o mapeamento do argumento e resultado do método da interface.



O modelo do tempo de execução para a importação EIS é o seguinte:

1. O método da interface de importação é chamado através da utilização do modelo de programação SCA.
2. O pedido, ao chegar à importação EIS, contém o nome do método e respectivos argumentos.
3. A importação cria primeiro uma implementação de associação de interface; depois, utilizando dados da associação de importação, cria uma `ConnectionFactory`, associando ambas. Ou seja, a importação pede o comando `setConnectionFactory` à associação da interface.
4. É então criada a implementação da associação de métodos correspondente ao método chamado.
5. A ocorrência `javax.resource.cci.InteractionSpec` é criada e preenchida; de seguida, são utilizadas associações de dados para associar os argumentos do método a um formato que seja entendido pelo adaptador de recurso.
6. A interface CCI é utilizada para executar a interação.
7. Quando o pedido regressa, a associação de dados é utilizada para criar o resultado da chamada, sendo o resultado devolvido ao autor do pedido.

## Exportações

A função da exportação EIS é estabelecer a comunicação entre componentes SCA e sistemas de EIS externos. As aplicações externas podem ser vistas como uma exportação EIS. Neste caso, a aplicação externa envia os dados na forma de notificações periódicas. Pode-se pensar numa exportação EIS como se fosse uma aplicação de subscrição que recebe um pedido externo de um EIS. O componente SCA que utiliza a exportação EIS trata-a como uma aplicação local.

A exportação EIS proporciona aos componentes SCA uma vista uniforme das aplicações exteriores ao módulo. Isto permite que os componentes comuniquem com um EIS, como o SAP, Siebel ou PeopleSoft, utilizando um modelo SCA consistente.

A exportação apresenta uma implementação de serviço de escuta, que recebe pedidos do EIS. O serviço de escuta implementa um serviço de escuta específico para o adaptador de recurso. A exportação contém também uma interface de implementação de componentes, exposta ao EIS através da exportação.

A implementação do tempo de execução de uma exportação EIS liga o serviço de escuta à interface de implementação do componente. A exportação mapeia o pedido feito pelo EIS para a chamada da operação adequada do componente. As associações são criadas em três níveis: a associação do sistema de escuta, que depois utiliza um método de associação nativo, que, por sua vez, utiliza uma associação de dados.

A associação do sistema de escuta relaciona o sistema de escuta que recebe os pedidos com o componente exposto através da exportação. A definição da exportação contém o nome do componente, que o tempo de execução localiza e para o qual reencaminha os pedidos.

A associação do método nativo associa o método nativo ou o tipo de evento recebido pelo serviço de escuta, à operação implementada pelo componente exposto por via da exportação. Não existe qualquer relação entre o método chamado no sistema de escuta e o tipo de evento; todos os eventos chegam através de um ou mais métodos do serviço de escuta. A associação do método nativo utiliza o selector de função definido na exportação para extrair o nome do método nativo dos dados internos e associações de dados para associar o formato de dados do EIS a um formato entendido pelo componente.

O modelo do tempo de execução para uma exportação EIS é o seguinte:

1. O pedido de EIS desencadeia a chamada do método na implementação do sistema de escuta.
2. O serviço de escuta localiza e chama a exportação, transferindo para esta todos os argumentos de chamada.
3. A exportação cria a implementação da associação do serviço de escuta.



4. A exportação instancia o selector de função e define-o na associação do sistema de escuta.
5. A exportação inicializa as associações do método nativo e adiciona-as à associação do sistema de escuta. Para cada associação do método nativo, são também inicializadas as associações de dados.
6. A exportação chama a associação do sistema de escuta.
7. A associação do sistema de escuta localiza componentes exportados e utiliza o selector de função para obter o nome do método.
8. Este nome é utilizado para localizar a associação do método nativo, que depois chama o componente-destino.

O estilo de interacção do adaptador permite que a associação da exportação EIS chame o componente-destino de forma assíncrona (a forma predefinida) ou síncrona.

### **Adaptadores de recurso**

Ao desenvolver uma importação ou exportação com o assistente do serviço externo, está a incluir um adaptador de recurso. Os adaptadores que vêm com IBM Integration Designer utilizados para aceder aos sistemas CICS, IMS, JD Edwards, PeopleSoft, SAP e Siebel destinam-se apenas a fins de desenvolvimento e teste. Isto significa que os irá utilizar para desenvolver e testar as suas aplicações.

Uma vez implementada a sua aplicação, irá precisar de adaptadores de tempo de execução licenciados para a executar. No entanto, ao criar o seu serviço, é possível incorporar o adaptador no mesmo. O licenciamento do adaptador pode permitir-lhe utilizar o adaptador incorporado como adaptador de tempo de execução licenciado. Estes adaptadores estão em conformidade com a Java EE Connector Architecture (JCA 1.5). A JCA, um padrão aberto, é o padrão Java EE para a conectividade EIS. A JCA fornece um contexto gerido, ou seja, a Qualidade de Serviço (QoS) é fornecida pelo servidor da aplicação que oferece a gestão do ciclo de vida e segurança às transacções. Estão também em conformidade com a especificação Identificação de metadados de empresa, com a excepção do Adaptador de recurso IBM CICS ECI e do IBM IMS Connector for Java.

O WebSphere Business Integration Adapters, um conjunto mais antigo de adaptadores, também é suportado pelo assistente.

### **Recursos Java EE**

O módulo EIS module, um módulo SCA que segue o padrão do módulo EIS, pode ser implementado na plataforma Java EE.

A implementação do módulo EIS na plataforma Java EE resulta numa aplicação pronta a ser iniciada, compactada como ficheiro EAR e implementada no servidor. São criados todos os artefactos e recursos Java EE, a aplicação é configurada e fica pronta a ser executada.

#### *Propriedades dinâmicas da Especificação de interacção JCA e Especificação de ligação JCA:*

A associação EIS pode aceitar a entrada para as InteractionSpec e ConnectionSpec especificadas, utilizando um objecto de dados descendente que acompanha a carga útil. Isto permite interacções dinâmicas de pedido-resposta com um adaptador de recurso através da InteractionSpec e autenticação de componente através da ConnectionSpec.

A `javax.cci.InteractionSpec` traz informação sobre a forma como o pedido de interacção com o adaptador de recurso deve ser processado. Também pode trazer informações sobre a forma como a interacção foi alcançada depois do pedido. Estas comunicações bidireccionais através das interacções são referidas, por vezes, como *conversações*.

A associação EIS espera que a carga útil, que será o argumento para o adaptador de recurso, contenha um objecto de dados descendente chamado **propriedades**. Este objecto de dados de propriedades vai conter pares nome-valor, com o nome das propriedades da especificação de interacção num formato específico. As regras de formatação são:

- Os nomes têm de começar pelo prefixo **IS**, seguido pelo nome da propriedade. Por exemplo, uma especificação de interacção com uma propriedade JavaBeans chamada **InteractionId** especifica o nome da propriedade como **ISInteractionId**.
- O par nome-valor representa o nome e o valor do tipo simples da propriedade de especificação de interacção.

Neste exemplo, uma interface específica que a entrada de uma operação é um **Account** objecto de dados. Esta interface chama uma aplicação de associação de importação EIS com a intenção de enviar e receber uma propriedade InteractionSpec dinâmica chamada **workingSet** com o valor **xyz**.

Os gráficos comerciais ou objectos de negócio no servidor contêm um objecto de negócio de propriedades **properties (propriedades)** subjacente que permite o envio de dados específicos de protocolo com a carga útil. Este objecto de negócio de **propriedades** está integrado e não necessita de ser especificado no esquema XML ao construir um objecto de negócio. Só precisa de ser criado e utilizado. Se tiver os seus próprios tipos de dados definidos com base num esquema XML, necessita de especificar um elemento de **properties** que contenha o seu par nome-valor esperado.

```
BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Conector para interfaces de estilo translineadas doc-lit,
//avançar para carga útil para não doc-lit
DataObject docLitWrapper = dataFactory.createByElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Criar a carga útil.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Execute a definição da carga útil
```

```
//Construir dados de propriedades para interacção dinâmica
```

```
DataObject properties = account.createDataObject("properties");
```

Para o nome **workingSet**, defina o valor esperado(**xyz**).

```
properties.setString("ISworkingSet", "xyz");
```

```
//Chamar o serviço com argumento
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Obter propriedade devolvida
```

```
DataObject retProperties = result.getDataObject("properties");
```

```
String workingset = retProperties.getString("ISworkingSet");
```

Pode utilizar as propriedades ConnectionSpec para uma autenticação de componente dinâmica. Aplicam-se as mesmas regras, excepto que o prefixo do nome da propriedade precisa de ser **CS** (em vez de **IS**). As propriedades ConnectionSpec não são bidireccionais. O mesmo objecto de dados de **propriedades** pode conter propriedades IS e CS.

Para utilizar propriedades ConnectionSpec, defina o **resAuth** especificado na associação de importação para **Aplicação**. Certifique-se também de que o adaptador de recurso suporta a autorização de componente. Consulte o capítulo 8 da Especificação de arquitectura de Conector J2EE para mais detalhes.

#### *Clientes externos com associações EIS:*

O servidor pode enviar mensagens a, receber mensagens de clientes externos utilizando associações EIS.

Um cliente externo, por exemplo um portal da web ou um EIS, necessita de enviar uma mensagem para um módulo SCA para o servidor ou necessita de ser chamado por um componente a partir do servidor.

O cliente chama a importação EIS com qualquer outra aplicação, utilizando a Interface de chamada dinâmica (DII, Dynamic Invocation Interface) ou a interface Java.

1. O cliente externo cria uma instância do ServiceManager e procura a importação EIS utilizando o seu nome de referência. O resultado da procura é a implementação da interface de serviço.
2. O cliente cria um argumento de entrada, um objecto de dados genérico, criado dinamicamente, utilizando o esquema de objecto de dados. Este passo é efectuado utilizando a implementação de interface DataFactory do objecto de dados de serviço.
3. O cliente externo chama o EIS e obtém os resultados pedidos.

Outra opção é o cliente poder chamar a importação EIS utilizando a interface Java.

1. O cliente cria uma instância do ServiceManager e procura a importação EIS utilizando o seu nome de referência. O resultado da procura é uma interface Java da importação EIS.
2. O cliente cria um argumento de entrada e o objecto de dados escrito.
3. O cliente chama o EIS e obtém os resultados pedidos.

A interface de exportação EIS define a interface do componente SCA exportado que está disponível para as aplicações EIS externas. Esta interface pode ser imaginada como a interface que uma aplicação externa (tal como SAP ou PeopleSoft) irá chamar através da implementação do tempo de execução da aplicação de exportação EIS.

A exportação utiliza EISExportBinding para associar serviços exportados à aplicação EIS externa. Permite que o utilizador subscreva uma aplicação contida no seu módulo SCA para escutar pedidos de serviço EIS. A associação de exportação EIS especifica o mapeamento entre a definição de eventos de entrada como é compreendida pelo adaptador de recurso (utilizando interfaces do Java EE Connector Architecture) e a chamada de operações SCA.

O EISExportBinding necessita de serviços EIS externos para ser basearem no nos contratos de entrada do Java EE Connector Architecture 1.5. O EISExportBinding necessita que uma rotina de tratamento de dados ou associação de dados seja especificada ao nível da associação ou ao nível do método.

#### **Associações JMS:**

Um fornecedor de Serviço de Mensagens Java (JMS - Message Service) permite mensagens com base na API de Java Messaging Service API e modelo de programação. Disponibiliza fornecedores de ligação JMS para criar ligações para destinos JMS e para enviar e receber mensagens.

As associações JMS podem ser utilizadas quando interage com a associação de fornecedor de barramento de integração de serviço (SIB, Service Integration Bus) e são compatíveis com JMS e JCA 1.5.

Pode utilizar as associações de importação e de exportação JMS e um módulo de arquitectura de componente de serviço (SCA) para chamar e receber mensagens de sistemas JMS externos.

As associações de importação e exportação JMS fornecem integração com aplicações JMS utilizando o fornecedor de JMS SIB baseado em JCA 1.5 que faz parte do WebSphere Application Server. Não são suportados outros adaptadores de recursos JMS baseados em JCA 1.5

*Descrição geral de associações JMS:*

As associações JMS fornecem conectividade entre o ambiente de Arquitectura do Componente de Serviço (SCA, Service Component Architecture) e sistemas de JMS.

### **Associações JMS**

Os principais componentes de ambas as associações de importação JMS e de exportação JMS são:

- Adaptador de recurso: permite conectividade bidireccional gerida entre um módulo SCA e sistemas JMS externos
- Ligações: encapsulam uma ligação virtual entre uma aplicação cliente e uma aplicação fornecedora
- Destinos: utilizados por um cliente para especificar o destino de mensagens que produz ou a origem de mensagens que consome
- Dados de autenticação: utilizados para proteger o acesso à associação

### **Funções-chave de associações JMS**

#### **Cabeçalhos especiais**

As propriedades de cabeçalho especial são utilizadas em importações e exportações JMS para dizer ao destino como processar a mensagem.

Por exemplo, TargetFunctionName faz a correlação a partir do método nativo para o método de operação.

#### **Recursos Java EE**

É criado um número de recursos Java EE quando as importações e exportações JMS são implementadas no ambiente Java EE.

#### **ConnectionFactory**

Utilizado por clientes para criar uma ligação ao fornecedor JMS.

#### **ActivationSpec**

As importações utilizam este comando para receber a resposta a um pedido; as exportações usam-no ao configurarem os pontos finais da mensagem que representam serviços de escuta da mensagem nas suas interacções com o sistema de mensagens.

#### **Destinos**

- Destino de envio: numa importação, é para onde é enviado o pedido ou mensagem de saída; numa exportação, é o destino para onde será enviada a mensagem de resposta, caso o campo de cabeçalho JMSReplyTo não se lhe sobreponha na mensagem de entrada.
- Destino de recepção: onde deve ser colocada a mensagem de entrada; nas importações, trata-se de uma resposta; nas exportações, trata-se de um pedido.
- Destino da chamada de retorno: destino do sistema SCA JMS utilizado para armazenar informações de correlação. Não utilizar este destino para leitura nem escrita.

A tarefa de instalação cria a ConnectionFactory e três destinos. Cria também o ActivationSpec para permitir que o serviço de escuta da mensagem do tempo de execução capte respostas no destino de recepção. As propriedades destes recursos vêm especificadas no ficheiro de importação ou exportação.

*Integração JMS e adaptadores de recurso:*

O Fornecedor de Serviços Java (JMS - Java Message Service) fornece integração através de um adaptador de recurso baseado em JCA 1.5 JMS disponível. O suporte completo para a integração JMS é fornecido para o adaptador de recurso JMS de barramento de integração de serviço (SIB - Service Integration Bus).

Utilize um fornecedor JMS para adaptador de recurso JCA 1.5 quando quiser integrar com um sistema JMS externo compatível com JCA 1.5. Os serviços externos compatíveis com JCA 1.5 podem receber mensagens e enviar mensagens para integrarem com componentes da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) utilizando o adaptador de recurso JMS SIB.

A utilização de outros adaptadores de recurso JCA 1.5 específicos de fornecedor não é suportada.

*Associações de importação e exportação JMS:*

Pode fazer com que os módulos SCA interajam com serviços fornecidos por aplicações JMS externas através da utilização de associações de importação e exportação JMS.

### **Associações de importação JMS**

As ligações ao fornecedor de JMS associado de destinos JMS são criadas através da utilização de um fornecedor de ligação JMS. Utilize objectos administrativos de fornecedor de ligação para gerir fornecedores de ligação JMS para o fornecedor de serviço de mensagens predefinido.

A interacção com sistemas JMS externos inclui a utilização de destinos para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação de JMS, dependendo do tipo de operação chamada:

- Unidireccional: A importação JMS coloca uma mensagem no destino de envio configurado na associação de importação. Não está definido nada no campo replyTo do cabeçalho JMS.
- Bidireccionalidade (resposta ao pedido): A importação JMS coloca uma mensagem no destino de envio e depois mantém a resposta que recebe do componente SCA.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação de mensagem de pedido. A associação de importação também pode ser configurada para utilizar um destino de resposta dinâmico para correlacionar respostas com pedidos. É criado um destino temporário para cada pedido e a importação utiliza esse destino para receber a resposta.

O destino receive é definido na propriedade do cabeçalho replyTo da mensagem de saída. É implementada uma escuta de mensagem para escutar o destino de recepção, e quando uma mensagem é recebida, a escuta de mensagem passa a resposta de volta para o componente.

É possível especificar propriedades de cabeçalho dinâmicas e estáticas para ambos os cenários de utilização. As propriedades estáticas podem ser definidas a partir da associação de método de importação JMS. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 32 na página 119 ilustra a forma como a importação está ligada ao serviço externo.

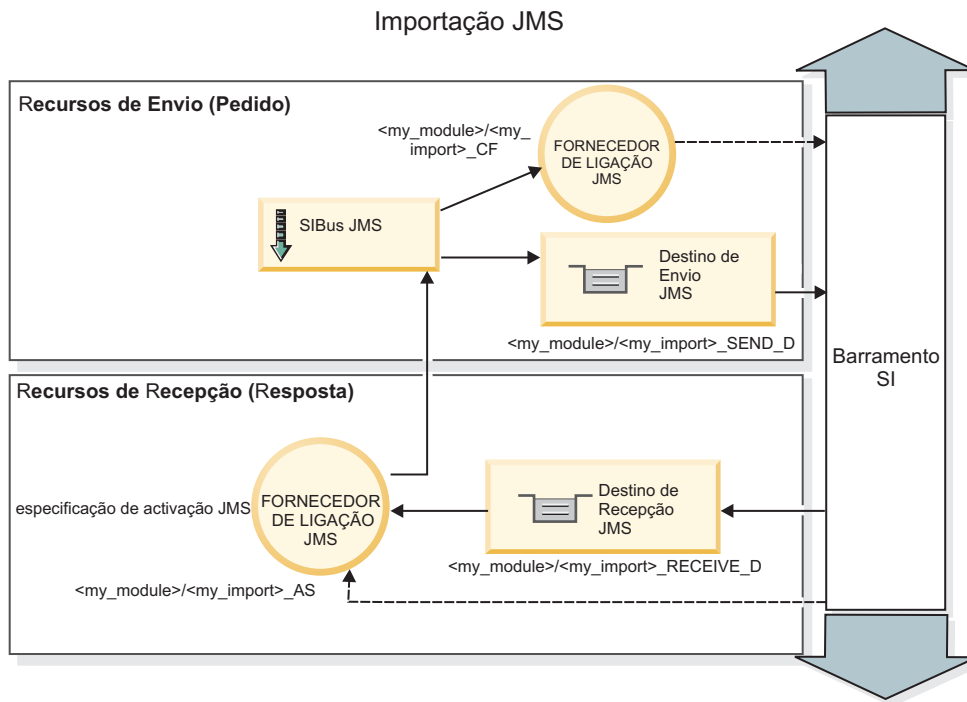


Figura 32. Recursos de associação de importação JMS

### Associações de exportação JMS

As associações de exportação JMS fornecem os meios para os módulos SCA fornecerem serviços para aplicações JMS externas.

A ligação que faz parte de uma exportação JMS é uma especificação de activação configurável.

Uma exportação JMS tem destinos de envio e de recebimento.

- O destino receive é onde a mensagem recebida do componente de destino deve ser colocada.
- O destino send é onde a resposta será enviada, excepto se a mensagem recebida o tiver substituído utilizando a propriedade de cabeçalho replyTo.

Uma escuta de mensagem é implementada para escutar pedidos de entrada no destino receive especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. O destino especificado no campo replyTo da mensagem recebida substitui o destino especificado em send.

A Figura 33 na página 120 ilustra como o solicitador externo está ligado à exportação.

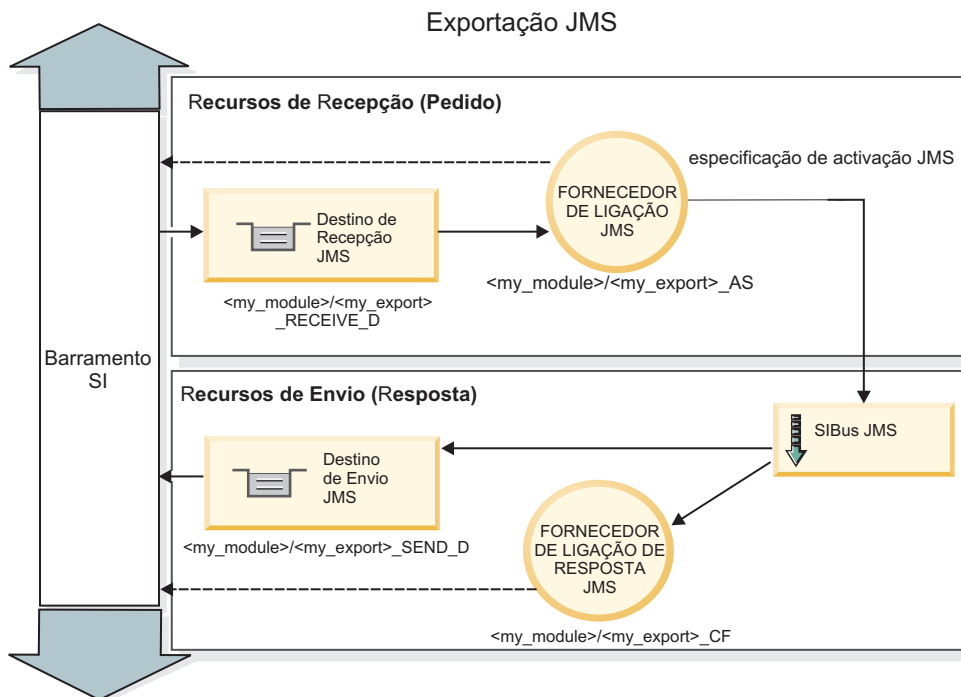


Figura 33. Recursos de associação de exportação JMS

#### Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

#### Cabeçalho de sistema JMS

O cabeçalho de sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos encontrados tipicamente num cabeçalho JMS. Apesar de estes poderem ser modificados na mediação (ou ContextService), alguns campos de cabeçalho de sistema JMS definidos no SMO não serão propagados na mensagem JMS de saída, uma vez que são substituídos por valores de sistema ou estáticos.

Os campos de chave no cabeçalho de sistema JMS que podem ser actualizados numa mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidos específicos
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não-persistente, a predefinição é persistente)
- **JMSPriority** - valor de prioridade (0 a 9, a predefinição é JMS\_Default\_Priority)

#### Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista Propriedades. As propriedades podem ser adicionadas, actualizadas ou eliminadas numa mediação ou através da utilização da API ContextService.



As propriedades também podem ser definidas estaticamente na associação JMS. As propriedades que estiverem definidas estaticamente substituem definições (com o mesmo nome) que estejam definidas dinamicamente.

As propriedades de utilizador propagadas de outras associações (por exemplo, uma associação HTTP) serão produzidas na associação JMS como propriedades JMS.

### Definições de propagação de cabeçalho

A propagação do cabeçalho de sistema JMS e propriedades quer da mensagem JMS interna para componentes na direcção do fluxo ou de componentes na direcção oposta ao fluxo para a mensagem JMS de saída podem ser controlados pelo sinalizador Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) na associação.

Quando o Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) está definido, a informação de cabeçalho pode fluir para a mensagem ou para o componente de destino, tal como descrito na lista seguinte:

- Pedido de exportação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.
- Resposta de exportação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de exportação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de exportação JMS.
- Pedido de importação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de importação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de importação JMS.
- Resposta de importação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.

*Esquema de correlação de destino de resposta dinâmico temporário JMS:*

O esquema de correlação de destino de resposta dinâmico temporário dá origem a uma fila dinâmica única ou tópico a ser criado para cada pedido enviado.

O destino de resposta estático especificado na importação é utilizado para derivar a natureza da fila de destino dinâmico temporário ou tópico. Isto é definido no campo **ReplyTo (Responder a)** do pedido e a importação JMS escuta por respostas nesse destino. Quando a resposta é recebida, ela é recolocada na fila do destino de resposta estática para processamento assíncrono. O campo **CorrelationID** da resposta não é utilizado e não precisa de ser definido.

### Questões transaccionais

Quando um destino dinâmico temporário está em utilização, a resposta tem de ser consumida no mesmo módulo que o da resposta enviada. O pedido tem de ser enviado fora da transacção global e tem de ser consolidado antes de ser recebido pelo serviço do programa emissor para que uma resposta tem seja devolvida.

## Persistência

As filas dinâmicas temporárias são entidades de duração breve e não garantem o mesmo nível de persistência associado a uma fila estática ou tópico. Uma fila dinâmica temporária ou tópico não sobrevivem a um reinício de servidor nem as mensagens. Depois de uma mensagem ter sido recolocada na fila do destino de resposta estático, ela retém a persistência definida na mensagem.

## Tempo limite

A importação espera para receber a resposta no destino de resposta dinâmico temporário durante um tempo fixo. Este intervalo de tempo é retirado do qualificador do momento de expiração de resposta SCA, se este estiver definido, de outro modo o tempo utilizado como predefinição é de 60 segundos. Se o tempo de espera for excedido, a importação apresenta uma `ServiceTimeoutRuntimeException`.

### *Cientes externos:*

O servidor pode enviar mensagens a ou receber mensagens de clientes externos utilizando associações JMS.

Um cliente externo (como um portal da web ou um sistema de informação empresarial) pode enviar uma mensagem a um módulo SCA no servidor ou pode ser chamado por um componente a partir do servidor.

Os componentes de exportação JMS fornecem escutas de mensagem para escutar pedidos que entram no destino especificado na associação de exportação. O destino especificado no campo de envio é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações com a associação de exportação.

As importações JMS interagem com clientes externos ao enviar mensagens e ao receber mensagens de filas JMS.

### *Trabalhar com clientes externos:*

Um cliente externo (isto é, que está fora do servidor) poderá precisar de interagir com uma aplicação instalada no servidor.

Imagine um cenário muito simples, em que o cliente externo quer interagir com uma aplicação do servidor. O exemplo ilustra um cenário típico muito simples.

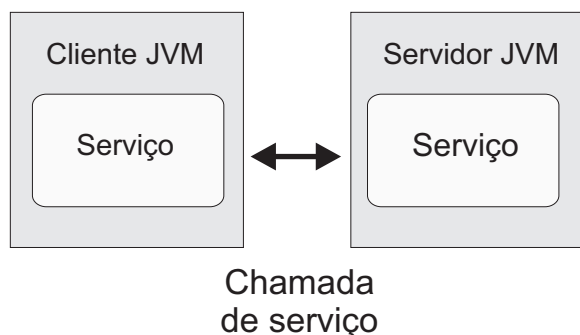


Figura 34. Cenário de um caso de uso simples: o cliente externo interage com a aplicação de servidor.

A aplicação SCA inclui uma exportação com uma associação JMS; o que faz com que a aplicação seja disponibilizada a clientes externos.

Quando tem um cliente externo numa máquina virtual Java (JVM virtual machine) separada do seu servidor, existem vários passos a tomar, de modo a estabelecer uma ligação e interagir com a exportação JMS. O cliente obtém um InitialContext (Contexto Inicial) com os valores correctos e, depois, pesquisa os recursos no JNDI. Nessa altura, o cliente utiliza a JMS 1.1 especificação do cliente para aceder a destinos, e às mensagens enviadas e recebidas nos destinos.

Os nomes JNDI predefinidos dos recursos criados automaticamente pelo tempo de execução estão catalogados no tópico de configuração desta secção. Contudo, se criou recursos previamente, utilize esses nomes de JNDI.

1. Configure os destinos JMS e o fornecedor de ligação para enviar mensagens.
2. Certifique-se de que o contexto JNDI, a porta do adaptador de recurso SIB, e a porta de envio de mensagens estão correctas.

O servidor utiliza algumas portas predefinidas mas, caso existam mais servidores instalados nesse sistema, são criadas portas alternadas na hora da instalação, para evitar conflitos com ocorrências de outros servidores. Pode utilizar a consola administrativa para determinar que portas o seu servidor está a utilizar. Aceda a **Servers (Servidores) > Application Servers (Servidores de Aplicação) > your\_server\_name > Configuration (Configuração)** e fazer clique **Ports (Portas)** sob **Communication (Comunicação)**. Em seguida, pode editar a porta que está a ser utilizada.

3. O cliente obtém um contexto inicial com os valores correctos e, depois, pesquisa os recursos no JNDI.
4. Ao utilizar as especificações JMS 1.1, o cliente acede a destinos, e às mensagens enviadas e recebidas nos destinos.

*Resolução de problemas de associação JMS:*

Poderá diagnosticar e corrigir problemas através de associações JMS.

### Excepções de implementação

Em resposta a várias condições de erro, a implementação de importações e exportações JMS poderá devolver um ou dois tipos de excepções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter o original excepção (JMSException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

### Principais condições de erro

As principais condições de erro das associações JMS são determinadas pela semântica transaccional, pela configuração do fornecedor JMS, ou por referência a um comportamento existente noutros componentes. As principais condições de erro incluem:

- Erro ao ligar-se ao fornecedor JMS ou ao destino.  
Erro ao ligar-se ao fornecedor JMS para receber mensagens irá resultar num erro ao iniciar a escuta de mensagens. Esta condição será registada no registo do WebSphere Application Server. Mensagens que persistam irão permanecer no destino, até serem obtidas com sucesso (ou até expirarem).  
Um erro ao ligar-se ao fornecedor JMS, para enviar mensagens de saída, irá causar a remoção de alterações da transacção que controla o envio.
- Um erro ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.

Um erro na associação de dados, ou na rotina de tratamento de dados, causa uma remoção de alterações do controlo de transacção do funcionamento.

- Um erro no envio da mensagem de saída.

Um erro no envio de uma mensagem causa uma remoção de alterações da transacção relevante.

- Mensagens de resposta tardias múltiplas ou inesperadas.

A importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Também o período de hora válida, em que se pode receber uma resposta, é determinado qualificador do Limite de Validade da Resposta SCA no pedido. Quando chega uma resposta, ou quando é ultrapassado o momento de expiração, apaga-se a gravação da correlação. Caso as mensagens de resposta cheguem de forma inesperada, ou tardia, é activada a Excepção de tempo de execução do serviço.

- A excepção do tempo de execução de espera do serviço causado por resposta tardia aquando da utilização do esquema de correlação do destino da resposta de dinâmica temporária.

O tempo de espera da importação JMS tem um período de tempo determinado pelo qualificador do limite de validade da resposta SCA ou, caso isto não esteja definido, passará a estar predefinido dentro de 60 segundos.

### As mensagens JMS baseadas em SCA que não aparecem no gestor de eventos falhado.

Se as mensagens SCA originadas através de uma interacção JMS falharem, poderá encontrá-las no gestor de eventos falhados. Se estas mensagens não forem apresentadas no gestor de eventos falhados, certifique-se de que o destino SIB subjacente ao destino JMS tem um valor máximo de entregas falhadas superior a 1. Definir este valor para 2, ou superior, activa a interacção com o gestor de eventos falhados, durante as chamadas para execução SCA para as associações JMS.

#### *Excepções de processamento:*

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal excepção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

Quando qualquer excepção de tempo de execução, incluindo uma **DataBindingException**, é lançada:

- Se o fluxo de mediação está configurado para ser transaccional, a mensagem JMS, por predefinição, é guardada no Gestor de eventos falhados para repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no Gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a excepção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer excepção da rotina de tratamento de dados é reiniciada ciclicamente numa excepção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

## Associações genéricas JMS:

A Associação JMS genérico fornece conectividade a fornecedores compatíveis com JMS 1.1 de terceiros. A operação das associações JMS Genérico é semelhante à de associações JMS.

O serviço fornecido através de uma associação JMS permite que um módulo da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) faça chamadas ou receba mensagens de sistemas externos. O sistema pode ser um sistema JMS externo.

A Associação JMS genérico fornece integração com fornecedores JMS compatíveis não JCA 1.5 que suportem JMS 1.1 e implementem o JMS Application Server Facility opcional. A associação JMS Generic suporta esses fornecedores JMS (incluindo Oracle AQ, TIBCO, SonicMQ, WebMethods, e BEA WebLogic) que não suportam JCA 1.5 mas suportam Application Server Facility da especificação JMS 1.1. O fornecedor JMS incorporado no WebSphere (SIBJMS), que é um fornecedor JMS JCA 1.5, não é suportado por esta associação; ao utilizar esse fornecedor, utilize "Associações JMS" na página 116.

Utilize esta Associação genérica ao integrar com um sistema baseado em JMS compatível com não JCA 1.5 dentro de um ambiente SCA. As aplicações de destino externas podem então receber mensagens e enviar mensagens para integrar com um componente SCA.

*Descrição geral das associações JMS genérico:*

As associações JMS genérico são associações JMS não JCA que fornecem conectividade entre o ambiente de Arquitectura do Componente de Serviço (SCA, Service Component Architecture) e sistemas de JM compatíveis com JMS 1.1 e que implementem o JMS Application Server Facility opcional.

## Associações genéricas JMS

Os principais aspectos das associações de importação e exportação JMS genérico incluem o seguinte:

- Porta de recepção: permite que fornecedores JMS baseados em não JCA recebam mensagens e as enviem para um bean orientado por mensagens (MDB - Message Driven Bean)
- Ligações: encapsulam uma ligação virtual entre uma aplicação cliente e uma aplicação fornecedora
- Destinos: utilizados por um cliente para especificar o destino de mensagens que produz ou a origem de mensagens que consome
- Dados de autenticação: utilizados para proteger o acesso à associação

## Associações de importação JMS genérico

As associações de importação JMS genérico permitem que componentes dentro do módulo SCA comuniquem com serviços fornecidos por fornecedores JMS compatíveis com não JCA 1.5.

A parte de ligação de uma importação JMS é um fornecedor de ligação. Um fornecedor de ligação, o objecto que um cliente utiliza para criar uma ligação a um fornecedor, encapsula um conjunto de parâmetros de configuração da ligação definido por um administrador. Cada fornecedor de ligação é uma instância da interface ConnectionFactory, QueueConnectionFactory ou TopicConnectionFactory.

A interacção com sistemas JMS externos inclui a utilização de destinos para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação JMS Genérico, dependendo do tipo de operação chamada:

- Unidireccional: A importação JMS genérico coloca uma mensagem no destino de envio configurado na associação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- Bidireccional (pedido-resposta): a importação JMS genérico coloca uma mensagem no destino de envio e depois persiste a resposta que recebe do componente SCA.

O destino receive é definido na propriedade do cabeçalho replyTo da mensagem de saída. Um bean orientado por mensagens (MDB) é implementado para escutar no destino de recepção, e quando uma mensagem é recebida, o MDB transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação da mensagem de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação da mensagem de pedido.

É possível especificar propriedades de cabeçalho dinâmicas e estáticas para ambos os cenários de utilização. As propriedades estáticas podem ser definidas a partir da associação de método de importação JMS genérico. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS genérico é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS genérico de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 35 ilustra a forma como a importação está ligada ao serviço externo.

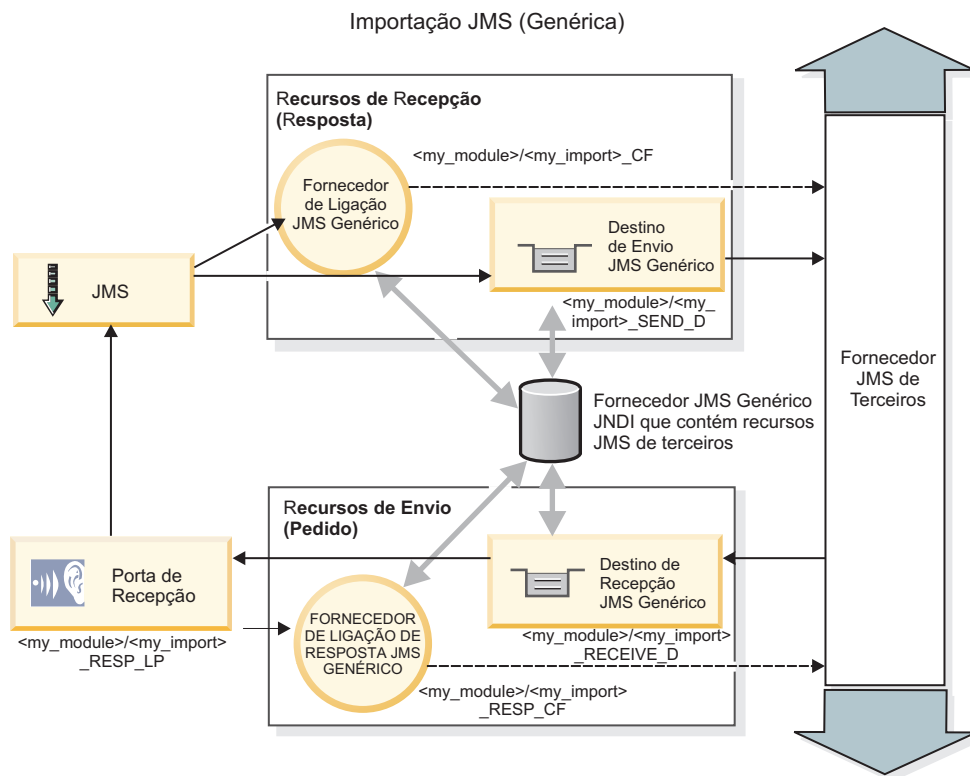


Figura 35. Recursos de associação de importação JMS genérico

### Associações de exportação JMS genérico

As associações de exportação JMS genérico fornecem os meios para módulos SCA fornecerem serviços a aplicações JMS externas.

A parte de ligação de uma exportação JMS é composta por um fornecedor de ligação e uma porta de recepção.

Uma exportação JMS Genérico tem destinos de envio e de recepção.

- O destino receive é onde a mensagem recebida do componente de destino deve ser colocada.
- O destino send é onde a resposta será enviada, excepto se a mensagem recebida o tiver substituído utilizando a propriedade de cabeçalho replyTo.

Um bean orientado por mensagens é implementado para escutar pedidos de entrada no destino receive especificado na associação de exportação.

- O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta.
- O destino especificado no campo replyTo da mensagem recebida substitui o destino especificado no campo send.
- Para cenários de pedido/resposta, a associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que a resposta copie o message ID (identificador da mensagem) do pedido para o campo correlation ID (identificador de correlação) da mensagem de resposta (predefinição) ou a mensagem pode copiar o correlation ID do pedido para o campo correlation ID da mensagem de resposta.

A Figura 36 ilustra como o solicitador externo está ligado à exportação.

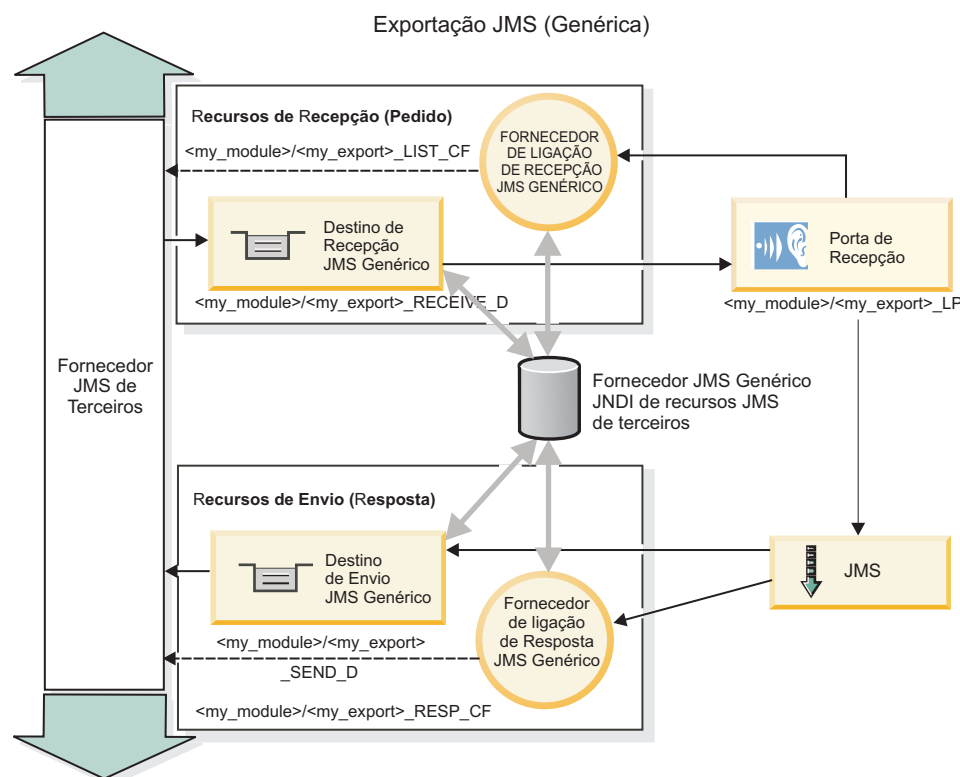


Figura 36. Recursos de associação de exportação JMS genérico

Funções-chave de associações JMS Genérico:

Os componentes da associação de importação e exportação JMS Genérico são consistentes com os do JMS incorporado e associações de importação MQ JMS do WebSphere. As funções-chave incluem definições de cabeçalho e acesso a recursos Java EE existentes. No entanto, dada a sua natureza genérica, não existem opções de conectividade específicas para o fornecedor JMS e esta associação tem uma capacidade limitada de gerar recursos no momento da implementação e instalação.

### Generic imports (Importações genéricas)



Tal como a aplicação de importação MQ JMS, a implementação do JMS Genérico é assíncrona e suporta três invocações: unidireccional, bidireccional (também chamada pedido-resposta) e chamada de retorno.

Quando a importação JMS é implementada, é implementado um bean orientado por mensagens (MDB) fornecido pelo ambiente do tempo de execução. O MDB procura obter respostas à mensagem de pedido. O MDB está associado ao (procura obter respostas no) destino enviado com o pedido no campo de cabeçalho `replyTo` da mensagem JMS.

### **Generic exports (Exportações genéricas)**

As associações de exportação JMS genéricas diferem das associações de exportação EIS no modo de processamento da devolução do resultado. Uma exportação JMS Genérico envia explicitamente a resposta ao destino `replyTo` especificado na mensagem de entrada. Se não estiver nenhum destino especificado, é utilizado o destino de envio.

Quando é implementada a exportação de JMS Genérico, é implementado um bean orientado por mensagens (um MDB diferente do utilizado para as importações de JMS Genérico. Escuta os pedidos de chegada no destino de recepção e envia-os para o tempo de execução SCA para serem processados.

### **Cabeçalhos especiais**

As propriedades dos cabeçalhos especiais são utilizadas nas importações e exportações de JMS Genérico para informar a associação-destino sobre como processar a mensagem.

Por exemplo, a propriedade `TargetFunctionName` é utilizada pelo selector de função predefinido para identificar o nome da operação na interface de exportação que está a ser chamada.

**Nota:** A associação de importação pode ser configurada para definir o cabeçalho `TargetFunctionName` para o nome de operação de cada operação.

### **Recursos Java EE**

São criados vários recursos Java EE quando é implementada uma associação JMS num ambiente Java EE.

- Porta de recepção para serviço de escuta no destino de recepção (resposta) – apenas bidireccional – para importações e no destino de recepção (pedido) para exportações
- Fornecedor de ligação a JMS Genérico para `outboundConnection` (importação) e `inboundConnection` (exportação)
- Destino de JMS Genérico para os destinos de envio (importação) e recepção (exportação; apenas bidireccional)
- Fornecedor de ligação ao JMS Genérico para `responseConnection` (apenas bidireccional e opcional; em alternativa, é utilizada a `outboundConnection` para importações e a `inboundConnection` para exportações)
- Destino de JMS Genérico para os destinos de envio (importação) e recepção (exportação) – apenas bidireccional.
- Destino JMS da chamada de retorno do fornecedor predefinido do serviço de mensagens utilizado para aceder ao destino da fila da chamada de retorno SIB (apenas bidireccional)
- Fornecedor de ligação JMS da chamada de retorno do fornecedor predefinido do serviço de mensagens utilizado para aceder ao destino (apenas bidireccional) JMS da chamada de retorno
- Destino da fila de chamadas de retorno SIB utilizado para armazenar informações sobre a mensagem de pedido a utilizar durante o processamento da resposta (apenas bidireccional)

A tarefa de instalação cria a `ConnectionFactory`, os três destinos e a `ActivationSpec` a partir das informações nos ficheiros de importação e exportação.

### *Cabeçalhos JMS genéricos:*

Cabeçalhos JMS genéricos são Objectos de dados de serviço (SDO, Service Data Objects) que contêm todas as propriedades das propriedades de mensagem JMS genéricas. Estas propriedades podem ser da mensagem de entrada ou podem ser as propriedades que serão aplicadas à mensagem de saída.

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

As seguintes propriedades são definidas estaticamente no methodBinding:

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

A associação JMS genérica também suporta a modificação dinâmica de cabeçalhos e propriedades JMS na mesma forma que as associações JMS e MQ JMS.

Alguns fornecedores de JMS Genérico colocam restrições sobre quais as propriedades que podem ser definidas pela aplicação e em que combinações. Tem de consultar a documentação de produto de terceiros para mais informações. No entanto, uma propriedade adicional pode ser acrescentada ao methodBinding, ignoreInvalidOutboundJMSProperties, que permite propagar quaisquer excepções.

Os cabeçalhos e propriedades de mensagem JMS genéricos são utilizados apenas quando está ligado o comutador de associação SCDL de arquitectura de componente de serviço base. Quando o comutador está ligado, são propagadas informações de contexto. Por predefinição, este comutador está ligado. Para prevenir a propagação de informação de contexto, altere o valor para **false**.

Quando a propagação de contexto é activada, é permitido que a informação de cabeçalho flua para a mensagem ou para o componente destino. Para ligar e desligar a propagação de contexto, especifique **true** ou **false** para o atributo contextPropagationEnabled das associações de importação e exportação. Por exemplo:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextPropagationEnabled="true">
```

A predefinição é **true**.

### *Resolução de problemas Generic Associações JMS:*

Poderá diagnosticar e corrigir problemas através de Generic associações JMS.

#### **Excepções de implementação**

Em resposta a várias condições de erro, a implementação de importações e exportações JMS Genéricas poderá devolver um ou dois tipos de excepções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter a excepção (JMSException) original.

#### **Resolução de problemas Genéricos Expiração de mensagens JMS**

Uma mensagem de pedido efectuada pelo fornecedor JMS está sujeita a um limite de validade.

*Expiração de pedido* refere-se ao limite da validade de uma mensagem de pedido efectuada pelo fornecedor JMS quando a JMSExpiration quando é alcançada a hora da mensagem de pedido. Conforme acontece com outras associações JMS, a associação JMS Genérica processa a expiração do pedido através da definição do limite de validade, na mensagem da chamada de retorno colocada pela importação, de modo a que seja a mesma do pedido de saída. A notificação do limite de validade da mensagem da chamada de retorno irá indicar que a mensagem de pedido atingiu o limite de validade, e o cliente deverá ser notificado por meios de uma excepção de negócio.

Contudo, caso o destino da chamada de retorno seja movido para o fornecedor da terceira parte, este tipo de expiração de pedido não é suportada.

*Expiração de pedido* refere-se ao limite da validade de uma mensagem de pedido efectuada pelo fornecedor JMS quando a hora do JMSExpiration da mensagem de pedido é alcançada.

A expiração da resposta para a associação JMS genérica não é suportada, devido ao facto de o comportamento de expiração exacto de um terceiro fornecedor JMS não estar definido. Pode, no entanto, verificar se a resposta excedeu o limite da validade caso, ou quando, a receba.

Para mensagens de pedidos externos, o valor Limite de validade JMS será calculado a partir da hora esperada e a partir dos valores Limite da validade do pedido levados a cabo no asyncHeader, IF SET.

### **Resolução de problemas Genéricosno fornecedor de ligação JMS.**

Quando definir determinados tipos de fornecedores de ligação no seu fornecedor JMS Genérico, pode receber uma mensagem de erro quando tentar iniciar a aplicação. É possível modificar o fornecedor de ligação externo para evitar este problema.

Quando iniciar uma aplicação, poderá receber as seguintes mensagens de erro:

Porta de Recepção MDB JMSConnectionFactory não corresponde  
Tipo JMSDestination

Este problema pode surgir quando definir fornecedores de ligação externos. A excepção pode ser fornecida, especificamente, quando criar um Fornecedor de Ligação Tópico JMS 1.0.2, em vez de um Fornecedor de Ligação JMS 1.1 (unificado), (isto é, um que seja capaz de suportar tanto a comunicação ponto a ponto, como a comunicação publicação/subscrição).

Para resolver esta questão, executar os seguintes passos:

1. Aceda ao fornecedor JMS Genéricoque está a utilizar.
2. Substituir o Fornecedor de Ligação Tópico JMS 1.0.2 que definiu por um Fornecedor de Ligação JMS 1.1. (unificado).

Quando iniciar a aplicação com o Fornecedor de Ligação JMS 1.1. que acabou de definir, não deverá voltar a receber uma mensagem de erro.

### **As mensagens JMS baseadas em SCA não aparecem no gestor de eventos falhados.**

Se as mensagens SCA originadas através de interacção JMS genérica falharem, poderá esperar encontrá-las no gestor de eventos falhados. Caso essas mensagens não constem do gestor de eventos falhados, certifique-se de que o valor da propriedade de tentativa máxima na porta de recepção sublinhada é igual ou superior a 1. Definir este valor para 1, ou para um número superior, permite a interacção com o gestor de eventos que falharam, durante chamadas para execução de SCA para as associações JMS genéricas.

*Excepções de processamento:*

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal excepção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer excepção da rotina de tratamento de dados é reiniciada ciclicamente numa excepção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer excepção de tempo de execução, incluindo uma excepção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a excepção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma excepção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

### **Associações JMS do WebSphere MQ:**

As associações JMS do WebSphere MQ fornecem integração com aplicações externas que utilizem um fornecedor baseado em JMS do WebSphere MQ.

Utilize as associações de exportação e importação JMS do WebSphere MQ para integrar directamente com sistemas JMS ou MQ JMS externos do ambiente de servidor. Isto elimina a necessidade de utilizar as funcionalidades MQ Link ou Client Link do Barramento de integração de serviço.

Quando um componente interage com um serviço baseado em JMS do WebSphere MQ através de uma importação, a associação de importação JMS do WebSphere MQ utiliza um destino para o qual os dados serão enviados e um destino onde a resposta possa ser recebida. A conversão dos dados para e de uma mensagem JMS é efectuada através da rotina de tratamento de dados JMS ou do componente de associação de dados.

Quando um módulo SCA fornece um serviço a clientes JMS do WebSphere MQ, a associação de exportação JMS do WebSphere MQ utiliza o destino onde o pedido pode ser recebido e a resposta pode ser enviada. A conversão dos dados para e de uma mensagem JMS é efectuada através da rotina de tratamento de dados JMS ou associação de dados.

O selector de função fornece um mapeamento para a operação no componente de destino a ser chamado.

*Descrição geral das associações JMS do WebSphere MQ:*

A associação JMS do WebSphere MQ fornece integração com aplicações externas que utilizem o fornecedor JMS do WebSphere MQ.

### **Tarefas administrativas do WebSphere MQ**

O administrador de sistema do WebSphere MQ deve criar o gestor de filas (Queue Manager) subjacente do WebSphere MQ, o qual as associações JMS do WebSphere MQ utilizarão antes de executar uma aplicação que contenha essas associações.

### **Associações de importação JMS do WebSphere MQ**

A importação JMS do WebSphere MQ permite que componentes dentro do módulo SCA comuniquem com serviços fornecidos por fornecedores baseados em JMS do WebSphere MQ. É necessário estar a utilizar uma versão suportada do WebSphere MQ. Os requisitos de hardware e software detalhados podem ser encontrados nas páginas de suporte da IBM.

São suportados dois tipos de cenários de utilização para associações de importação JMS do WebSphere MQ, dependendo do tipo de operação chamada:

- **Unidireccional:** a importação JMS do WebSphere MQ coloca uma mensagem no destino de envio configurado na associação de importação. Nada é enviado para o campo `replyTo` do cabeçalho JMS.
- **Bidireccional (pedido-resposta):** a importação JMS do WebSphere MQ coloca uma mensagem no destino de envio.

O destino recebe é definido no campo do cabeçalho `replyTo`. Um bean orientado por mensagens (MDB) é implementado para escutar no destino de recepção, e quando uma mensagem é recebida, o MDB transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação da mensagem de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação da mensagem de pedido.

Para ambos os cenários de utilização unidireccional e bidireccional, as propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser definidas a partir da associação do método de importação JMS. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS do WebSphere MQ é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS do WebSphere MQ de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 37 na página 133 ilustra a forma como a importação está ligada ao serviço externo.

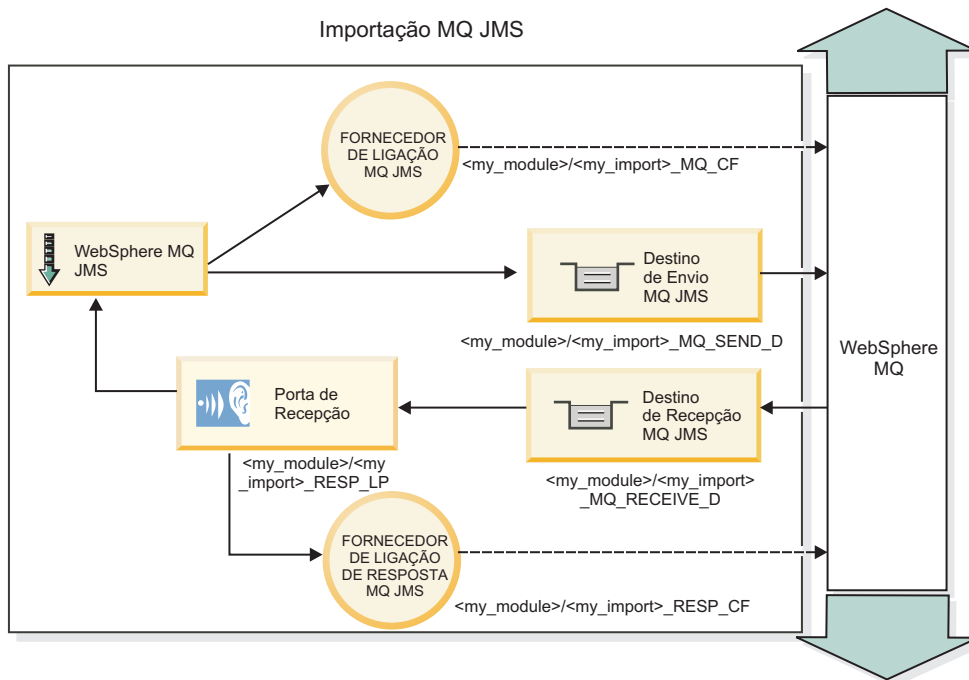


Figura 37. Recursos de associação de importação JMS do WebSphere MQ

### Associações de exportação JMS do WebSphere MQ

A associação de exportação JMS do WebSphere MQ fornece os meios para os módulos SCA fornecerem serviços para aplicações JMS externas no fornecedor JMS baseado no WebSphere MQ.

Um bean orientado por mensagens é implementado para escutar pedidos de entrada no destino recebe especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. O destino especificado no campo replyTo da mensagem de resposta substitui o destino especificado no campo send.

A Figura 38 na página 134 ilustra como o solicitador externo está ligado à exportação.

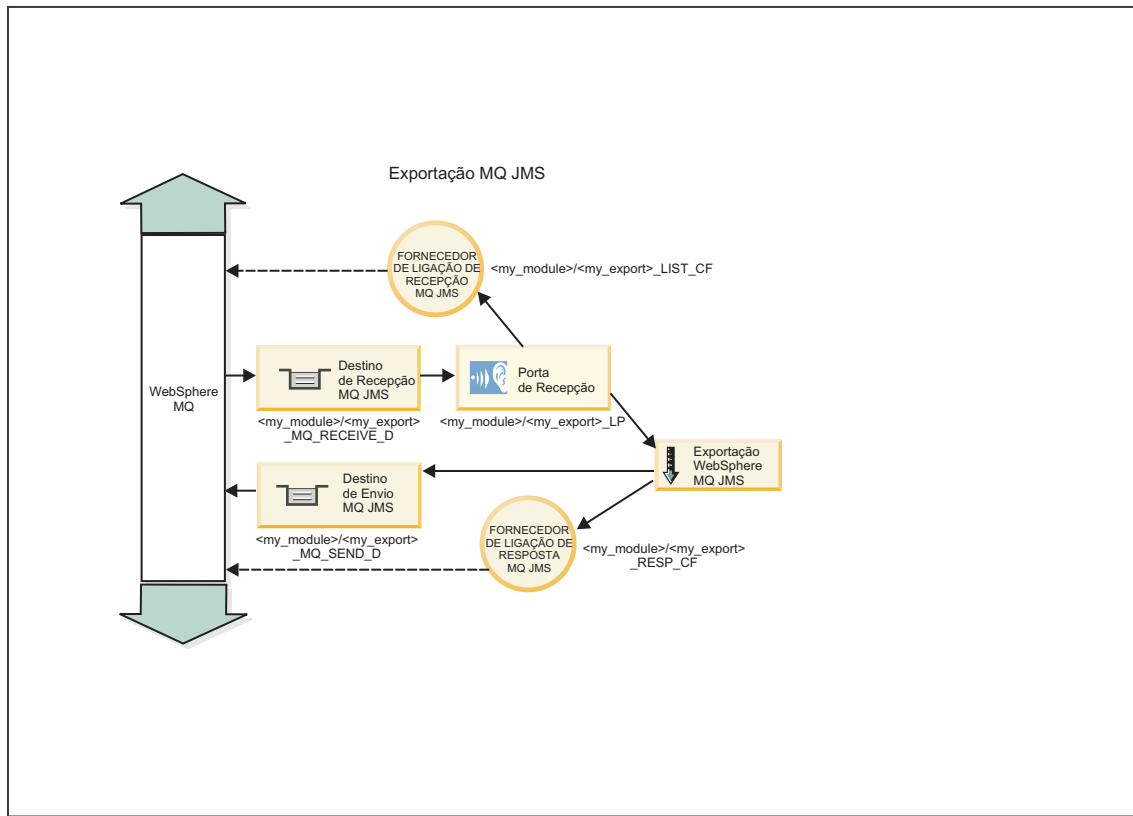


Figura 38. Recursos de associação de exportação JMS do WebSphere MQ

**Nota:** Figura 37 na página 133 e Figura 38 ilustram a forma como uma aplicação de uma versão anterior de IBM Business Process Manager está ligado a um serviço externo. Para aplicações desenvolvidas para a versão 7.0 de IBM Business Process Manager, a especificação de activação é utilizada em vez da porta de recepção e do fornecedor de ligação.

*Funções-chave de associações ao WebSphere MQ JMS:*

As funções-chave de associações ao WebSphere MQ JMS incluem cabeçalhos, artefactos Java EE e recursos criados Java EE.

## Cabeçalhos

Um cabeçalho com mensagem JMS contém uma série de campos predefinidos com valores utilizados tanto por clientes como fornecedores para identificar e encaminhar mensagens. É possível propriedades de associação para configurar estes cabeçalhos com valores fixos ou especificar dinamicamente os cabeçalhos no tempo de execução.

### JMSCorrelationID

Liga-se a uma mensagem relacionada. Normalmente, este campo fica definido para a cadeia do identificador de mensagem da mensagem à qual se está a responder.

### TargetFunctionName

Este cabeçalho é utilizado por um dos selectores de função fornecidos para identificar a operação que está a ser chamada. Definir adequadamente o cabeçalho TargetFunctionName JMS em mensagens enviadas para uma exportação JMS permite que este selector de função seja utilizado. A propriedade pode ser definida directamente nas aplicações de cliente JMS ou ao ligar uma importação a uma



associação JMS para a referida exportação. Neste caso, a associação de importação JMS deve ser configurada para definir o cabeçalho TargetFunctionName em cada operação da interface para o nome da operação.

### Esquemas de correlação

As associações WebSphere MQ JMS oferecem vários esquemas de correlação que são utilizados para determinar a forma de correlação entre mensagens de pedido e mensagens de resposta.

#### **RequestMsgIDToCorrelID**

O JMSMessageID é copiado para o campo JMSCorrelationID. Esta é a definição predefinida.

#### **RequestCorrelIDToCorrelID**

O JMSCorrelationID é copiado para o campo JMSCorrelationID.

### Recursos Java EE

São criados vários recursos Java EE quando é implementada uma importação MQ JMS num ambiente Java EE.

#### **Parameters (Parâmetros)**

##### **Fornecedor de ligação MQ**

Utilizado por clientes para criar uma ligação ao fornecedor MQ JMS.

##### **Fornecedor de ligação de respostas**

Utilizado pelo tempo de execução SCA MQ JMS quando o destino de envio está num Gestor de fila diferente do destino de recepção.

##### **Especificação de activação**

É associada uma especificação de activação MQ JMS a um ou mais beans orientados por mensagens, que proporciona a configuração necessária para que estes recebam mensagens.

#### **Destinos**

- Destino de envio:
  - Importações: para onde será enviado o pedido ou mensagem de saída.
  - Exportações: para onde será enviada a mensagem de resposta, caso o campo de cabeçalho JMSReplyTo da mensagem de entrada não se lhe sobreponha.
- Destino de recepção:
  - Importações: onde deverá ficar colocada a resposta ou mensagem de entrada.
  - Exportações: onde deverá ficar colocada a mensagem de entrada ou de pedido.

#### *Cabeçalhos JMS:*

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

#### **Cabeçalho de sistema JMS**

O cabeçalho de sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos encontrados tipicamente num cabeçalho JMS. Apesar de estes poderem ser modificados na mediação (ou ContextService), alguns campos de cabeçalho de sistema JMS definidos no SMO não serão propagados na mensagem JMS de saída, uma vez que são substituídos por valores de sistema ou estáticos.

Os campos de chave no cabeçalho de sistema JMS que podem ser actualizados numa mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidos específicos
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não-persistente, a predefinição é persistente)
- **JMSPriority** - valor de prioridade (0 a 9, a predefinição é JMS\_Default\_Priority)

### Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista Propriedades. As propriedades podem ser adicionadas, actualizadas ou eliminadas numa mediação ou através da utilização da API ContextService.

As propriedades também podem ser definidas estaticamente na associação JMS. As propriedades que estiverem definidas estaticamente substituem definições (com o mesmo nome) que estejam definidas dinamicamente.

As propriedades de utilizador propagadas de outras associações (por exemplo, uma associação HTTP) serão produzidas na associação JMS como propriedades JMS.

### Definições de propagação de cabeçalho

A propagação do cabeçalho de sistema JMS e propriedades quer da mensagem JMS interna para componentes na direcção do fluxo ou de componentes na direcção oposta ao fluxo para a mensagem JMS de saída podem ser controlados pelo sinalizador Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) na associação.

Quando o Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) está definido, a informação de cabeçalho pode fluir para a mensagem ou para o componente de destino, tal como descrito na lista seguinte:

- Pedido de exportação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.
- Resposta de exportação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de exportação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de exportação JMS.
- Pedido de importação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de importação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de importação JMS.
- Resposta de importação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.

*Cientes externos:*

O servidor pode enviar mensagens para ou receber mensagens de clientes externos utilizando associações JMS WebSphere MQ.

Um cliente externo (como um portal Web ou um sistema de informação empresarial) pode enviar uma mensagem para um componente SCA na aplicação através de uma exportação ou pode ser chamado por um componente SCA na aplicação através de uma importação.

A associação de exportação JMS WebSphere MQ fornece beans orientados por mensagens (MDBs) para escutar por pedidos de entrada no destino receive especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações através da associação de exportação.

As importações JMS WebSphere MQ são associadas a, e podem entregar mensagens, a clientes externos. Esta mensagem pode ou não exigir uma resposta do cliente externo.

Pode encontrar mais informações sobre como interagir com clientes externos utilizando o WebSphere MQ no Information Center do WebSphere MQ.

*Resolução de problemas WebSphere MQ Associações JMS:*

Poderá diagnosticar e corrigir problemas através de WebSphere MQ associações JMS.

### **Exceções de implementação**

Em resposta a várias condições de erro, a implementação de importações e exportações MQ JMS poderá devolver um ou dois tipos de exceções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter a excepção original (JMSException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

**As mensagens MQ WebSphere baseadas em SCA não estão a surgir no gestor de eventos falhados.**

Se as mensagens SCA que originadas através de uma WebSphere interacção JMS falharem, poderá encontrá-las no gestor de eventos falhados. Se essas mensagens não aparecerem no gestor de eventos falhados, certifique-se de que o valor da propriedade de tentativa máxima na porta de recepção sublinhada é igual, ou superior ao **1**. Definir esta valor para **1** ou mais permite a interacção com o gestor de eventos que falhou, durante chamadas para execução da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) para as associações MQ JMS.

### **Cenários de uso indevido: comparação com WebSphere associações MQ**

A WebSphere associação MQ JMS é concebida para cooperar com as aplicações JMS implementadas contra WebSphere MQ, o que expõe mensagens de acordo com o modelo de mensagens JMS. Contudo, as WebSphere importações e exportações MQ são concebidas, sobretudo, para cooperar com WebSphere aplicações MQ nativas, e para expor o conteúdo completo do WebSphere corpo de mensagens MQ TO MEDIATIONS.

Os cenários seguintes devem ser construídos utilizando a WebSphere associação MQ JMS, e não a associação WebSphere MQ:

- Chamar um bean orientado por mensagens (MDB) JMS de um módulo SCA, onde o bean orientado por mensagens é implementado contra o WebSphere fornecedor MQ JMS. Utilizar uma WebSphere importação MQ JMS.
- Permitir que o módulo da SCA seja chamado por um componente servlet Java EE ou EJB por meio de JMS. Utilizar uma WebSphere exportação MQ JMS.
- Mediar os conteúdos de um MapMessage JMS, em trânsito por WebSphere MQ. Utilizar WebSphere exportações e importações MQ JMS conjuntamente com a associação de dados apropriada tratamento ou dados.

Em algumas situações é provável que a associação WebSphere MQ e a WebSphere associação MQ JMS cooperem. Em específico, quando se abre caminho entre EE Java e EE não-JAVA WebSphere as aplicações MQ utilizam WebSphere exportações MQ e WebSphere importações MQ JMS (ou vice-versa) conjuntamente com associações de dados ou módulos de mediação apropriados (ou ambos).

#### *Exceções de processamento:*

A forma em que a associação é configurada determina a forma como as exceções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal exceção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal exceção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer exceção da rotina de tratamento de dados é reiniciada ciclicamente numa exceção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a exceção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma exceção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

#### **Associações do WebSphere MQ:**

A associação do WebSphere MQ fornece uma conectividade Service Component Architecture (SCA) com aplicações do WebSphere MQ.

Utilize as associações de exportação e importação do WebSphere MQ para serem integradas directamente um sistema baseado no WebSphere MQ a partir do ambiente de servidor. Isto elimina a necessidade de utilizar as funcionalidades MQ Link ou Client Link do Barramento de integração de serviço.

Quando um componente interage com um serviço do WebSphere MQ através de uma importação, a associação de importação do WebSphere MQ utiliza uma fila para a qual os dados são enviados e uma fila onde a resposta pode ser recebida.

Quando um módulo SCA fornece um serviço a clientes do WebSphere MQ, a associação de exportação do WebSphere MQ utiliza uma fila onde o pedido pode ser recebido e a resposta pode ser enviada. O selector de função fornece um mapeamento para a operação no componente de destino a ser chamado.

A conversão de dados de carga útil para e de uma mensagem MQ é efectuada através da rotina de tratamento de dados ou associação de dados do corpo de MQ. A conversão dos dados de cabeçalho para e de uma mensagem MQ é efectuada através da associação de dados de cabeçalho MQ.

Para obter mais informações sobre as versões do WebSphere MQ suportadas, consulte a página da web requisitos detalhados do sistema.

*Descrição geral das associações do WebSphere MQ:*

A associação do WebSphere MQ fornece integração com aplicações baseadas em MQ nativas.

### **Tarefas administrativas do WebSphere MQ**

O administrador de sistema do WebSphere MQ deve criar o gestor de filas subjacente do WebSphere MQ, que as associações do WebSphere MQ utilizarão antes de executarem um aplicação que contém essas associações.

### **Tarefas administrativas do WebSphere**

Tem de definir a propriedade do **Native library path (Caminho da biblioteca nativa)** do adaptador de recurso MQ no Websphere para a versão do WebSphere MQ suportada pelo servidor e reiniciar o servidor. Isto garante que as bibliotecas de uma versão suportada do WebSphere MQ estão a ser utilizadas. Os requisitos detalhados do hardware e do software podem ser encontrados nas páginas de suporte da IBM.

### **Associações de importação do WebSphere MQ**

A associação de importação do WebSphere MQ permite aos componentes dentro do módulo SCA para comunicar com serviços fornecidos por aplicações baseadas no WebSphere MQ. É necessário estar a utilizar uma versão suportada do WebSphere MQ. Os requisitos detalhados do hardware e do software podem ser encontrados nas páginas de suporte da IBM.

A interacção com sistemas do WebSphere MQ inclui a utilização de filas para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação do WebSphere MQ, dependendo do tipo de operação a ser chamada:

- Unidireccional: a importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Send destination queue (Fila de Destino de Envio)** da associação de importação. Nada é enviado para o campo `replyTo` do cabeçalho MQMD.
- Bidireccional (pedido-resposta): a importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Send destination queue (Fila de Destino de Envio)**

A fila receive é definida no campo do cabeçalho `replyTo` MQMD. Um bean orientado por mensagens (MDB) é implementado para escutar na fila de recepção, e quando uma mensagem é recebida, o bean orientado por mensagens transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)**) para esperar o identificador de correlação de mensagem de

resposta tenha sido copiado pelo identificador da mensagem de pedido (a predefinição) ou do identificador de correlação de mensagem de pedido.

É importante notar que o WebSphere MQ é uma associação assíncrona. Se um componente de chamada invocar uma importação de forma sincronizada do WebSphere MQ (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço WebSphere MQ.

A Figura 39 ilustra como a importação está ligada ao serviço externo.

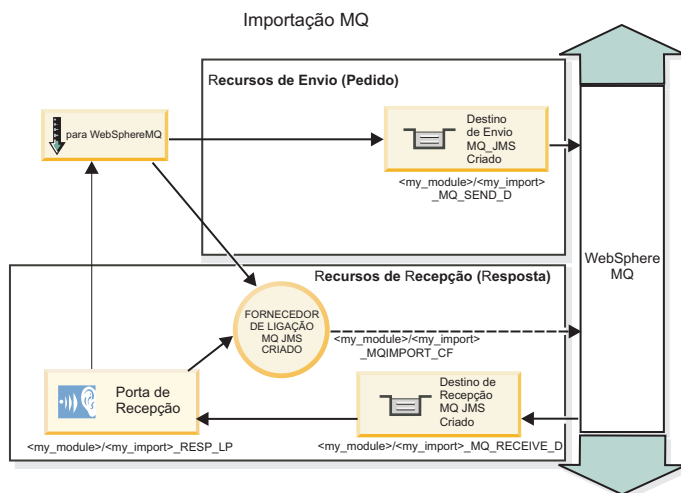


Figura 39. Recursos de associação de importação do WebSphere MQ

### Associações de exportação do WebSphere MQ

A associação de exportação do WebSphere MQ fornece os meios para os módulos SCA fornecerem serviços a aplicações baseadas no WebSphere MQ externas.

Um bean orientado por mensagens é implementado para escutar por pedidos de entrada no campo **Receive destination queue (Fila de Recepção de Destino)** especificado na associação de exportação. A fila especificada no campo **Send destination queue (Fila de Destino de Envio)** é utilizada para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. A fila especificada no campo `replyTo` da mensagem de resposta substitui a fila especificada no campo **Send destination queue (Fila de Destino de Envio)**.

A Figura 40 na página 141 ilustra como o solicitador externo está ligado à exportação.

## Exportação MQ

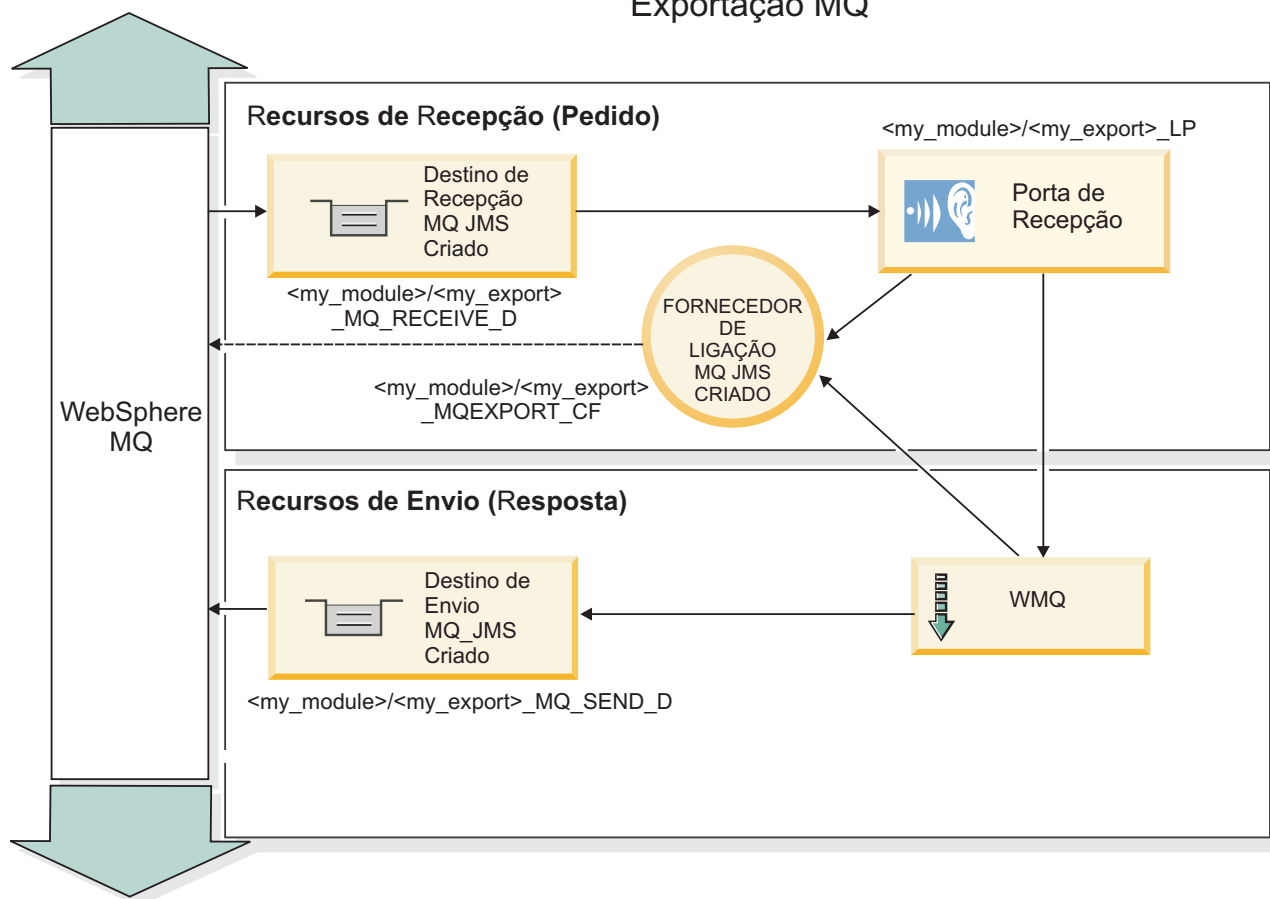


Figura 40. Recursos de associação de exportação do WebSphere MQ

**Nota:** A Figura 39 na página 140 e a Figura 40 ilustram a forma como uma aplicação de uma versão anterior do IBM Business Process Manager está ligada a um serviço externo. Para aplicações desenvolvidas para a versão 7.x, ou superior, do IBM Business Process Manager, a especificação de activação é utilizada em vez da porta de recepção e do fornecedor de ligação.

*Funções-chave de uma associação WebSphere MQ:*

As funções-chave de uma associação WebSphere MQ incluem cabeçalhos, artefactos Java EE e recursos criados Java EE.

### Esquemas de correlação

A aplicação de pedido/resposta WebSphere MQ pode utilizar uma de várias técnicas para correlacionar mensagens de resposta com pedidos, criadas com base nos campos MQMD MessageID e CorrelID. Na grande maioria dos casos, o solicitador permite que o gestor de filas selecione um MessageID e espera que a aplicação de resposta o copie para o CorrelID da resposta. Na maioria dos casos, o solicitador e aplicação de resposta sabem implicitamente qual a técnica de correlação que está a ser utilizada. Por vezes, a aplicação de resposta irá respeitar vários sinalizadores no campo Relatório do pedido que descrevem como processar estes campos.

As associações de exportação para mensagens WebSphere MQ podem ser configuradas com as seguintes opções:

### Response MsgId options (Opções de resposta MsgId):



**Novo MsgID**

Permite que o gestor de filas selecione um MsgId único para a resposta (por predefinição).

**Copiar a partir do MsgID do Pedido**

Copia o campo MsgId a partir do campo MsgId do pedido.

**Copiar a partir da mensagem SCA**

Define o MsgId para que seja incluído nos cabeçalhos WebSphere MQ na mensagem de resposta da SCA, ou permite que o gestor de filas defina um novo Id, caso o valor não exista.

**Como opções de relatório**

Inspecciona o campo Relatório do MQMD no pedido de sugestão sobre o procedimento a adoptar no processamento do MsgId. As opções MQRO\_NEW\_MSG\_ID e MQRO\_PASS\_MSG\_ID são suportadas e comportam-se como Novo MsgId e Copiar a partir do MsgID do pedido.

**Response CorrelId options (Opções de resposta CorrelId):****Copiar a partir do MsgID do Pedido**

Copia o campo CorrelId a partir do campo MsgId do pedido (por predefinição).

**Copiar a partir do CorrelID do Pedido**

Copia o campo CorrelId a partir do campo CorrelId do pedido.

**Copiar a partir da mensagem SCA**

Define o CorrelId para ser incluído nos cabeçalhos WebSphere MQ na mensagem de resposta da SCA, ou deixa-o em branco, caso o valor não exista.

**Como opções de relatório**

Inspecciona o campo Relatório do MQMD no pedido de sugestão sobre o procedimento a adoptar no processamento do CorrelId. As opções MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID e MQRO\_PASS\_CORREL\_ID são suportadas e comportam-se como Copiar a partir do MsgID do pedido e Copiar a partir do CorrelID do pedido.

As associações de importação para mensagens WebSphere MQ podem ser configuradas com as seguintes opções:

**Request MsgId options (Opções do MsgId do pedido):****Novo MsgID**

Permite que o gestor de filas selecione um MsgId único para o pedido (por predefinição).

**Copiar a partir da mensagem SCA**

Define o MsgId para ser incluído nos cabeçalhos WebSphere MQ na mensagem de pedido da SCA, ou permite que o gestor de filas defina um novo Id, caso o valor não exista.

**Response correlation options (Opções de correlação de resposta):****A resposta copiou o CorrelID a partir do MsgId**

Espera-se que a mensagem de resposta tenha um campo CorrelId definido, de acordo com o MsgId do pedido (por predefinição).

**A resposta copiou o MsgID a partir do MsgId**

Espera-se que a mensagem de resposta tenha um campo MsgId definido, de acordo com o MsgId do pedido.

**A resposta copiou o CorrelID a partir do CorrelId**

Espera-se que a mensagem de resposta tenha um campo CorrelId definido, de acordo com o CorrelId do pedido.

## Recursos Java EE

São criados vários recursos Java EE quando é implementada uma associação WebSphere MQ num ambiente Java EE.

### Parameters (Parâmetros)

#### Fornecedor de ligação MQ

Utilizado por clientes para criar uma ligação ao fornecedor WebSphere MQ.

#### Fornecedor de ligação de respostas

Utilizado pelo tempo de execução SCA MQ quando o destino de envio está num Gestor de fila diferente do destino de recepção.

#### Especificação de activação

É associada uma especificação de activação MQ JMS a um ou mais beans orientados por mensagens, que proporciona a configuração necessária para que estes recebam mensagens.

#### Destinos

- Destino de envio: é para onde é enviado o pedido ou mensagem de saída (importação); é para onde será enviada a mensagem de resposta (exportação), caso o campo de cabeçalho MQMD ReplyTo não se lhe sobreponha na mensagem de entrada.
- Destino de recepção: é onde deverá ficar colocada a resposta/pedido ou a mensagem de entrada.

#### *Cabeçalhos WebSphere MQ:*

Os cabeçalhos WebSphere MQ incorporam certas convenções para conversão de mensagens de Arquitectura do Componente de Serviço SCA).

As mensagens WebSphere MQ consistem num cabeçalho do sistema (o MQMD), zero ou outros mais cabeçalhos MQ (sistema ou personalizados) e num corpo de mensagem. Se existirem múltiplos cabeçalhos de mensagem na mensagem, a ordem dos cabeçalhos é significativa.

Cada cabeçalho contém informação que descreve a estrutura do seguinte cabeçalho. MQMD descreve o primeiro cabeçalho.

### Como são analisados os cabeçalhos MQ

É utilizada uma associação de dados do cabeçalho MQ para analisar os cabeçalhos MQ. Os seguintes cabeçalhos são automaticamente suportados:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Os cabeçalhos que começam com **MQH** são processados de modo diferente. Campos específicos do cabeçalho não são analisados; permanecem como bytes não analisados.

Para outros cabeçalhos MQ, pode inserir associações de dados de cabeçalho MQ personalizadas para analisar esses cabeçalhos.

### Como os cabeçalhos MQ são acedidos

Os cabeçalhos MQ podem ser acedidos no produto de uma de duas maneiras:

- Através do objecto de mensagem de serviço (SMO) numa mediação
- Através de ContextService API

Os cabeçalhos MQ são representados internamente com o elemento SMO MQHeader. MQHeader é um contentor de dados do cabeçalho que expande MQControl mas contém um elemento de valor de anyType. Contém MQMD, MQControl (informação de controlo do corpo da mensagem MQ) e uma lista de outros cabeçalhos MQ.

- MQMD representa os conteúdos da descrição da mensagem WebSphere MQ, excepto para informação que determine a estrutura e codificação do corpo.
- MQControl contém informação que determina a estrutura e codificação de um corpo de mensagem.
- MQHeaders contém uma lista de objectos de MQHeader.

A cadeia do cabeçalho MQ está desenrolada para que, dentro do SMO, cada cabeçalho MQ transporte a sua própria informação de controlo (CCSID, Codificação e Formato). Os cabeçalhos podem ser facilmente adicionados ou eliminados, sem alterar outros dados do cabeçalho.

### Campos de definição no MQMD

É possível actualizar o MQMD utilizando o Contexto API ou através do objecto de mensagem de serviço (SMO) numa mediação. Os seguintes campos são automaticamente propagados para a mensagem MQ de saída:

- Codificação
- CodedCharacterSet
- Formato
- Relatório
- Expiração
- Comentários
- Prioridade
- Persistência
- CorrelId
- MsgFlags

Configure a associação MQ numa Importação ou Exportação para propagar as seguintes propriedades para a mensagem MQ de saída:

#### MsgID

Defina **Request Message ID (Pedir ID de Mensagem)** para copy from SCA message (copiar de mensagem SCA).

#### MsgType

Limpe a caixa de verificação de **Set message type to MQMT\_DATAGRAM or MQMT\_REQUEST for request-response operation (Definir tipo de mensagem para MQMT\_DATAGRAM ou MQMT\_REQUEST para operação de pedido-resposta)**.

#### ReplyToQ

Limpe a caixa de verificação **Override reply to queue of request message (Substituir resposta para fila de mensagem de pedido)**.

#### ReplyToQMgr

Limpe a caixa de verificação **Override reply to queue of request message (Substituir resposta para fila de mensagem de pedido)**.

A partir da versão 7.0, os campos de contexto podem ser substituídos utilizando uma propriedade personalizada na definição de destino de JNDI. Defina a propriedade personalizada MDCTX com o valor SET\_IDENTITY\_CONTEXT no destino de envio para propagar os seguintes campo para a mensagem MQ de saída:

- UserIdentifier

- AppIdentityData

Defina a propriedade personalizada MDCTX com o valor SET\_ALL\_CONTEXT no destino de envio para propagar as seguintes propriedades para a mensagem MQ de saída:

- UserIdentifier
- AppIdentityData
- PutApplType
- PutApplName
- ApplOriginData

Alguns campos não são propagados para a mensagem MQ de saída. Os seguintes campos são substituídos durante o envio da mensagem:

- BackoutCount
- AccountingToken
- PutDate
- PutTime
- Deslocamento
- OriginalLength

*Acrescentar estatisticamente MQCIH numa associação WebSphere MQ:*

O IBM Business Process Manager suporta a adição estatística de informação de cabeçalho MQCIH sem utilizar um módulo de mediação.

Existem várias formas de adicionar informação de cabeçalho MQCIH a uma mensagem (por exemplo, ao utilizar a primitiva de mediação Definidor de Cabeçalho). Pode ser útil adicionar estatisticamente esta informação de cabeçalho, sem a utilização de um módulo de mediação adicional. A informação estatística de cabeçalho, incluindo o nome de programa, identificador de transacção e outros detalhes de cabeçalho do CICS podem ser definidos e adicionados como parte da associação WebSphere MQ.

O WebSphere MQ, o MQ CICS Bridge e o CICS devem ser configurados para informação de cabeçalho MQCIH, para serem adicionados estatisticamente.

Pode utilizar o Integration Designer para configurar a importação WebSphere MQ com os valores estáticos que são necessários para a informação de cabeçalho MQCIH.

Quando chega uma mensagem e é processada pela importação WebSphere MQ, é realizada uma verificação para ver se a informação de cabeçalho MQCIH já está presente na mensagem. Se a MQCIH estiver presente, os valores estáticos definidos na importação WebSphere MQ são utilizados para substituir os correspondentes valores dinâmicos na mensagem. Se o MQCIH não estiver presente, é criado um na mensagem e os valores estáticos definidos na importação WebSphere MQ são acrescentados.

Os valores estáticos definidos na importação WebSphere MQ são específicos a um método. Pode especificar diferentes valores estáticos MQCIH para diferentes métodos dentro da mesma importação WebSphere MQ.

Este serviço não é utilizado para fornecer valores padrão se o MQCIH não contiver informação de cabeçalho específica porque um valor estático definido na importação WebSphere MQ vai substituir um correspondente valor fornecido na mensagem recebida.

*Cientes externos:*

O IBM Business Process Manager pode enviar mensagens para ou receber mensagens de clientes externos utilizando associações WebSphere MQ.

Um cliente externo (por exemplo, um portal Web ou um sistema de informação empresarial) pode enviar uma mensagem para um componente SCA na aplicação através de uma exportação ou pode ser chamado por um componente SCA na aplicação através de uma importação.

A associação de exportação do WebSphere MQ fornece beans orientados por mensagens (MDBs) para escutar por pedidos de entrada no destino receive especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações através da associação de exportação.

As importações do WebSphere MQ são associadas a, e podem entregar mensagens, a clientes externos. Esta mensagem pode ou não exigir uma resposta do cliente externo.

Pode encontrar mais informações sobre como interagir com clientes externos utilizando o WebSphere MQ no Information Center do WebSphere MQ.

*Resolução de problemas das associações WebSphere MQ:*

Poderá diagnosticar e corrigir falhas e condições de erro que ocorram com as WebSphere associações MQ.

### **Principais condições de erro**

As principais condições de erro WebSphere das associações MQ são determinadas pela semântica transaccional, pela WebSphere configuração MQ, ou por referência a um comportamento existente noutros componentes. As principais condições de erro incluem:

- Erro ao ligar-se ao WebSphere gestor de filas MQ ou à fila MQ.  
Erro ao ligar-se ao WebSphere MQ para receber mensagens irá resultar num erro inicial, ao nível da porta de recepção do bean orientado por mensagens. Esta condição será registada no registo do WebSphere Application Server. Mensagens que persistam irão permanecer na WebSphere fila MQ até serem obtidas com sucesso (ou até expirarem por WebSphere MQ).  
Um erro ao ligar-se ao WebSphere MQ para enviar mensagens de saída, irá causar a remoção de alterações da transacção que controla o envio.
- Um erro ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.  
Um erro na associação de dados causa uma remoção de alterações do controlo de transacção do funcionamento.
- Um erro no envio da mensagem de saída.  
Um erro no envio de uma mensagem causa uma remoção de alterações da transacção relevante.
- Mensagens de resposta múltiplas ou inesperadas.  
A importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

### **Cenários de uso indevido: comparação com WebSphere associações MQ JMS**

As WebSphere importações e exportações MQ são concebidas, sobretudo, para cooperar com WebSphere aplicações MQ e expor o conteúdo completo do WebSphere corpo de mensagens MQ para as mediações.

A WebSphere associação MQ JMS é concebida para cooperar com as aplicações JMS implementadas contra WebSphere MQ, o que expõe mensagens de acordo com o modelo de mensagens JMS.

Os cenários seguintes devem ser construídos utilizando a WebSphere associação MQ JMS, e não a associação WebSphere MQ:

- Chamar um bean orientado por mensagens (MDB) JMS de um módulo SCA, onde o bean orientado por mensagens é implementado contra o WebSphere fornecedor MQ JMS. Utilizar uma WebSphere importação MQ JMS.
- Permitir que o módulo da SCA seja chamado por um componente servlet Java EE ou EJB por meio de JMS. Utilizar uma WebSphere exportação MQ JMS.
- Mediar os conteúdos de um MapMessage JMS, em trânsito por WebSphere MQ. Utiliza WebSphere exportações e importações MQ JMS conjuntamente com a associação de dados apropriada.

Em algumas situações é provável que a associação WebSphere MQ e a WebSphere associação MQ JMS cooperem. Em específico, quando se abre caminho entre EE Java e EE não-JAVA WebSphere as aplicações MQ utilizam WebSphere exportações MQ e WebSphere importações MQ JMS (ou vice-versa) conjuntamente com associações de dados ou módulos de mediação apropriados (ou ambos).

### **Mensagens não entregues**

Caso WebSphere o MQ não consiga entregar uma mensagem a um destino pretendido (devido a erros de configuração, por exemplo), em vez disso, este envia a mensagem para uma fila de mensagens retidas.

Ao fazê-lo, adiciona um cabeçalho de mensagens retidas para o início do corpo da mensagem. Este cabeçalho contém os motivos do erro, o destino original e outras informações.

### **As mensagens MQ baseadas em SCA não aparecem no gestor de eventos falhados.**

Se as mensagens SCA originadas através de uma WebSphere falharem, poderá encontrá-las no gestor de eventos falhados. Se essas mensagens não aparecerem no gestor de eventos falhados, certifique-se de que o destino WebSphere MQ que está por baixo apresenta um valor máximo de entregas falhadas superior a 1. Definir este valor para 2, ou superior, permite a interação com o gestor de eventos falhados durante as chamadas para execução da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) para as associações WebSphere MQ.

### **Os eventos MQ que falharam são reproduzidos para o gestor de filas errado.**

Quando um fornecedor de ligação predefinido está prestes a ser utilizado para ligações de envio, as propriedades de ligação devem corresponder às propriedades definidas, na especificação de activação utilizada para ligações de recepção.

O fornecedor de ligação predefinido é utilizado para criar uma ligação aquando da reprodução de um evento falhado e deve, por isso, ser configurado para o mesmo gestor de filas em que a mensagem foi inicialmente recebida.

#### *Excepções de processamento:*

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal excepção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer excepção da rotina de tratamento de dados é reiniciada ciclicamente numa excepção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer excepção de tempo de execução, incluindo uma excepção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a excepção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma excepção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

#### **Limitações de associações:**

As associações têm algumas limitações na sua utilização que estão listadas aqui.

*Limitações da associação MQ:*

A associação MQ tem algumas limitações na sua utilização, que são listadas aqui.

#### **Nenhuma distribuição publicação-subscrição**

O método publicação-subscrição de distribuir mensagens não é suportado actualmente pela associação MQ, apesar do próprio WMQ suportar publicação-subscrição. No entanto, a associação JMS MQ suporta este método de distribuição.

#### **Filas de recebimento partilhadas**

Várias associações de exportação e importação do WebSphere MQ esperam que quaisquer mensagens presentes nas suas filas de recebimento configuradas são para essa exportação ou importação. As associações de importação e exportação devem ser configuradas com as seguintes considerações:

- Cada importação MQ tem de ter uma fila de recebimento diferente, porque a associação de importação MQ assume que todas as mensagens na fila de recebimento são respostas a pedidos que enviou. Se a fila de recebimento for partilhada por mais de uma importação, as respostas podem ser recebidas pela importação incorrecta e não será correlacionada com a mensagem de pedido original.
- Cada exportação MQ deve ter uma fila de recebimento diferente, porque de outro modo não é possível prever qual a exportação que receberá qualquer mensagem de pedido em particular.
- As importações e exportações MQ podem indicar a mesma fila de envio.

*Limitações das associações JMS, MQ JMS e JMS genérico:*

As associações JMS e MQ JMS têm algumas limitações.



## Implicações de gerar associações predefinidas

As limitações de utilização das associações JMS, MQ JMS e JMS genérico são debatidas nas secções seguintes:

- Implicações de gerar associações predefinidas
- Esquema de correlação de resposta
- Suporte bidireccional

Quando gera uma associação, vários campos serão preenchidos como predefinições, se não escolher introduzir os valores. Por exemplo, um nome de fornecedor de ligação será criado. Se souber que irá colocar a aplicação num servidor e aceder-lhe remotamente com um cliente, deverá introduzir na criação da associação os nomes JNDI, em vez de aceitar a predefinição, uma vez que provavelmente quererá controlar esses valores através da consola administrativa no tempo de execução.

Porém, se tiver aceite as predefinições e descobrir mais tarde que não consegue aceder a partir de um cliente remoto, pode utilizar a consola administrativa para definir explicitamente o valor do fornecedor de ligação. Localize os pontos de terminação nas definições do fornecedor de ligação e adicione um valor como <server\_hostname>:7276 (se estiver a utilizar o número da porta predefinido).

## Esquema de correlação de resposta

Se utilizar o esquema de correlação de resposta CorrelationId To CorrelationId, utilizado para correlacionar mensagens numa operação de pedido-resposta, tem de ter um identificador de correlação dinâmico na mensagem.

Para criar um ID de correlação dinâmico num módulo de mediação ao utilizar o editor de fluxo de mediação, adicione uma primitiva de mediação de mapeamento antes da importação com a associação JMS. Abra o editor de mapeamento. Os cabeçalhos de arquitectura de componente de serviço estarão disponíveis na mensagem de destino. Arraste um campo que contenha um ID único na mensagem de origem para o ID de correlação no cabeçalho JMS na mensagem de destino.

## Suporte bidireccional

Apenas são suportados caracteres ASCII para nomes Java Naming and Directory Interface (JNDI) no tempo de execução.

## Filas de recebimento partilhadas

Várias associações de exportação e importação esperam que quaisquer mensagens presentes nas suas filas de recebimento configuradas são para essa exportação ou importação. As associações de importação e exportação devem ser configuradas com as seguintes considerações:

- Cada associação de importação tem de ter uma fila de recebimento diferente, porque a associação de importação assume que todas as mensagens na fila de recebimento são respostas a pedidos que enviou. Se a fila de recebimento for partilhada por mais de uma importação, as respostas podem ser recebidas pela importação incorrecta e não será correlacionada com a mensagem de pedido original.
- Cada exportação deve ter uma fila de recebimento diferente, porque de outro modo não é possível prever qual a exportação que receberá qualquer mensagem de pedido em particular.
- As importações e exportações podem indicar a mesma fila de envio.

## Objectos de negócio

A indústria do software de computadores desenvolveu vários modelos e contextos de programação em que os *objectos de negócio* fornecem uma representação natural dos dados de negócio para a Process Application.

Em geral, estes objectos de negócio:

- São definidos utilizando padrões da indústria
- Mapeiam de forma transparente dados para as tabelas de bases de dados ou sistemas de informações empresariais
- Suportam protocolos de chamada para execução remota
- Fornecem a fundação do modelo de programação de dados para a programação de aplicações

Process Designer e Integration Designer fornece aos programadores um modelo de objecto de negócio comum para representar diferentes tipos de entidades de negócio de diferentes domínios. Em termos de desenvolvimento, este modelo permite que os programadores definam objectos de negócio como definições de esquema de XML.

Em tempo de execução, os dados de negócio definidos pelas definições de esquema de XML são representados como Objectos de Negócio Java. Neste modelo, os objectos de negócio baseiam-se em rascunhos antecipados da especificação SDO (Service Data Object) e fornecem o conjunto completo de interfaces de aplicação de modelo de programação requeridas para manipular os dados de negócio.

### **Definir objectos de negócio**

É possível definir objectos de negócio utilizando o editor de objectos de negócio no Integration Designer. O editor de objectos de negócio armazena os objectos de negócio como definições de esquema de XML.

Utilizar o esquema de XML para definir objectos de negócio tem várias vantagens:

- O esquema de XML fornece um modelo de definição de dados baseado em padrões e uma fundação para a interoperabilidade entre sistemas e aplicações heterogéneas diferentes. O esquema de XML é utilizado em conjunto com a Web Services Description Language (WSDL) para fornecer contratos de interface baseados em padrões entre componentes, aplicações e sistemas.
- O esquema de XML define um modelo de definição de dados ricos para representar dados de negócio. Este modelo inclui tipos complexos, tipos simples, tipos definidos pelo utilizador, herança de tipos e cardinalidade, entre outras características.
- Os objectos de negócio podem ser definidos por interfaces e dados empresariais definidos na Web Services Description Language, bem como pelo esquema de XML das organizações de padrões da indústria ou de outros sistemas e aplicações. O Integration Designer pode importar estes objectos de negócio directamente.

O Integration Designer também fornece suporte para detectar dados de negócio em bases de dados e sistemas de informações empresariais e para gerar em seguida a definição de objecto de negócio do esquema de XML baseado em padrões desses dados de negócio. Os objectos de negócio gerados desta forma são muitas vezes referidos como *objectos de negócio específicos de aplicação* porque imitam a estrutura dos dados de negócio definidos no sistema de informações empresariais.

Quando um processo está a manipular dados de muitos sistemas de informações diferentes, pode ser importante transformar a representação distinta de dados de negócio (por exemplo, CustomerEIS1 e CustomerEIS2 ou OrderEIS1 e OrderEIS2) numa única representação canónica (por exemplo, Customer ou Order). A representação canónica é muitas vezes referida como o *objecto de negócio genérico*.

As definições de objectos de negócio, em particular objectos de negócio genéricos, são frequentemente utilizadas por mais de uma aplicação. Para suportar esta reutilização, o Integration Designer permite que os objectos de negócio sejam criados em bibliotecas que possam em seguida ser associados a múltiplos módulos de aplicação.

A Linguagem de Descrição de Serviços da Web (WSDL, Web Services Description Language) define os contratos para os serviços fornecidos e consumidos por um módulo de aplicação Arquitectura do Componente de Serviço (SCA, Service Component Architecture), bem como os contratos utilizados para

criar os componentes no módulo de aplicação. Num contrato, uma WSDL pode representar tanto operações como objectos de negócio (que são definidos pelo esquema XML para representar os dados de negócio).

## Trabalhar com objectos de negócio

A SCA (Service Component Architecture) fornece o contexto para definir um módulo de aplicação, os serviços que fornece, os serviços que consome e a composição de componentes que fornecem a lógica empresarial do módulo de aplicação. Os objectos de negócio desempenham um papel importante na aplicação, definindo os dados de negócio que são utilizados para descrever os contratos de serviço e componente e os dados de negócio que os componentes manipulam.

O seguinte diagrama representa um módulo de aplicação de SCA e ilustra muitos dos locais onde o programador trabalha com objectos de negócio.

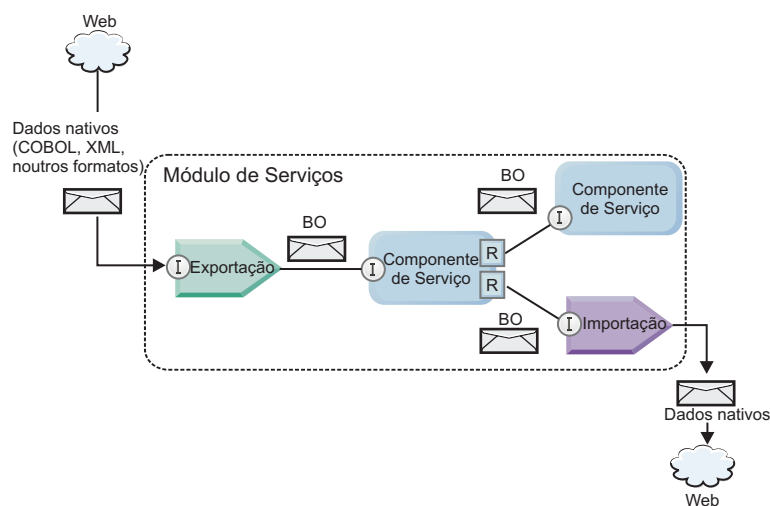


Figura 41. Os objectos de negócio representam os dados que fluem entre os serviços numa aplicação

**Nota:** Este tópico descreve como os objectos de negócio são utilizados pelos módulos de aplicação de SCA. Se estiver a utilizar interfaces de Java, os módulos de aplicação de SCA também podem processar objectos de Java.

## Modelo de programação de objecto de negócio

O modelo de programação de objecto de negócio consiste num conjunto de interfaces de Java que representam:

- A definição de objecto de negócio e os dados da instância
- Um conjunto de serviços que suportam as operações nos objectos de negócio

As definições de tipo de objecto de negócio são representadas pelas interfaces `commonj.sdo.Type` e `commonj.sdo.Property`. O modelo de programação de objecto de negócio fornece um conjunto de regras para o mapeamento de informações de tipo complexo do esquema XML para a interface `Type` e cada um dos elementos na definição de tipo complexo para a interface `Property`.

As instâncias de objectos de negócio são representadas pela interface `commonj.sdo.DataObject`. O modelo de programação de objecto de negócio não tem tipo, o que significa que a mesma interface `commonj.sdo.DataObject` pode ser utilizada para representar diferentes definições de objecto de negócio, como `Customer` e `Order`. A definição das propriedades que podem ser definidas e obtidas de cada objecto de negócio é determinada pelas informações de tipo definidas no esquema de XML associado a cada objecto de negócio.

O comportamento do modelo de programação de objecto de negócio é baseado na especificação Service Data Object 2.1. Para mais informações, consulte a especificação SDO 2.1 para Java, os guias de iniciação e os javadocs na web: <http://osoa.org/display/Main/Service+Data+Objects+Specifications>.

Os serviços de objectos de negócio suportam várias operações de ciclo de vida (tais como criação, igualdade, análise e serialização) em objectos de negócio.

Para especificidades no modelo de programação de objectos de negócio, consulte Programação utilizando serviços de objectos de negócio e a Documentação de API e SPI gerada sobre objectos de negócio.

## **Associações, associações de dados e rotinas de tratamento de dados**

Tal como mostrado na Figura 41 na página 151, os dados de negócio que são utilizados para chamar serviços fornecidos por módulos de aplicação de SCA são transformados em objectos de negócio para que os componentes de SCA possam manipular os dados de negócio. Do mesmo modo, os objectos de negócio manipulados por componentes de SCA são convertidos no formato de dados requerido pelos serviços externos.

Em alguns casos, tal como a associação do serviço da Web, a associação utilizada para exportar e importar serviços automaticamente transforma os dados no formato apropriado. Noutros casos, tal como a associação de JMS, os programadores podem fornecer uma associação de dados ou uma rotina de tratamento de dados que converte formatos não nativos em objectos de negócio representados pela interface `DataObject`.

Para mais informações sobre como desenvolver associações de dados e rotinas de tratamento de dados, consulte “Rotinas de tratamento de dados” na página 57 e “Associações de dados” na página 59.

## **Componentes**

Os componentes de SCA definem os respectivos contratos de serviço de provisão e consumo utilizando uma combinação da Linguagem de Descrição dos Serviços da Web e do esquema de XML. Os dados de negócio que o SCA transmite entre componentes são representados como objectos de negócio utilizando a interface `DataObject`. O SCA verifica que estes tipos de objecto de negócio são compatíveis com o contrato de interface definido pelo componente a ser chamado.

As abstrações do modelo de programação para manipular objectos de negócio variam de componente para componente. O componente POJO e a primitiva `Custom` do componente de fluxo de mediação fornecem uma manipulação directa dos objectos de negócio activando a programação de Java directamente utilizando as interfaces e serviços de programação de objectos de negócio. A maior parte dos componentes fornecem abstrações de nível superior para a manipulação de objectos de negócio, mas também fornecem fragmentos de código Java para definir um comportamento personalizado nas interfaces e serviços de objectos de negócio.

Os objectos de negócio podem ser transformados utilizando a combinação do componente Mediação de Fluxo da Interface e Mapa de Objecto de Negócio ou o componente de fluxo de mediação e a respectiva primitiva XML Map. Estas capacidades de transformação de objectos de negócio são úteis para converter objectos de negócio específicos da aplicação em e de objectos de negócio genéricos.

## **Objectos de negócio especiais**

Os objectos de mensagem de serviço e gráficos comerciais são dois tipos especializados de objectos de negócio que são utilizados para finalidades de aplicação específicas.

## Objecto de mensagem de serviço

Um objecto de mensagem de serviço (SMO) é um objecto de negócio especializado que é utilizado por componentes de fluxo de mediação para representar a colecção de dados associados a uma chamada para execução de serviço.

Um SMO tem uma estrutura de nível superior fixa que consiste em cabeçalhos, contexto, corpo e anexos (se estiverem presentes).

- Os cabeçalhos transportam informações relacionadas com a chamada para execução de serviço através de um determinado protocolo ou associação. Os exemplos são cabeçalhos de SOAP e cabeçalhos de JMS.
- Os dados de contexto transportam informações lógicas adicionais associadas à chamada para execução enquanto estão a ser processadas pelo componente de fluxo de mediação. Estas informações não fazem normalmente parte dos dados de aplicação enviados ou recebidos pelos clientes.
- O corpo do SMO transporta os dados de negócio da carga que representa a mensagem de aplicação principal ou dados de chamada para execução sob a forma de um objecto de negócio padrão.

O SMO também pode transportar dados de anexos para a chamada para execução de serviço da web utilizando SOAP com anexos.

Os fluxos de mediação executam essas tarefas como encaminhamento de pedidos e transformação de dados e o SMO fornece a vista combinada de conteúdo de cabeçalho e carga numa única estrutura unificada.

## Gráfico empresarial

Os gráficos comerciais são um objecto de negócio especial utilizado para fornecer suporte para a sincronização de dados nos cenários de integração.

Considere um exemplo em que dois sistemas de informação empresarial têm uma representação de uma encomenda específica. Quando a encomenda é alterada num sistema, uma mensagem pode ser enviada para o outro sistema para sincronizar os dados da encomenda. Os gráficos comerciais suportam a noção de envio apenas da parte da encomenda que mudou para o outro sistema e da sua anotação com informações de resumo de alterações para definir o tipo de alteração.

Neste exemplo, os gráficos comerciais da encomenda informarão o outro sistema que um dos itens de linha na encomenda foi eliminado e que a propriedade da data de expedição prevista da encomenda foi actualizada.

Os gráficos comerciais podem ser facilmente adicionados a objectos de negócio no Integration Designer. São muitas vezes encontrados em cenários em que os adaptadores WebSphere estão a ser utilizados e para suportar a migração das aplicações WebSphere InterChange Server.

## Modo de análise do objecto de negócio

O Integration Designer fornece uma propriedade de módulos e bibliotecas que pode utilizar para configurar o modo de análise XML para objectos de negócio para activo ou passivo.

- Se a opção for definida para *activo*, as sequências de bytes de XML são transmitidas de forma activa para criar o objecto de negócio.
- Se a opção for definida para *passivo*, o objecto de negócio é criado normalmente, mas a análise real da sequência de fluxo de XML é adiada e só parcialmente analisada quando as propriedades de objecto de negócio forem acedidas.

No modo de análise XML, os dados não XML são sempre analisado de forma activa para criar o objecto de negócio.

## Considerações ao escolher o modo de análise do objecto de negócio:

O modo de análise do objecto de negócio determina a forma como os dados XML são analisados no tempo de execução. Um modo de análise do objecto de negócio é definido num módulo ou biblioteca quando é criado. É possível alterar o modo de análise do módulo ou biblioteca, mas deve ter atenção às implicações.

O modo de análise de objectos de negócio é definido ao nível do módulo e da biblioteca. Os módulos que foram criados numa versão do IBM Integration Designer anterior à versão 7 são executados no modo de análise activo sem necessidade de quaisquer alterações. Por predefinição, os módulos e bibliotecas que são criados no IBM Integration Designer versão 7 e posteriores irão receber o modo de análise mais adequado consoante vários factores, tais como o modo de análise dos projectos existentes no respectivo espaço de trabalho ou o modo de análise dos projectos dependentes ou outros projectos na mesma solução, etc. É possível alterar o modo de análise do objecto de negócio de um módulo ou biblioteca para se adequar à sua implementação, mas tem de ter em consideração o seguinte.

### Considerações

- O modo de análise do objecto de negócio passivo processa os dados XML mais rapidamente. Contudo, existem diferenças de compatibilidade entre o modo activo e o passivo que é necessário ter em atenção antes de alterar a configuração de um módulo ou biblioteca. Estas diferenças irão afectar o comportamento de tempo de execução dos módulos. Para obter mais informações sobre que tipo de análise é otimizada para a aplicação, consulte "Benefits of using lazy versus eager parsing mode (Benefícios da utilização do modo de análise passivo versus activo)" nas ligações relacionadas.
- Um módulo só pode ser configurado para ser executado num modo de análise. As bibliotecas podem ser configuradas para suportar qualquer um dos modos de analisar ou ambos os modos de analisar. Uma biblioteca que esteja configurada para suportar ambos os modos de analisar pode ser referenciada por um módulo que utiliza o modo de análise activo e por um módulo que utiliza o modo de análise passivo. O modo de análise de uma biblioteca em tempo de execução é determinado pelos módulos que consultam a biblioteca. No tempo de execução, um módulo declara o seu modo de análise e este é utilizado pelo módulo e por quaisquer bibliotecas que esse módulo utilizar.
- Os módulos e as bibliotecas que estão configurados para modos de analisar diferentes são compatíveis nos seguintes casos:
  - Os módulos e bibliotecas configurados com o modo de análise passivo são compatíveis com bibliotecas que utilizam tanto o modo de análise passivo como ambos os modos de análise activo e passivo.
  - Os módulos e bibliotecas configurados com o modo de análise activo são compatíveis com as bibliotecas que utilizam tanto o modo de análise activo como ambos os modos de análise activo e passivo.
  - As bibliotecas configuradas com os modos de análise passivo e activo são compatíveis apenas com bibliotecas que utilizam ambos os modos de análise passivo e activo.
- Utilize o mesmo modo de análise para os módulos de interacção que comunicam utilizando a associação SCA. Se os módulos comunicarem utilizando modos de analisar diferentes, podem surgir problemas de desempenho.



### **Conceitos relacionados:**

“Benefícios da utilização do modo de análise passivo versus activo”

Algumas aplicações beneficiam de um modo de análise XML passivo, enquanto o desempenho de outros é melhorado através de um modo de análise activo. É recomendável que teste a sua aplicação em ambos os modos de análise para determinar qual o modo que melhor se adequa às características específicas da sua aplicação.

### **Benefícios da utilização do modo de análise passivo versus activo:**

Algumas aplicações beneficiam de um modo de análise XML passivo, enquanto o desempenho de outros é melhorado através de um modo de análise activo. É recomendável que teste a sua aplicação em ambos os modos de análise para determinar qual o modo que melhor se adequa às características específicas da sua aplicação.

As aplicações que analisam grandes sequências de dados XML provavelmente verão melhoramentos no desempenho quando é utilizado o modo de análise de XML passivo. As vantagens de desempenho aumentam na medida que aumenta a sequência de bites XML e decresce a quantidade de dados na sequência de bites que é acedida pela aplicação.

**Nota:** O modo de análise passivo de objectos de negócios é suportado no WebSphere Process Server versão 7.0.0.3 e posteriores. Também é suportado no IBM Process Server. Os módulos e os módulos de mediação que incluem componentes de fluxo de mediação não são suportados.

As aplicações seguintes têm mais probabilidade de terem um melhor desempenho utilizando um modo de análise activo:

- Aplicações que analisam sequências de dados não XML
- Aplicações que utilizam mensagens criadas através do serviço BOFactory
- Aplicações que analisam mensagens de XML muito pequenas

### **Referências relacionadas:**

“Considerações ao escolher o modo de análise do objecto de negócio” na página 154

O modo de análise do objecto de negócio determina a forma como os dados XML são analisados no tempo de execução. Um modo de análise do objecto de negócio é definido num módulo ou biblioteca quando é criado. É possível alterar o modo de análise do módulo ou biblioteca, mas deve ter atenção às implicações.

### **Considerações sobre desenvolvimento e migração de aplicações:**

Se estiver a configurar uma aplicação que tenha sido originalmente desenvolvida utilizando um modo de análise activo para utilizar agora um modo de análise passivo ou se estiver a planear alternar uma aplicação entre o modo de análise passivo e activo, tenha em atenção as diferenças entre os modos e as considerações quando estiver a alternar de modo.

### **Parâmetro identificador de erros**

Se a sequência de bytes de XML a ser analisada estiver incorrectamente formada, ocorrem excepções na análise.

- No modo de análise XML activo, estas excepções ocorrem assim que o objecto de negócio for analisado na sequência de XML interna.
- Se o modo de análise XML passivo for configurado, as excepções de análise ocorrem de forma dissimulada quando as propriedades de objecto de negócio forem acedidas e a parte do XML incorrectamente formado for analisada.

Para processar XML incorrectamente formado, seleccione uma das seguintes opções:

- Implementar um barramento de serviço de empresa nas extremidades para validar o XML interno



- Criar uma lógica de detecção de erros passiva no ponto onde as propriedades de objecto de negócio são acedidas

### **Pilhas de excepções e mensagens**

Visto que os modos de análise XML activo e passivo têm implementações subjacentes diferentes, os rastreios de pilha emitidos pelos serviços e interfaces de programação de objecto de negócio têm o mesmo nome de classe de excepção, mas poderão não conter a mesma mensagem de excepção ou conjunto translineado de classes de excepção específicas de implementação.

### **Formato de serialização de XML**

O modo de análise XML passivo fornece uma optimização do desempenho que tenta copiar XML não modificado da sequência de bytes interna para a sequência de bytes externa segundo a serialização. O resultado é um desempenho aumentado, mas o formato de serialização da sequência de bytes de XML externa poderá ser diferente se o objecto de negócio inteiro tiver sido actualizado no modo de análise XML passivo ou se tiver sido executado no modo de análise XML activo.

Embora o formato de serialização de XML possa não ser precisamente equivalente na forma sintáctica, o valor semântico fornecido pelo objecto de negócio é equivalente e independente dos modos de análise e o XML pode ser transmitido com segurança entre as aplicações em execução nos diferentes modos de análise com equivalência semântica.

### **Controlo de validação de instâncias de objecto de negócio**

O controlo de validação de ocorrências do modo de objectos de negócio de análise XML passivo fornece uma validação com a mais alta fidelidade dos objectos de negócio, em particular a validação dos valores de propriedade. Devido a estes melhoramentos, o controlo de validação de ocorrências de modo de análise passivo detecta problemas adicionais que não são detectados no modo de análise activo e fornece mensagens de erro mais detalhadas.

### **XML Maps, versão 602**

Os fluxos de mediação que se desenvolveram originalmente antes de WebSphere Integration Developer Versão 6.1 poderão conter as primitivas de mapeamento que utilizam um mapa ou folha de estilo que não podem ser executadas directamente no modo de análise XML passivo. Quando a aplicação migra para utilização no modo de análise XML passivo, os ficheiros de mapa associados com as primitivas de mapeamento podem ser actualizadas automaticamente pelo assistente de migração para ser executado no novo modo. No entanto, de uma primitiva de mapeamento se referir directamente a uma folha de estilos que foi editada manualmente, a folha de estilos não migra e não pode ser executada no modo de análise XML passivo.

### **APIs não publicadas privadas**

Se uma aplicação tirar partido de interfaces de programação de objecto de negócio não publicadas, privadas e específicas da implementação, é provável que a aplicação falhe a compilação quando o modo de análise é trocado. No modo de análise intenso, estas interfaces privadas são normalmente classes de implementação de objecto de negócio definidas pelo EMF (Eclipse Modeling Framework).

Em todos os casos, é recomendável que as APIs privadas sejam removidas da aplicação.

### **APIs Service Message Object EMF**

Um componente de mediação no IBM Integration Designer fornece a capacidade de manipular o conteúdo da mensagem utilizando as classes e interfaces de Java fornecidas no pacote `com.ibm.websphere.sibx.smobo`. No modo de análise XML passivo, as interfaces de Java no pacote

com.ibm.websphere.sibx.smobo podem ainda ser utilizadas, mas provavelmente os métodos que se referem directamente às classes e interfaces de EMF (Eclipse Modeling Framework) ou que são herdados das interfaces de EMF irão falhar.

O ServiceMessageObject e o respectivo conteúdo não podem ser convertidos em objectos EMF no modo de análise XML passivo.

### Serviço BOMode

O serviço BOMode é utilizado para determinar se o modo de análise XML actualmente em execução é activo ou passivo.

### Migração

Todas as aplicações anteriores à versão 7.0.0.0 são executadas no modo de análise XML activo. Quando são migradas em tempo de execução utilizando as ferramentas de migração em tempo de execução BPM, continuam a ser executadas no modo de análise XML activo.

Para activar uma aplicação anterior à versão 7.0.0.0 para ser configurada para utilizar o modo de análise XML passivo, primeiro tem de utilizar o Integration Designer para migrar os artefactos da aplicação. Após a migração, configura a aplicação para utilizar a análise de XML passiva.

Consulte Migrar artefactos de origem para obter informações sobre como migrar artefactos no Integration Designer e consulte Configurar o modo de análise de objecto de negócio de módulos e bibliotecas para obter informações sobre como definir o modo de análise.

## Relações

Um relação é uma associação entre duas ou mais entidades de dados, tipicamente objectos de negócios. Em IBM Business Process Manager Advanced, as relações podem ser utilizadas para transformar dados equivalentes em objectos de negócios e outros dados, mas que são representados de forma diferente ou podem ser utilizados para criar associações em objectos diferentes que se encontram em aplicações diferentes. Podem ser partilhados entre aplicações, entre soluções e até entre produtos.

O serviço de relações do IBM Business Process Manager Advanced fornece a infra-estrutura e operações para gerir relações. Uma vez que permite lidar com objectos de negócio independentemente da respectiva localização, pode fornecer uma vista holística unificada em todas as aplicações de uma empresa e funciona como bloco de construção para soluções BPM. Dado que as relações podem ser extensíveis e geridas, podem ser utilizadas em soluções de integração complexas.

### O que são relações?

Uma relação é uma associação entre objectos de negócio. Cada um dos objectos de negócio de uma relação é denominado um *participante* na relação. Cada um dos participantes da relação distingue-se de outros participantes com base na *função*, na relação em questão. Uma relação possui uma lista de funções.

A *definição* da relação descreve cada uma das funções e especifica a forma como as funções estão relacionadas. Descreve também a "forma" global da relação. Por exemplo, esta função apenas pode conter um participante, mas a outra função pode conter todos os participantes que forem necessários. Pode, por exemplo, ser definida uma relação *veículo-proprietário*, em que o proprietário pode possuir diversos veículos. Por exemplo, uma ocorrência pode conter os participantes seguintes para cada uma das funções:

- Veículo (Ferrari)
- Proprietário (Joaquim)

A definição da relação é um modelo para a *ocorrência* da relação. A ocorrência da relação é a instanciação em tempo de execução. No exemplo do *veículo-proprietário*, uma ocorrência poderá descrever qualquer das associações seguintes:

- Joaquim proprietário Ferrari
- Sara proprietária Mazda
- Bruno Proprietário Ferrari

A utilização de relações liberta-o da necessidade de personalizar a criação do controlo da relação persistente na lógica empresarial. Em determinados cenários, o serviço de relações faz um trabalho completo. Consulte o exemplo descrito na secção em Relações de identidade.

## Cenários

Este é um exemplo típico de uma situação em que uma solução de integração poderá utilizar relações. Uma grande empresa adquire diversas empresas ou unidades de negócio. Cada uma das unidades de negócio utiliza um software diferente para supervisionar o pessoal e os blocos de notas. A empresa necessita de uma forma para supervisionar os empregados e os respectivos blocos de notas. Pretende-se uma solução que permita:

- Ver todos os empregados das diversas unidades de negócio como se estivessem numa base de dados
- Obter uma vista única de todos os blocos de notas
- Permitir aos empregados ligarem-se ao sistema e comprarem um bloco de notas
- Acomodar os diferentes sistemas de aplicações empresariais das várias unidades de negócio

Para conseguir isso, a empresa necessita de uma forma para assegurar, por exemplo, que o John Smith e o John A. Smith em aplicações diferentes são vistos como o mesmo empregado. Por exemplo, é necessário obter uma forma de consolidar uma entidade única em múltiplos espaços de aplicação.

Os cenários mais complexos de relações envolvem criar processos BPEL que chamam relações em diferentes objectos encontrados em diversas aplicações. Nos cenários complexos de relações, os objectos de negócio residem na solução da integração, e não na aplicação. O serviço de relações fornece uma plataforma para gerir relações de forma persistente. Antes da utilização do serviço de relação, era necessário criar um objecto de serviço persistente. Dois exemplos de cenários de relações complexas, são:

- Existe um objecto de negócio de **car** com um número VIN na aplicação SAP e, pretende-se controlar o facto do veículo ser propriedade de terceiros. No entanto, a relação de propriedade está com alguém da aplicação PeopleSoft. Neste padrão de relações, existem duas soluções, pelo que é necessário criar uma ponte entre ambas.
- Uma grande empresa de retalho pretende ser capaz de supervisionar mercadoria devolvida para crédito ou devolução do dinheiro. Existem duas aplicações diferentes envolvidas: um sistema de gestão de compras (OMS, order management system) para compras e um sistema de gestão de devoluções (RMS, returns management system) para devoluções. Os objectos de negócio encontram-se em mais do que uma aplicação e é necessário mostrar as relações existentes entre elas.

## Padrões de utilização comum

Os padrões de relações mais comuns são padrões de *equivalência*. São baseados em referências cruzadas ou correlação. Estes são dois tipos de relações que se enquadram neste padrão: *não-identidade* e *identidade*.

- **relações de não-identidade** estabelecem associações entre objectos de negócio ou outros dados numa base um-para-muitos ou muitos-para-muitos. Para cada uma das ocorrências de relação, podem existir uma ou mais ocorrências de cada um dos participantes. Um tipo de relação não-identidade é uma relação de procura estática. Um exemplo disto é uma relação na qual **CA** numa aplicação SAP está relacionada com **California** numa aplicação Siebel.

•

**Relações de identidade** estabelecem associações entre objectos de negócio ou outros dados numa base um-para-um. Para cada ocorrência de relação, apenas poderá existir uma ocorrência de cada um dos participantes. As relações de identidade capturam referências cruzadas entre objectos de negócio equivalentes semanticamente, mas isso é identificado de forma diferente em aplicações diferentes. Cada um dos participantes da relação está associado a um objecto de negócio que possui um valor (ou combinação de valores) que identifica esse objecto de forma única. As relações de identidade transformam tipicamente os atributos chave de objectos de negócio, tais como números de ID e códigos do produto.

Por exemplo, se existir um objecto de negócio **car** nas aplicações SAP, PeopleSoft, e Siebel, e se pretender criar uma solução que faça a respectiva sincronização, normalmente necessitaria de introduzir uma lógica de sincronização de relação manual em seis mapas:

SAP -> genérico

genérico -> SAP

PeopleSoft-> genérico

genérico-> PeopleSoft

Siebel-> genérico

genérico-> Siebel

No entanto, se foram utilizadas relações na solução, o serviço de relações fornece implementações de padrões pré-construídos fazem a gestão de todas as ocorrências destas relações.

## Ferramentas para funcionar com relações

O *editor de relações* no Integration Designer é a ferramenta utilizada para modelar e conceber relações e funções de integração de negócio. Para obter informações de segundo plano e tarefas detalhadas sobre criar relações e utilizar o editor de relação, consulte Criar relações.

O *serviço de relações* é um serviço de infra-estrutura no IBM Business Process Manager que mantém relações e funções no sistema e, fornece operações para gerir relações e funções.

O *gestor de relações* é a interface administrativa para gerir relações. É acessado através das páginas do Gestor de Relações (Relationship Manager) na consola administrativa.

As relações podem ser chamadas programaticamente através das APIs do serviço de relações.

## Serviço de relações

O serviço de relações guarda dados de relações em tabelas de relações, onde mantém o controlo de valores específicos da aplicação entre aplicações e entre soluções. O serviço de relações fornece operações para gestão de relações e funções.

## Como funcionam as relações

As relações e as funções são definidas utilizando a interface gráfica da ferramenta de edição de relações no Integration Designer. O serviço de relações guarda os dados de correlação em tabelas na base de dados da relação na origem de dados predefinida especificada durante a configuração do serviço de relação. Uma tabela separada (por vezes denominada tabela participante) guarda informações para cada um dos participantes na relação. O serviço de relações utiliza estas tabelas de relações para controlar os valores específicos da aplicação relacionados e as informações actualizadas propagadas em todas as soluções.

As relações que são artefactos de negócio, são implementadas num projecto ou numa biblioteca partilhada. Durante a primeira implementação, o serviço de relações preenche os dados.

Em tempo de execução, quando os mapas ou outros componentes do IBM Business Process Manager necessitam de uma ocorrência de relação, as ocorrências da relação são actualizadas ou obtidas, dependendo do cenário.

Dados da relação e da ocorrência da função podem ser manipulados através de três formas:

- Chamada para execução de fragmentos do componente IBM Business Process Manager Java do serviço de relações das APIs
- Transformações da relação do servido de mapeamento de objectos de negócio do IBM Business Process Manager
- A ferramenta de gestão de relações

Para obter informações detalhadas sobre a tarefa de criação de relações, identificar tipos de relações e utilizar o editor de relações, consulte o tópico Criar relações.

## Gestor de relações

O gestor de relações é a interface administrativa para gestão de relações. É acedido através das páginas do Gestor de Relações (Relationship Manager) na consola administrativa.

O gestor de relações fornece uma interface gráfica do utilizador para criar e manipular dados de relações e de funções em tempo de execução. É possível gerir entidades de relações em todos os níveis: ocorrência da relação, ocorrência da função, dados do atributo e níveis de dados da propriedade. A utilização do gestor de relações, permite:

- Ver uma lista de relações no sistema e informações detalhadas para relações individuais
- Gerir ocorrências de relações:
  - Consultar dados de relações para ver subconjuntos de dados de ocorrências
  - Consultar dados de relações para ver subconjuntos de dados de ocorrências utilizando vistas de base de dados
  - Ver uma lista de ocorrências de relações que correspondem a uma consulta de relação e informações detalhadas sobre uma ocorrência
  - Editar os valores da propriedade para uma ocorrência de relação
  - Criar e eliminar ocorrências de relações
- Gerir funções e ocorrências de funções:
  - Ver detalhes sobre uma função ou uma ocorrência de função
  - Editar propriedades de ocorrência de uma função
  - Criar e eliminar ocorrências de função para uma relação
  - Remover alterações de dados de uma ocorrência de relação para um ponto no tempo em que tenha conhecimento que os dados são fidedignos
- Importar dados de uma relação estática existente para o sistema ou exportar dados de uma relação estática existente para um ficheiro RI ou CSV
- Remover esquema da relação e os dados do repositório quando a aplicação a utilizar a relação é desinstalada

## Relações em ambientes de Implementação de Rede

As relações podem ser utilizadas em ambientes de Implementação de Rede (ND, Network Deployment), sem qualquer configuração adicional.

Em ambientes de Implementação de Rede (ND, Network Deployment), as relações são instaladas num conjunto de aplicação. As relações são então visíveis no conjunto de unidades e todos os servidores do conjunto de unidades têm acesso aos dados da ocorrência guardados na base de dados da relação. A capacidade de executar o serviço de relação num ambiente ND torna-o escalável e altamente disponível.

O gestor de relações permite a gestão de relações ao nível de diferentes conjuntos de unidades através de uma interface administrativa centralizada. É possível ligar o gestor de relações a um servidor num conjunto de unidades seleccionando a respectiva relação MBean.

### **APIs do serviço de relações**

As relações podem ser chamadas programaticamente através das APIs do serviço de relações, dentro ou fora dos mapas de objectos de negócio.

Estão disponíveis três tipos de API:

- APIs para manipulação de ocorrências de relações (incluindo a criar, actualizar, eliminar directamente dados da ocorrência)
- APIs para suporte de padrões de relações (incluindo `correlate()`, `correlateforeignKeyLookup`)
- Padrões de procura de relações (APIs de procura)

## **O barramento de serviços empresariais no IBM Business Process Manager**

O IBM Business Process Manager suporta a integração de servidos de aplicação, incluindo as mesmas funções do WebSphere Enterprise Service Bus.

### **Ligar serviços através de um barramento de serviço de empresas**

Com um barramento de serviço de empresas (ESB, enterprise service bus), pode maximizar a flexibilidade de um SOA. Os participantes numa interacção de serviço são ligados ao ESB, em vez de um ao outro, directamente.

Quando o solicitador de serviços se liga ao ESB, este fica responsável por entregar os seus pedidos, por meio de mensagens, a um fornecedor de serviços que oferece a função e qualidade de serviço necessárias. O ESB facilita as interacções solicitador-fornecedor e endereça protocolos díspares, padrões de interacção ou funções de interacção. Um ESB pode também activar ou melhorar a monitorização e a gestão. O ESB fornece as funções de virtualização e gestão que implementam e prolongam as funções centrais da SOA.

O ESB resume as seguintes funções:

#### **Localização e identidade**

Os participantes não necessitam de saber a localização ou identidade dos outros participantes. Por exemplo, os solicitadores não necessitam de saber que um pedido pode ser reparado por qualquer um dos vários fornecedores; podem ser adicionados ou removidos fornecedores de serviços sem ruptura.

#### **Protocolo de interacção**

Os participantes não necessitam de partilhar o mesmo protocolo de comunicação ou estilo de interacção. Por exemplo, um pedido expresso como SOAP sobre HTTP pode ser prestado por um servidor que apenas compreenda SOAP sobre Serviço de Mensagens Java (JMS).

#### **Interface**

Os solicitadores e os fornecedores não necessitam de concordar numa interface comum. Um ESB reconcilia diferenças ao converter mensagens de pedido e resposta numa forma esperada pelo fornecedor.

#### **Qualidades do serviço (de interacção)**

Os participantes, ou administradores de sistema, declaram os seus requisitos de qualidade de serviço, incluindo a autorização de pedidos, encriptação e desencriptação de conteúdos de mensagens, auditoria automática de interacções de serviços e como os seus pedidos devem ser encaminhados (por exemplo, optimizar para velocidade ou custo).

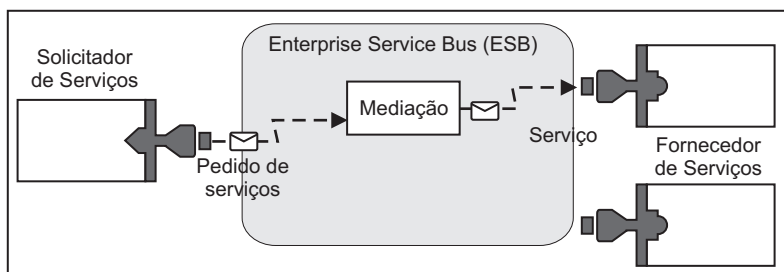
Interpor o ESB entre participantes permite-lhe modular a sua interacção através de uma construção lógica chamada *mediação*. As mediações operam em mensagens em curso entre solicitadores e fornecedores. Por exemplo, as mediações podem ser utilizadas para encontrar serviços com características específicas que



um solicitador está a pedir, ou para resolver as diferenças de interface entre solicitadores e fornecedores. As mediações podem ser encadeadas sequencialmente para interacções complexas.

Ao utilizar mediações, um barramento de serviço de empresa efectua as seguintes acções entre o solicitador e o serviço:

- *Encaminhar* mensagens entre serviços. Um barramento de serviço de empresas oferece uma infra-estrutura de comunicação comum que pode ser utilizada para ligar serviços e, assim, as funções de negócios que representam, sem a necessidade que os programadores escrevam e mantenham uma lógica de conectividade complexa.
- *Converter* os protocolos de transporte entre o solicitador e o serviço. Um barramento de serviço de empresas fornece uma forma consistente e baseada em padrões de integrar funções de negócios que utilizam padrões IT diferentes. Isto permite a integração de funções de negócios que normalmente não podiam comunicar, como ligar aplicações em silos departamentais ou permitir que aplicações em diferentes companhias participem em interacções de serviços.
- *Converter* formatos de mensagens entre solicitador e serviço. Um barramento de serviço de empresas permite que funções de negócios troquem informação em diferentes formatos, com o barramento a assegurar que a informação entregue a uma função de negócios está no formato requerido pela aplicação.
- *Processar* eventos de negócios de origens desiguais. Um barramento de serviço de empresas suporta interacções baseadas em eventos, para além das trocas de mensagens para processarem pedidos de serviço.



*Figura 42. Um barramento de serviço de empresas. O barramento de serviço de empresas está a encaminhar mensagens entre aplicações, que são solicitadores ou fornecedores de serviços. O barramento está a converter os protocolos de transporte e a converter os formatos de mensagens entre solicitadores e fornecedores. Nesta figura, cada aplicação utiliza um protocolo diferente (representado pelas formas geométricas dos seus conectores) e formatos de mensagens diferentes.*

Ao utilizar o barramento de serviço de empresa pode focar-se no seu negócio principal, em vez de se focar nos seus sistemas de computador. Pode alterar ou adicionar aos serviços, se necessário; por exemplo, para responder às alterações nos requisitos de negócios, para adicionar uma capacidade de serviço extra ou para adicionar nova capacidades. Pode fazer as alterações necessárias ao reconfigurar o barramento, com pouco ou nenhum impacto em serviços e aplicações existentes que utilizem o barramento.

### **Infra-estrutura de mensagens de barramento de serviços da empresa**

O IBM Business Process Manager inclui funções de barramento de serviços da empresa. O IBM Business Process Manager suporta a integração de tecnologias orientadas para serviços, orientadas para mensagens e orientadas para eventos, para fornecer uma infra-estrutura de mensagens baseada em standards num barramento de serviços da empresa integrado.

As funções do serviço empresarial que podem ser utilizadas nas aplicações da empresa, fornecem não só uma camada de transporte como também, suporte de mediação para facilitar as interacções do serviço. O barramento de serviços da empresa é criada a partir de padrões abertos e da Arquitectura Orientada para Serviços (SOA, service-oriented architecture). É baseado na infra-estrutura robusta Java EE e serviços de plataforma associados, fornecidos por IBM WebSphere Application Server Network Deployment.



IBM Business Process Manager is powered by the same technology available with IBM WebSphere Enterprise Service Bus. Esta função faz parte da funcionalidade subjacente do IBM Business Process Manager e não é necessária qualquer licença adicional do WebSphere Enterprise Service Bus para tirar partido destas funções.

No entanto, é possível implementar licenças autónomas adicionais do WebSphere Enterprise Service Bus na empresa para expandir o alcance da conectividade das soluções de integração do processo suportadas pelo IBM Business Process Manager. Por exemplo, o WebSphere Enterprise Service Bus pode ser instalado juntamente com uma aplicação SAP para alojar um IBM WebSphere Adapter para SAP e para transformar mensagens SAP antes de enviar essa informação pela rede para um processo empresarial coreografado por IBM Business Process Manager.

### **Sistema central de destino para o serviço de mensagens ou fila:**

Um sistema central de destino para o serviço de mensagens ou fila fornece a função de serviço de mensagens num servidor. Um servidor torna-se o sistema central de destino para o serviço de mensagens quando é configurado como destino do serviço de mensagens.

Um motor de serviço de mensagens é executado num servidor. O motor do serviço de mensagens fornece funções de serviço de mensagens e um ponto de ligação entre as aplicações e o barramento. A comunicação assíncrona da Arquitectura de Componente de Serviço (SCA, Service Component Architecture), a importação e exportação JMS, o processamento interno assíncrono utilizam filas de mensagens no motor do serviço de mensagens.

O ambiente de implementação liga a origem da mensagem ao destino da mensagem através do barramento durante a implementação dos módulos da aplicação. Conhecer a origem da mensagem e o destino da mensagem ajuda a determinar o tipo de ambiente de implementação necessário.

As aplicações podem armazenar dados sistemáticos num armazenamento de dados, que é um conjunto de tabelas numa base de dados ou esquema ou num armazenamento de ficheiros. O motor do serviço de mensagens utiliza uma ocorrência de uma origem de dados JDBC para interagir com essa base de dados.

A configuração do sistema central de destino para o serviço de mensagens deve ser efectuada durante a definição do ambiente de implementação utilizando **Server (Servidor)** na consola administrativa ou designando o servidor como sistema central de destino durante a instalação do software.

#### *Armazenamentos de dados:*

Todos os motores de serviço de mensagens podem utilizar um armazenamento de dados, que é um conjunto de tabelas numa base de dados ou esquema que armazena dados sistemáticos.

Todas as tabelas do armazenamento de dados são retidas no mesmo esquema de base de dados. É possível criar cada um dos armazenamentos de dados numa base de dados individual. Em alternativa, é possível criar múltiplos armazenamentos de dados na mesma base de dados, utilizando cada um dos armazenamentos de dados um esquema diferente.

O motor do serviço de mensagens utiliza uma ocorrência da origem de dados JDBC para interagir com a base de dados que contém o armazenamento de dados para esse motor de serviço de mensagens.

### **Fornecedores JDBC:**

É possível utilizar fornecedores JDBC para interagir aplicações com bases de dados relacionais.

As aplicações utilizam fornecedores JDBC para interagir com bases de dados relacionais. O fornecedor JDBC fornece a classe de implementação do controlador JDBC específico para acesso a um tipo específico de base de dados. Para criar um conjunto de ligações para determinada base de dados, deverá associar

uma origem de dados com o fornecedor JDBC. Em conjunto, o fornecedor JDBC e os objectos de origem de dados são funcionalmente equivalentes ao fornecedor de ligação Java EE Connector Architecture (JCA), que proporciona a conectividade com bases de dados não-relacionais.

Consulte os exemplos para Configuração típica de ambiente autónomo e Configuração típica de ambiente de implementação no tópico anterior.

Para obter mais informações acerca de fornecedores JDBC, consulte “JDBC providers (fornecedores JDBC)” no Information Center do WebSphere Application Server.

### **Barramento de integração de serviço para IBM Business Process Manager:**

O barramento de integração de serviço é um mecanismo de comunicações gerido que suporta integração de serviços através do serviço de mensagens assíncrono e síncrono. Um barramento consiste em interligar motores de serviço de mensagens que gerem recursos de barramento. É uma das tecnologias WebSphere Application Server em que o IBM Business Process Manager é baseado.

Um único barramento de serviço de integração e único motor de serviço de mensagens utiliza o mesmo esquema de base de dados que a base de dados do produto por predefinição. Cada ambiente de implementação tem o próprio barramento. O único barramento é denominado **BPM.deployment\_environment\_name.Bus**.

Um barramento destino é um endereço lógico ao qual as aplicações podem ligar-se como produtores, consumidores ou ambos. Uma fila destino é um barramento destino utilizado para serviço de mensagens ponto a ponto.

Cada um dos barramentos possui um ou mais membros de barramento, em que cada um é um servidor ou um conjunto.

A *topologia de barramento* é a disposição física do servidor de aplicações, motores do serviço de mensagens, gestores de filas WebSphere MQ e, o padrão de ligações de barramento e ligações entre eles, que representam o barramento do serviço da empresa.

### **Aplicações e módulos de serviços**

Um módulo de serviço é um módulo de Arquitectura de Componentes de Serviços (SCA) que fornece serviços no ambiente em tempo de execução. Quando é implementado um módulo de serviço para o IBM Business Process Manager, é construída uma aplicação de serviço associada que é empacotada como um ficheiro de Arquivo de Empresas (EAR).

Os módulos de serviço são unidades básicas de implementação e podem conter componentes, bibliotecas e módulos de simulação utilizados pela aplicação de serviços associada. Os módulos de serviço têm exportações e, opcionalmente, importações para definir as relações entre módulos e solicitadores e fornecedores de serviços. O WebSphere Process Server suporta módulos para serviços de negócios e módulos de mediação. Ambos os módulos e módulos de mediação são tipos de módulos SCA. Um módulo de mediação permite a comunicação entre aplicações ao converter a chamada para execução de serviços para um formato compreendido pelo destino, passando o pedido e devolvendo o resultado ao autor. Um módulo para um serviço de negócios implementa a lógica para um processo empresarial. Contudo, um módulo pode também incluir a mesma lógica de mediação que pode ser empacotada num módulo de mediação.

### **Implementar uma aplicação de serviços**

O processo de implementar um ficheiro EAR que contém uma aplicação de serviços é o mesmo que o de implementar um ficheiro EAR. Os valores de parâmetros de mediação podem ser alterados na altura de implementação. Depois de ser implementado um ficheiro EAR que contém um módulo SCA, podem ser revistos detalhes acerca da aplicação de serviços e o seu módulo associado. Pode ser visto como um

módulo de serviço está ligado a solicitadores de serviços (através de exportações) e solicitadores de serviços (através de importações).

## **Ver detalhes de módulos SCA**

Os detalhes de módulos de serviços que podem ser vistos dependem do módulo SCA. Incluem os seguintes atributos.

- Nome do módulo SCA
- Descrição de módulo SCA
- Nome de aplicação associada
- Informação de versão de módulo SCA, se o módulo tiver uma versão
- Importações de módulos SCA:
  - As interfaces de importações são definições abstractas que descrevem como um módulo SCA acede a um serviço.
  - As associações de importação são definições concretas que especificam o mecanismo físico pelo qual um módulo SCA acede a um serviço. Por exemplo, utilizando SOAP/HTTP.
- Exportações de módulos SCA:
  - As interfaces de exportação são definições abstractas que descrevem como os solicitadores de serviços acedem a um módulo SCA.
  - As associações de exportação são definições concretas que especificam o mecanismo físico pelo qual um solicitador de serviços acede a um módulo SCA e, indirectamente, a um serviço.
- Propriedades de módulos SCA

### **Importações e associações de importação:**

Importações definem interações entre módulos SCA e fornecedores de serviços. Os módulos SCA utilizam importações para permitir que os componentes acedam aos serviços externos (serviços que estão fora do módulo SCA) utilizando uma representação local. As associações de importações definem a forma específica em que um serviço externo é acedido.

Se os módulos SCA não necessitarem de aceder serviços externos, não necessitam de ter importações. Os módulos de mediação têm, geralmente, uma ou mais importações que são utilizadas para passar mensagens ou pedidos aos seus destinos.

### **Interfaces e associações**

Uma importação de módulo SCA necessita pelo menos de uma interface e tem uma única associação.

- As interfaces de importação são definições abstractas que definem um conjunto de operações utilizando a Linguagem de Descrição de Serviços da Web (Web Services Description Language) (WSDL), uma língua XML para descrever serviços da web. Um módulo SCA pode ter várias interfaces de importação.
- As associações de importação são definições concretas que especificam o mecanismo físico que os módulos SCA utilizam para aceder a um serviço.

### **Associações de importação suportadas**

O IBM Business Process Manager suporta as seguintes associações de importações:

- As associações SCA ligam módulos SCA a outros módulos SCA. As associações SCA também são referidas como associações padrão.
- As associações de Serviço da Web permitem que componentes chamem serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.

Pode ser utilizada uma associação SOAP1.1/HTTP ou SOAP1.2/HTTP baseada no API Java de Serviços da Web XML (JAX-WS), que permite a interacção com serviços utilizando associações literais de documentos ou RPC e que utiliza processadores JAX-WS para personalizar chamadas para execução. É fornecida outra associação SOAP1.1/HTTP para permitir a interacção com serviços que utilizam uma associação de codificação RPC ou onde houver uma necessidade de utilizar processadores JAX-RPC para personalizar as chamadas para execução.

- As associações HTTP permitem-lhe aceder a aplicações utilizando o protocolo HTTP.
- As associações de importação de JavaBeans de Empresa (Enterprise JavaBeans) (EJB) permitem aos componentes SCA chamar serviços fornecidos pela lógica de negócios Java EE executados num servidor Java EE.
- As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é alcançada através da utilização de adaptadores de recursos.
- As associações 1.1 do Serviço de Mensagens Java (JMS) permitem interoperacionalidade com o fornecedor de mensagens padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente.
- As associações genéricas JMS permitem interoperacionalidade com fornecedores JMS de terceiros que se integram com o WebSphere Application Server utilizando o JMS Application Server Facility (ASF).
- As associações JMS do WebSphere MQ permitem interoperacionalidade com os fornecedores JMS baseados no WebSphere MQ. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente. Se quiser utilizar o WebSphere MQ como fornecedor JMS, utilize associações JMS do WebSphere MQ.
- As associações do WebSphere MQ permitem interoperacionalidade com o WebSphere MQ. As associações do WebSphere MQ apenas podem ser utilizadas com gestores de fila remotos por via de uma ligação de cliente WebSphere MQ; não podem ser utilizadas com gestores de fila locais. Deverão ser utilizadas associações do WebSphere MQ se for desejado comunicar com aplicações nativas do WebSphere MQ.

### **Chamada para execução dinâmica de serviços**

Podem ser chamados serviços através de qualquer associação de importação suportada. Um serviço é normalmente encontrado num ponto de terminação especificado na importação. Este ponto de terminação chama-se ponto de terminação estático. É possível chamar um serviço diferente ao substituir o ponto de terminação estático. A substituição dinâmica de pontos de terminação estáticos permite chamar um serviço em qualquer outro ponto de terminação através de qualquer associação de importação suportada. A chamada dinâmica de serviços também permite chamar para execução um serviço em que a associação de importação suportada não tem um ponto de terminação estático.

Uma importação com uma associação associada é utilizada para especificar o protocolo e a sua configuração para uma chamada para execução dinâmica. A importação utilizada para a invocação dinâmica pode ser ligada ao componente de chamada ou pode ser seleccionada de forma dinâmica no tempo de execução.

Para chamadas para execução de Serviços da web e SCA, é possível fazer uma chamada para execução dinâmica sem uma importação, com o protocolo e a configuração deduzidas do URL do ponto de terminação. O tipo de destino de chamada para execução é identificado do URL do ponto de terminação. Se for utilizada uma importação, o URL deve ser compatível com o protocolo da associação de importação.

- Um URL de SCA indica a chamada para execução de outro módulo SCA.
- Um URL HTTP ou JMS indica, por defeito, a chamada para execução de um serviço da web; para estes URLs é possível fornecer um valor de tipo de associação adicional que indique que o URL representa uma invocação por via de uma associação HTTP ou JMS.

- Para um URL HTTP de serviço da web, o normal é utilizar SOAP 1.1, e pode ser especificado um valor de associação que indique a utilização de SOAP 1.2.

### **Exportações e associações de exportações:**

As exportações definem interacções entre módulos SCA e solicitadores de serviços. Os módulos SCA utilizam exportações que oferecem serviços a outros. As associações de exportações definem a forma específica que um módulo SCA é acedido por solicitadores de serviços.

### **Interfaces e associações**

Uma exportação de módulo SCA necessita de pelo menos uma interface.

- As interfaces de exportações são definições abstractas que definem um conjunto de operações utilizando a Linguagem de Descrição de Serviços da Web (WSDL, Web Services Description Language), uma língua XML para descrever serviços da web. Um módulo SCA pode ter várias interfaces de exportação.
- As associações de exportação são definições concretas que especificam o mecanismo físico que os solicitadores de serviços utilizam para aceder a um serviço. Normalmente, uma exportação de módulo SCA tem uma associação especificada. Uma exportação sem nenhuma associação especificada é interpretada pelo ambiente em tempo de execução como uma exportação com uma associação SCA.

### **Associações de exportação suportadas**

O IBM Business Process Manager suporta as seguintes associações de exportação:

- As associações SCA ligam módulos SCA a outros módulos SCA. As associações SCA também são referidas como associações padrão.
- As associações de Serviços da Web permitem que as exportações sejam chamadas como serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.  
Pode utilizar uma associação SOAP1.1/HTTP ou SOAP1.2/HTTP baseada no API Java para Serviços da Web XML (JAX-WS), o que permite a interacção com serviços utilizando associações literais de documentos ou RPC e que utiliza processadores JAX-WS para personalizar as chamadas para execução. É fornecida outra associação SOAP1.1/HTTP para permitir a interacção com serviços que utilizam uma associação de codificação RPC ou onde houver uma necessidade de utilizar processadores JAX-RPC para personalizar as chamadas para execução.
- As associações HTTP permitem que as exportações sejam acedidas utilizando o protocolo HTTP.
- As associações de exportação Enterprise JavaBeans (EJB) permitem que os componentes SCA sejam expostos como EJBs, para que a lógica de negócios Java EE possa chamar componentes SCA que de outra forma lhes estariam indisponíveis.
- As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é alcançada através da utilização de adaptadores de recursos.
- As associações 1.1 do Serviço de Mensagens Java (JMS) permitem interoperacionalidade com o fornecedor de mensagens padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente.
- As associações genéricas JMS permitem interoperacionalidade com fornecedores JMS de terceiros que se integram com o WebSphere Application Server utilizando o JMS Application Server Facility (ASF).
- As associações JMS do WebSphere MQ permitem interoperacionalidade com os fornecedores JMS baseados no WebSphere MQ. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente. Se quiser utilizar o WebSphere MQ como fornecedor JMS, utilize associações JMS do WebSphere MQ.
- As associações do WebSphere MQ permitem interoperacionalidade com o WebSphere MQ. Utilize uma ligação remota (ou de cliente) para se ligar a um gestor de fila MQ numa máquina remota. Uma ligação local (ou associações) é uma ligação directa ao WebSphere MQ. Isto pode apenas ser utilizado

para uma ligação a um gestor de fila MQ na mesma máquina. O WebSphere MQ vai permitir ambos os tipos de ligação, mas as associações MQ apenas suportam a ligação "remota" (ou "de cliente").

### **Módulos de mediação:**

Os módulos de mediação são módulos de Arquitectura de Componentes de Serviços (Service Component Architecture) (SCA) que podem alterar o formato, conteúdo ou destino dos pedidos de serviços.

Os módulos de mediação operam em mensagens que estão em curso entre solicitadores de serviços e fornecedores de serviços. Pode encaminhar mensagens para vários fornecedores de serviços e para corrigir o conteúdo ou a forma da mensagem. Os módulos de mediação podem disponibilizar funções como o registo de mensagens e processamento de erros personalizados de acordo com os seus requisitos.

Certos aspectos dos módulos de mediação podem ser alterados, a partir da consola administrativa, sem ser necessário implementá-los novamente.

### **Componentes de módulos de mediação**

Os módulos de mediação contêm os seguintes itens:

- Importações, que definem interacções entre os módulos SCA e os fornecedores de serviços. Permitem que os módulos SCA chamem serviços externos como se fossem locais. As importações de módulos de mediação podem ser vistas e a sua associação alterada.
- Exportações, que definem interacções entre módulos SCA e solicitadores de serviços. Permitem que um módulo SCA ofereça um serviço e defina as interfaces externas (pontos de acesso) de um módulo SCA. As exportações dos módulos de mediação podem ser vistas.
- Os componentes SCA, que são blocos de construção para módulos SCA modules, como os módulos de mediação. É possível criar e personalizar graficamente módulos e componentes SCA, utilizando o Integration Designer. Depois da implementação de um módulo de mediação, é possível personalizar certos aspectos do mesmo a partir da consola administrativa, sem ser necessário implementá-lo novamente.

Normalmente, os módulos de mediação contêm um tipo específico de componente SCA chamado *componente de fluxo de mediação*. Os componentes de fluxo de mediação definem fluxos.

Um componente de fluxo de mediação pode algumas ou nenhuma primitivas de mediação. O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação que fornecem funcionalidade ao encaminhamento e conversão de mensagens. Para uma flexibilidade de primitiva de mediação adicional, utilize a primitiva de mediação personalizada para chamar a lógica de personalização.

O objectivo de um módulo de mediação que não contém um componente de fluxo é o de converter pedidos de serviços de um protocolo para outro. Por exemplo, pode ser feito um pedido de serviço utilizando SOAP/JMS mas pode necessitar de conversão para SOAP/HTTP antes de ser enviado.

**Nota:** Certas alterações podem ser vistas e executadas a partir do IBM Business Process Manager. Contudo, não podem ser vistas nem executadas alterações aos componentes SCA dentro de um módulo a partir do IBM Business Process Manager. O Integration Designer é utilizado para personalizar componentes SCA.



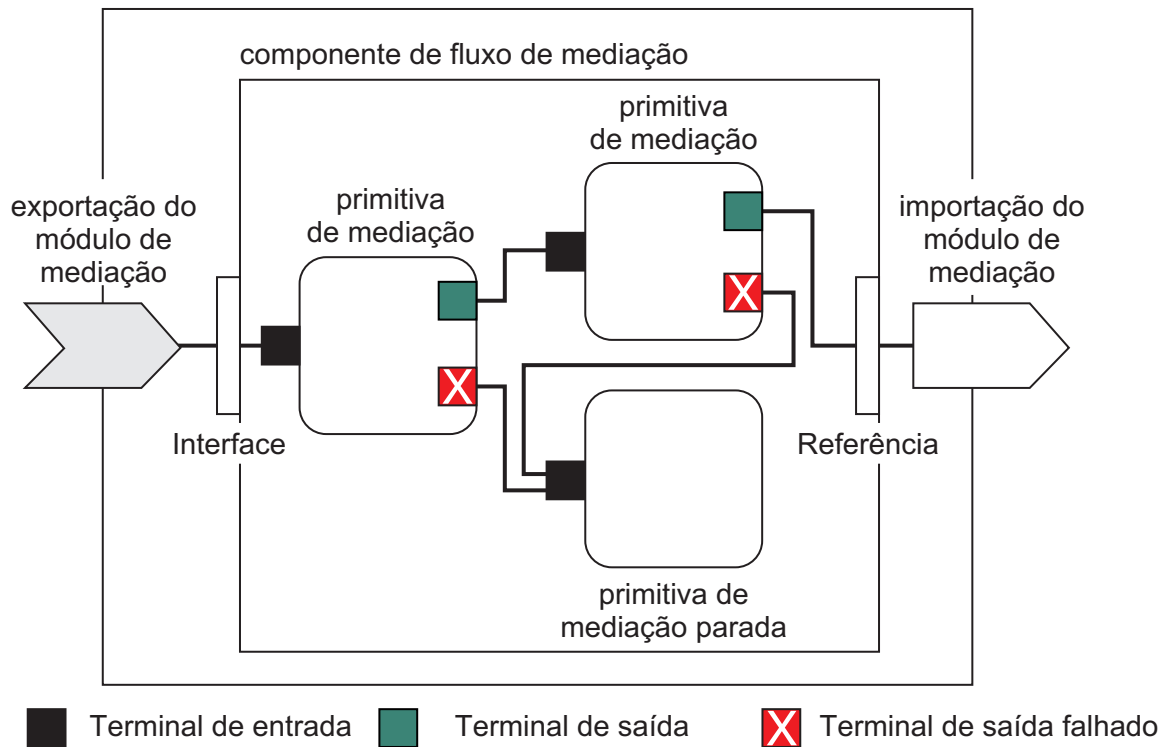


Figura 43. Exemplo simplificado de um módulo de mediação. O módulo de mediação contém um componente de fluxo de mediação, que contém primitivas de mediação.

- Propriedades

As primitivas de mediação têm propriedades, algumas das quais podem ser apresentadas na consola administrativa como propriedades adicionais de um módulo SCA.

Para que as propriedades das primitivas sejam visíveis a partir da consola administrativa do IBM Business Process Manager, o programador de integração deve promover as propriedades. Certas propriedades deixam-se ser configuradas administrativamente e o Integration Designer descreve-as como propriedades passíveis de promoção, porque podem ser promovidas do ciclo de integração para o ciclo administrativo. Outras propriedades não são adequadas para a configuração administrativa, porque modificá-las pode afectar o fluxo de mediação de uma forma que o módulo de mediação necessite de ser novamente implementado. O Integration Designer lista as propriedades seleccionáveis para promover de entre as propriedades promovidas de uma primitiva de mediação.

A consola administrativa do IBM Business Process Manager pode ser utilizada para alterar o valor de propriedades promovidas sem ser necessário implementar novamente um módulo de mediação ou reiniciar o servidor ou módulo.

Na generalidade, os fluxos de mediação utilizam imediatamente as alterações de propriedades. Contudo, se ocorrerem alterações de propriedades numa célula de gestão de implementação, estas tomam efeito em cada nó, enquanto este é sincronizado. Para além disso, os fluxos de mediação que estão em curso continuam a utilizar valores anteriores.

**Nota:** Da consola administrativa só é possível alterar os valores de propriedade e não os grupos, nomes ou tipos de propriedades. Se for desejável alterar os grupos, nomes ou tipos de propriedades, é necessária a utilização do Integration Designer.

- Um módulo de mediação ou biblioteca dependente também podem definir sub-fluxos. Um sub-fluxo encapsula um conjunto de primitivas de mediação ligadas como uma peça reutilizável de lógica de integração. Uma primitiva pode ser adicionada a um fluxo de mediação para chamar um sub-fluxo.



## Implementar módulos de mediação

Os módulos de mediação são criados utilizando o Integration Designer, e são, geralmente, implementados no IBM Business Process Manager, dentro de um ficheiro de arquivo de empresa (enterprise archive, EAR).

O valor de propriedades promovidas pode ser alterado na altura de implementação.

Pode ser exportado um módulo de mediação do Integration Designer, causando com que o Integration Designer armazene o módulo de mediação dentro de um ficheiro de arquivo Java (Java archive) e o ficheiro JAR dentro de um ficheiro EAR. O ficheiro EAR pode ser implementado com a instalação de uma nova aplicação a partir da consola administrativa.

Os módulos de mediação podem ser visto como uma entidade. Contudo, os módulos SCA são definidos por um número de ficheiros XML armazenados num ficheiro JAR.

### Exemplo de ficheiro EAR, que contém um módulo de mediação

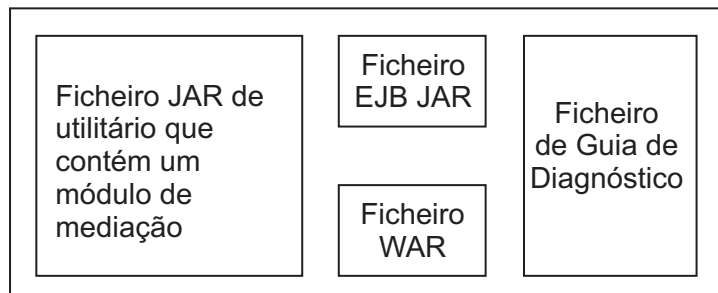


Figura 44. Um exemplo simplificado de um ficheiro EAR que contém um módulo de mediação. O ficheiro EAR contém JARs. O ficheiro JAR utilitário contém um módulo de mediação.

### Primitivas de mediação:

Os componentes de fluxo de mediação operam em fluxos de mensagens entre componentes de serviços. As funções de um componente de mediação são implementadas por *primitivas de mediação*, que implementam tipos de implementação padrão.

Um componente de fluxo de mediação tem um ou mais fluxos. Por exemplo, um para pedidos e um para respostas.

O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação, que implementam funções de mediação padrão para módulos de mediação ou módulos implementados no IBM Business Process Manager. Se necessitar de funções de mediação especiais, pode desenvolver as suas próprias primitivas de mediação personalizadas.

Uma primitiva de mediação define uma operação “in” que processa ou conduz mensagens que são representadas por objectos de serviços de mensagens (SMOs). Uma primitiva de mediação pode também definir operações “out” que enviam mensagens para outro componente ou módulo.

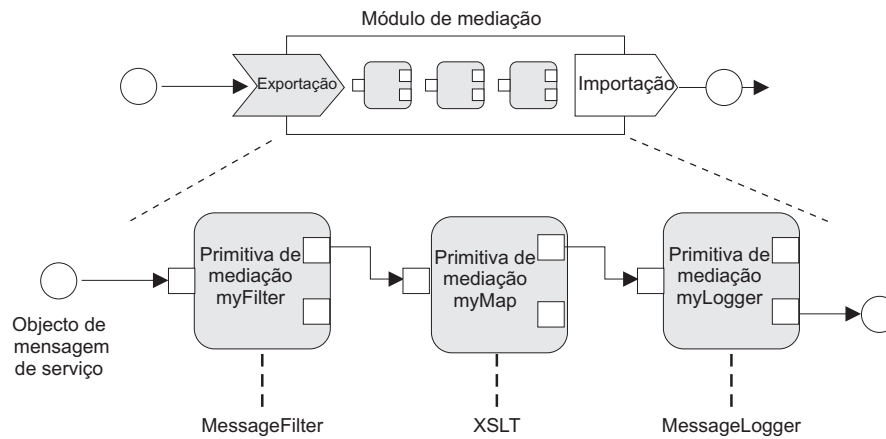


Figura 45. Módulo de mediação que contém três primitivas de mediação

Pode utilizar o Integration Designer para configurar as primitivas de mediação e definir as suas propriedades. Algumas destas propriedades podem ser tornadas visíveis ao administrador de tempo de execução ao promovê-las. Qualquer propriedade primitiva de mediação que possa ser promovida, pode também ser uma propriedade dinâmica. Uma propriedade dinâmica pode ser substituída, no tempo de execução, utilizando um ficheiro de política.

O Integration Designer também lhe permite modelar e montar graficamente componentes de fluxo de mediação a partir das primitivas de mediação e módulos de montagem ou módulos dos componentes de fluxo de mediação. A consola administrativa refere-se a módulos de mediação e módulos como os módulos SCA.

O Integration Designer também permite a definição dos sub-fluxos em módulos ou nas suas bibliotecas independentes. Um sub-fluxo pode conter qualquer primitiva de mediação, à excepção da primitiva de mediação de Resolução de Política. Um sub-fluxo é chamado um fluxo de pedido ou resposta, ou de outro sub-fluxo, utilizando a primitiva de mediação de Sub-fluxos. As propriedades promovidas das primitivas de mediação num sub-fluxo são expostas como propriedades nas primitivas de mediação de Sub-fluxos. Estes podem então ser promovidos outra vez até chegarem ao nível de módulo, altura em que podem ser modificados pelo administrador de tempo de execução.

### Primitivas de mediação suportadas

O seguinte conjunto de primitivas de mediação é suportado pelo IBM Business Process Manager:

#### Mapa de objecto de negócio

Converte mensagens.

- Define conversões de mensagens utilizando um mapa de objecto de negócio, que pode ser re-utilizado.
- Permite-lhe definir graficamente conversões de mensagens, utilizando o editor de mapa de objecto de negócio.
- Pode alterar o conteúdo de uma mensagem.
- Pode converter um tipo de mensagem de entrada num tipo diferente de mensagem de saída.

## Mediação Personalizada

Permite-lhe implementar a sua lógica de mediação em código Java. A primitiva de Mediação Personalizada combina a flexibilidade de uma primitiva de mediação definido por um utilizador com a simplicidade de uma primitiva de mediação pré-definida. Pode criar conversões complexas e padrões de encaminhamento ao:

- Criar código Java.
- Criar as suas próprias propriedades.
- Adicionar novos terminais.

Pode chamar um serviço a partir de uma primitiva de Mediação Personalizada, mas a primitiva de mediação de Invocação de Serviços é concebida para chamar serviços e fornece funcionalidade adicional, como a de tentar novamente.

## Rotina de Tratamento de Dados

Permite converter uma parte de uma mensagem. É utilizado para converter um elemento de uma mensagem de um formato físico para uma estrutura lógica ou de um a estrutura lógica para um formato físico. A utilização primária da primitiva é a de converter um formato físico, como uma cadeia de texto dentro de um objecto de Mensagem de Texto JMS, para uma estrutura lógica de um Objecto de Negócio e vice-versa. Esta mediação é normalmente utilizada para:

- Transforme a secção da mensagem de entrada de uma estrutura definida para outra - um exemplo disto seria em que SMO inclui um valor de cadeia que é delimitado por vírgulas e que pretende analisar num objecto de negócios específico.
- Alterar o tipo de mensagem - um exemplo seria quando uma exportação JMS fosse configurada para utilizar uma associação básica de dados JMS e, dentro do módulo de mediação, o programador de integração decidisse que o conteúdo devesse ser inflacionado para uma estrutura BO específica.

## Procura da Base de Dados

Modifica mensagens, utilizando informação de uma base de dados fornecida pelo utilizador.

- Deve definir uma base de dados, uma origem de dados e quaisquer definições de autenticação do servidor para serem utilizados pela primitiva de mediação Procura de Base de Dados. Utilize a consola administrativa para o ajudar a fazê-lo.
- A primitiva de mediação Procura da Base de Dados pode ler a partir de apenas uma tabela.
- A coluna chave especificada deve conter um valor único.
- Os dados nas colunas de valores devem ser de um tipo de esquema XML simples ou de um tipo de esquema XML que expande um tipo de esquema XML simples.

## Procura do Ponto de Terminação

Permite o encaminhamento dinâmico de pedidos, ao procurar pontos de terminação de serviços no repositório.

- A informação de pontos de terminação de serviços é obtida de um WebSphere Service Registry and Repository (WSRR). O registo WSRR pode ser local ou remoto.
- Podem ser efectuadas alterações do registo a partir da consola administrativa do WSRR.
- O IBM Business Process Manager necessita de saber que registo utilizar e, por isso, devem ser criadas definições de acesso ao WSRR, utilizando a consola administrativa do IBM Business Process Manager.

## Emissor de Eventos

Melhora a supervisão ao deixá-lo enviar eventos de dentro de um componente de fluxo de mediação.

- A acção de mediação pode ser suspensa ao desmarcar a caixa de verificação.
- Pode ver eventos do Emissor de Eventos utilizando o navegador de Eventos de Base Comum no IBM Business Process Manager.
- Por razões de desempenho, só deverá conseguir enviar eventos numa altura significativa de um fluxo de mediação.

- É possível definir as partes da mensagem que o evento contém.
- Os eventos são enviados na forma de Eventos de Base Comum para o servidor da Infra-Estrutura de Eventos Comuns.
- Para utilizar completamente a informação do Emissor de Eventos, os consumidores de eventos necessitam de compreender a estrutura dos Eventos de Base Comum. Os Eventos de Base comum têm um esquema geral, mas isto não modela os dados específicos da aplicação, que é contida nos elementos de dados expandidos. Para modelar os elementos de dados expandidos, as ferramentas do Integration Designer geram um ficheiro de definição de catálogo de eventos de Infra-Estrutura de Eventos Comuns para cada uma das primitivas de mediação do Emissor de Eventos. Os ficheiros de definição de catálogo de eventos são artefactos de exportação fornecidos para ajudar; não são utilizados pelo Integration Designer ou pelo tempo de execução do IBM Business Process Manager. Os ficheiros de definição de catálogo de evento devem ser consultados quando são criadas aplicações para consumir os eventos de Emissão de Eventos.
- Pode ser especificada outra supervisão a partir do IBM Business Process Manager. Por exemplo, podem ser supervisionados eventos que serão emitidos das importações e exportações.

**Falha** Pára um caminho em particular no fluxo e gera uma excepção.

#### **Concentrar**

Ajuda a agregar (combinar) mensagens.

- Pode apenas ser utilizado em conjunto com a primitiva de mediação Dispersar.
- Juntas, as primitivas de mediação Dispersar e Concentrar permitem a agregação de dados numa mensagem de saída.
- A primitiva de mediação Concentrar recebe mensagens até ser atingido um ponto de decisão, em que uma mensagem se torna de saída.
- O contexto partilhado deve ser utilizado para reter os dados de agregação.

#### **Dispersar**

Ajuda a dividir e agregar (combinar) mensagens.

- Juntas, as primitivas de mediação Dispersar e Concentrar permitem a agregação de dados numa mensagem de saída.
- No modo iterativo, a primitiva de mediação Dispersar permite iterar uma única mensagem de entrada que contém um elemento repetido. Para cada ocorrência do elemento repetido é enviada uma mensagem.
- O contexto partilhado deve ser utilizado para reter os dados de agregação.

#### **Compositor de Cabeçalhos HTTP**

Fornece um mecanismo para gerir cabeçalhos de mensagens HTTP.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens HTTP.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos HTTP.

#### **Mapeamento**

Converte mensagens.

- Permite-lhe executar transformações Extensible Stylesheet Language (XSL) (linguagem de folha de estilos extensível) ou transformações Business Object Map (mapeamento do objecto de negócios).
- Se transformar mensagens ao utilizar uma transformação XSLT 1.0 ou XSLT 2.0 ou uma transformação Business Object Map (mapa de objecto de negócios). As transformações XSL funcionam numa serialização XML da mensagem, em que a transformação de Business Object Map funciona nos Service Data Objects (SDO) (objectos de dados de serviços).

#### **Compositor de Elementos de Mensagem**

Fornece um mecanismo simples para definir o conteúdo de mensagens.

- Pode alterar, adicionar ou apagar elementos de mensagens.
- Não altera o tipo de mensagem.
- Os dados nas colunas de valores devem ser de um tipo de esquema XML simples ou de um tipo de esquema XML que expande um tipo de esquema XML simples.

### **Filtro de Mensagens**

Encaminha mensagens por caminhos diferentes, com base no conteúdo das mensagens.

- A ação de mediação pode ser suspensa ao desmarcar a caixa de verificação.

### **Registador de Mensagens**

Regista as mensagens numa base de dados relacional ou através do próprio registador de mensagens personalizado. As mensagens são guardadas como XML e, por isso, os dados podem ser depois processados por aplicações que detectem XML.

- A ação de mediação pode ser suspensa ao desmarcar a caixa de verificação.
- O esquema de base de dados relacional (estrutura de tabela) é definido pela IBM.
- Por defeito, a primitiva de mediação Registador de Mensagens utiliza a base de dados Comum. O ambiente em tempo de execução correlaciona a origem de dados em **jdbc/mediation/messageLog** com a base de dados Comum.
- Podem ser definidas classes de implementação de Processador para personalizar o comportamento do registador personalizado. Opcionalmente, podem ser fornecidas classes de implementação de Formatador, classes de implementação de Filtro ou ambas, para personalizar o comportamento do registador personalizado.

### **Compositor de Cabeçalhos MQ**

Fornece um mecanismo para gerir cabeçalhos em mensagens MQ.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens MQ.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos MQ.

### **Resolução de Política**

Permite a configuração dinâmica de pedidos, ao procurar, num repositório, pontos de terminação de serviços e ficheiros de política associados.

- Pode ser utilizado um ficheiro de política para substituir de forma dinâmica as propriedades promovidas de outras primitivas de mediação.
- A informação de pontos de terminação de serviços e informação de política são obtidas de um WebSphere Service Registry and Repository (WSRR). O registo WSRR pode ser local ou remoto.
- Podem ser efectuadas alterações do registo a partir da consola administrativa do WSRR.
- O IBM Business Process Manager necessita de saber que registo utilizar e, por isso, devem ser criadas definições de acesso ao WSRR, utilizando a consola administrativa do IBM Business Process Manager.

### **Chamada de Serviços**

Chama um serviço dentro de um fluxo de mediação, em vez de esperar até ao fim do fluxo de mediação e utilizar o mecanismo de chamada.

- Se o serviço devolver uma falha, pode ser feita uma nova tentativa ou chamar outro serviço.
- A primitiva de mediação de Chamada de Serviços pode ser utilizada sozinha para chamadas simples de serviço ou em conjunto com outras primitivas de mediação para mediações complexas.

### **Definir Tipo de Mensagem**

Durante o desenvolvimento da integração, permite tratar campos de mensagem pouco tipificados como se fossem muito tipificados. Um campo é pouco tipificado se puder conter mais de um tipo de dados. Um campo é muito tipificado se o seu tipo e estrutura interna forem conhecidos.

- No tempo de execução, a primitiva de mediação Definir Tipo de Mensagens permite verificar se o conteúdo de uma mensagem corresponde aos tipos de dados esperados.

### **Compositor de Cabeçalhos SOAP**

Fornecer um mecanismo para gerir cabeçalhos de mensagens SOAP.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens SOAP.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos SOAP.

**Parar** Pára um caminho específico do fluxo sem gerar uma excepção.

### **Filtro de Tipos**

Permite direccionar mensagens num caminho diferente de um fluxo, com base no seu tipo.

### **Obter WebSphere eXtreme Scale**

Pode obter informações de um ambiente de memória cache do servidor eXtreme Scale.

- Pode procurar valores na memória cache e armazená-los como elementos na mensagem através de uma mensagem.
- Ao combinar as primitivas de mediação de armazenamento e de obtenção eXtreme Scale, pode colocar na memória cache a resposta de um sistema emissor. Os pedidos futuros não irão requerer acesso a esse sistema emissor.
- Deverá criar as definições eXtreme Scale ao utilizar a consola administrativa WebSphere ESB, pelo que pode especificar o servidor eXtreme Scale a utilizar.

### **Armazém WebSphere eXtreme Scale**

Pode armazenar informações num ambiente de memória cache do servidor eXtreme Scale.

- Pode armazenar informação numa memória cache eXtreme Scale ao utilizar uma chave e um objecto.
- Ao combinar as primitivas de mediação de armazenamento e de obtenção eXtreme Scale, pode utilizar a primitiva de mediação de armazenamento para armazenar dados na memória cache e utilizar a primitiva de mediação de obtenção para obter dados anteriormente gravados na memória cache.
- Deverá criar as definições eXtreme Scale ao utilizar a consola administrativa WebSphere ESB, pelo que pode especificar o servidor eXtreme Scale a utilizar.

### **Encaminhamento dinâmico:**

É possível encaminhar mensagens de várias formas utilizando pontos de terminação definidos no tempo de integração ou pontos de terminação determinados, de forma dinâmica, no tempo de execução.

O encaminhamento dinâmico cobre dois casos de encaminhamento de mensagens:

- O encaminhamento de mensagens em que o fluxo é dinâmico, mas todos os pontos finais possíveis são predefinidos num nóduo Service Component Architecture (SCA)(arquitectura de componente de serviço).
- O encaminhamento de mensagens em que o fluxo é dinâmico e a selecção de pontos finais também é dinâmica. Os pontos finais do serviço são seleccionados em tempo de execução, a partir de uma fonte externa

### **Seleccção de ponto de terminação dinâmico**

O tempo de execução tem a função de encaminhar mensagens de pedido e resposta para um endereço de ponto de terminação identificado por um elemento de cabeçalho de mensagem. Este elemento de cabeçalho de mensagem pode ser actualizado por primitivas de mediação, num fluxo de mediação. O endereço de ponto de terminação pode ser actualizado com informação de um registo, uma base de dados ou da própria mensagem. O encaminhamento de mensagens de resposta aplica-se apenas quando a resposta está a ser enviada por uma exportação de serviço da web WAX-WS.

Para que o tempo de execução implemente um encaminhamento dinâmico num pedido ou numa resposta, o módulo SCA deve ter o conjunto de propriedades Use dynamic endpoint if set in the message header (Utilizar pontos de terminação dinâmico se definidos no cabeçalho de mensagens).



Os programadores de integração podem definir a propriedade Utilizar o ponto de terminação dinâmico se for definido no cabeçalho da mensagem ou podem promovê-la (torná-la visível no tempo de execução), para que o administrador de tempo de execução a possa definir. As propriedades de módulos podem ser vistas na janela Module Properties (Propriedades do Módulo). Para ver a janela, é necessário fazer clique em **Applications (Aplicações) > SCA Modules (Módulos SCA) > Module Properties (Propriedades de Módulos)**. O programador de integração atribui nomes alternativos às propriedades promovidas e estes nomes são apresentados na consola administrativa.

## **Registo**

O IBM WebSphere Service Registry and Repository (WSRR) pode ser utilizado para armazenar informação de ponto de terminação e depois criar módulos SCA para obter pontos de terminação do registo do WSRR.

Quando se desenvolvem módulos SCA, é utilizado a primitiva de mediação Pesquisa de Pontos de Terminação para permitir que um fluxo de mediação consulte um registo WSRR sobre um ponto de terminação de serviço ou um conjunto de pontos de terminação de serviço. Se um módulo SCA obtiver um conjunto de pontos de terminação, este deve utilizar outra primitiva de mediação para seleccionar o preferido.

## **Controlo da política de mediação dos pedidos de serviço:**

É possível utilizar as políticas de mediação para controlar fluxos de mediação entre solicitações de serviços e fornecedores de serviços.

É possível controlar fluxos de mediação utilizando políticas de mediação guardadas no IBM WebSphere Serviço de Registo e Repositório (WSRR, Service Registry and Repository). A implementação da gestão da política de serviço do WSRR é baseada no Web Services Policy Framework (Política WS).

Para controlar os pedidos de serviço utilizando políticas de mediação, é necessário possuir módulos de Arquitectura de Componente de Serviço (SCA, Service Component Architecture) e documentos de política de mediação no registo WSRR.

## **Procedimento para anexar uma política de mediação a um pedido de serviço**

Quando é desenvolvido um módulo SCA que necessite de utilizar uma política de mediação, deve ser incluído uma primitiva de mediação Resolução de Política (Policy Resolution) no fluxo de mediação. Durante o tempo de execução, a mediação da Resolução Política (Policy Resolution) primitiva, obtém as informações da política de mediação a partir do registo. Assim, um módulo SCA deve conter um componente de fluxo de mediação para suportar o controlo da política de mediação dos pedidos de serviço.

No registo, é possível anexar uma ou mais políticas de mediação a um módulo SCA ou a um serviço destino utilizado pelo módulo SCA. As políticas de mediação anexadas podem ser utilizadas (dentro do âmbito) para todas as mensagens de serviço processadas por esse módulo SCA. As políticas de mediação podem conter anexos de políticas para definir condições. As condições de políticas de mediação permitem a aplicação de diferentes políticas de mediação dependendo em diferentes contextos. Além disso, as políticas de mediação podem ter classificações, que podem ser utilizadas para especificar o estado de controlo.

## **Registo e Repositório de Serviços WebSphere:**

O produto de Registo e Repositório de Serviços WebSphere (WSRR) permite armazenar, aceder e gerir informação acerca dos pontos de terminação de serviços e políticas de mediação. O WSRR pode ser utilizado para tornar as aplicações de serviço mais dinâmicas e mais adaptáveis a condições de negócios em constante alteração.



## Introdução

Os fluxos de mediação podem utilizar o WSRR como um mecanismo de procura dinâmica, fornecendo informação acerca dos pontos de terminação de serviço ou políticas de mediação.

Para configurar o acesso ao WSRR, criam-se documentos de definição do WSRR utilizando a consola administrativa. Alternativamente, os comandos de administração do WSRR podem ser utilizados a partir do cliente de script wsadmin. As definições do WSRR e as suas propriedades de ligação são o mecanismo utilizado para ligar a uma instância de registo e obter um ponto de terminação de serviço ou política de mediação.

### Pontos de Terminação de serviços

O WSRR pode ser utilizado para armazenar informação acerca de serviços já utilizados, que poderão ser utilizados ou que sejam de interesse. Estes serviços podem estar em sistemas próprios ou noutros sistemas. Por exemplo, uma aplicação pode utilizar o WSRR para localizar o serviço mais apropriado para satisfazer as suas necessidades funcionais e de desempenho.

Quando é desenvolvido um módulo SCA que necessita de aceder pontos de terminação de serviços a partir do WSRR, é necessário incluir a primitiva de mediação Procurar Pontos de Terminação no fluxo de mediação. No tempo de execução, a primitiva de mediação Pesquisa de Pontos de Terminação obtém pontos de terminação de serviços a partir do registo.

### Políticas de mediação

O WSRR também pode ser utilizado para armazenar informação de política de mediação. As políticas de mediação podem ajudar a controlar pedidos de serviços, ao substituir propriedades de módulos de forma dinâmica. Se o WSRR contiver políticas de mediação que estão ligadas a um objecto que representa o módulo SCA ou o serviço destino, então as políticas de mediação podem substituir as propriedades de módulos. Se for desejado que diferentes políticas de mediação se apliquem a contextos diferentes, é possível criar condições de política de mediação.

**Nota:** As políticas de mediação dizem respeito ao controlo de fluxos de mediação e não à segurança.

Quando é desenvolvido um módulo SCA que necessite de utilizar uma política de mediação, deve ser incluído uma primitiva de mediação Resolução de Política (Policy Resolution) no fluxo de mediação. Durante o tempo de execução, a mediação da Resolução Política (Policy Resolution) primitiva, obtém as informações da política de mediação a partir do registo.

### WebSphere eXtreme Scale:

Ao utilizar o produto WebSphere eXtreme Scale (eXtreme Scale) pode disponibilizar um sistema de memória cache que pode integrar com uma aplicação IBM Business Process Manager. Ao utilizar eXtreme Scale com IBM Business Process Manager pode melhorar os tempos de resposta e a fiabilidade do serviço e disponibilizar uma função de integração adicional.

eXtreme Scale age como uma grelha de dados na memória, elástica e escalável. A grelha de dados coloca na memória cache, particiona, duplica e gere dados de aplicação e lógica de negócios em vários servidores. Com eXtreme Scale, também pode obter qualidades de serviço como a integridade transaccional, a elevada disponibilidade e tempos de resposta previsíveis.

Pode utilizar os fluxos de mediação para aceder à função de memória cache eXtreme Scale ao incluir as primitivas de mediação WebSphere eXtreme Scale no seu fluxo. Quando desenvolve um módulo de arquitectura de componente de serviço (SCA) que necessita de armazenar informação numa memória cache eXtreme Scale, deverá incluir a primitiva de mediação de armazenamento WebSphere eXtreme Scale no fluxo de mediação. Se pretende recuperar informação de uma memória cache eXtreme Scale,

deverá incluir a primitiva de mediação de recuperação WebSphere eXtreme Scale. Ao combinar duas primitivas de mediação num fluxo de mediação, pode obter a resposta de memória cache de um sistema de sistema emissor, para que os pedidos futuros possam obter a resposta da memória cache.

Para configurar o acesso a eXtreme Scale, deverá criar uma definição WebSphere eXtreme Scale através da consola administrativa. Como alternativa, pode utilizar os comandos administrativos WebSphere eXtreme Scale do cliente de script wsadmin. Uma definição eXtreme Scale é um mecanismo utilizado pelas primitivas de mediação de obtenção e de armazenamento WebSphere eXtreme Scale para ligar a um servidor eXtreme Scale.

## **Clientes do Message Service**

Os clientes do Message Service estão disponíveis para C/C++ e .NET de forma a permitir que aplicações que não sejam Java se liguem ao Enterprise Service Bus.

O Message Service Clients for C/C++ and .NET fornece um API chamado XMS que tem o mesmo conjunto de interfaces que o API de Serviço de Mensagens Java (JMS). O Message Service Client for C/C++ contém duas implementações de XMS, uma para utilização por aplicações C e outra para utilização por aplicações C++. O Message Service Client for .NET contém uma implementação totalmente gerida de XMS, que pode ser utilizada por qualquer linguagem utilizada compatível com .NET.

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

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

Pode também instalar e utilizar o suporte de cliente Java EE de WebSphere Application Server Network Deployment, incluindo o cliente de serviços da web, o cliente EJB e o cliente JMS.

---

## Capítulo 2. Saiba mais sobre os conceitos chave

Utilize esta secção como ponto inicial para investigar as tecnologias utilizada no e pelo IBM Business Process Manager.

---

### Criar cenários

Utilize cenários para compreender e trabalhar com componentes e produtos da família de gestão de processos de negócio.

---

### Criação de versões

O ciclo de vida de uma process application começa com a criação da process application e continua através de um ciclo de actualização, implementação, co-implementação, anulação da implementação e arquivo da process application. O *Arquivo de versões* é um mecanismo utilizado para gerir o ciclo de vida da process application através da identificação, de forma exclusiva, das versões exclusivas da process application.

O modo como o arquivo de versões funciona no IBM Business Process Manager depende daquilo que se implementar numa Process Application, implementada a partir do repositório em IBM Process Center, ou uma aplicação empresarial directamente implementada a partir do IBM Integration Designer.

As Process Applications e Toolkits que implementar num ambiente em tempo de execução a partir do Process Center são, por predefinição, alvo de arquivo de versões. Para aplicações empresariais, pode seleccionar arquivar as versões de módulos e bibliotecas em IBM Integration Designer.

Em acréscimo, pode criar versões de uma tarefa humana ou máquina de estado, de modo a que possam coexistir versões múltiplas da tarefa ou máquina de estado no ambiente em tempo de execução.

### Criação de versões das Process Applications

A criação de versões fornece a capacidade do ambiente em tempo de execução de identificar Snapshots no ciclo de vida de uma Process Application e a ser capaz de executar de forma concorrente múltiplos snapshots num process server.

Para entender como as process applications com versões, é importante lembrar-se de que uma process application é um contentor que retém vários artefactos utilizados em ou por process application (por exemplo, modelos de processos, serviços, pistas, modelos de supervisão e referências de toolkit). Qualquer arquivo de versões é feito neste nível de contentor, não ao nível dos artefactos individuais. Para process applications, isto significa que o arquivo de versões acontece quando tira um snapshot.

Pode comparar snapshots para determinar diferenças entre as versões. Por exemplo, se um programador corrigiu um problema com um serviço e tirou, nessa altura, um Snapshot da sua Process Application ou Toolkit e, depois, outro programador fez várias alterações ao mesmo serviço e tirou um novo Snapshot, o gestor de projecto pode comparar os dois Snapshots para determinar que alterações foram feitas e por quem foram feitas. Se o gestor de projecto tiver decidido que as alterações adicionais ao serviço não foram satisfatórias, este pode voltar ao Snapshot da correcção original.

Pode executar diferentes versões (snapshots) de uma process application concorrentemente num servidor; quando instala um novo snapshot, ou para remover o original ou deixá-lo a executar.

## Contexto da versão

Cada snapshot tem metadados únicos para identificar a versão (referida como contexto de versão). Atribua esse identificador, mas a IBM recomenda a utilização de um sistema de versão numérica de três dígitos no formato <major>.<minor>.<service>. Consulte os tópicos sobre as convenções de nomenclatura para uma descrição mais detalhada deste esquema de criação de versões.

O IBM Business Process Manager atribui um espaço de nomes global para cada Process Application . O espaço de nomes global é, especificamente, a sugestão da Process Application ou um Snapshot em particular da Process Application. O nome da versão utilizado pelo servidor não pode ser maior do que sete caracteres, para que o nome atribuído seja um acrónimo que utiliza caracteres do nome atribuído ao Snapshot. Os acrónimos dos Snapshot são idênticos aos seus nomes, se estes estiverem de acordo com o estilo IBM VRM e não tiverem mais de sete caracteres. Por exemplo, um nome do Snapshot de 1.0.0 terá um acrónimo de 1.0.0, e um nome de Snapshot de 10.3.0 terá o acrónimo de 10.3.0. O acrónimo do Snapshot terá a garantia de ser único dentro do contexto da Process Application dentro do âmbito do Process Center Server. Por essa razão, o utilizador não pode editar o acrónimo do Snapshot.

## Considerações de arquivo de versões para o process applications em múltiplos conjuntos

Pode instalar a mesma versão de uma Process Application em vários conjuntos dentro da mesma célula. Para diferenciar entre estas múltiplas instalações da mesma versão da Process Application, crie um Snapshot para cada instalação e inclua um Snapshot de uma única célula no nome do Snapshot (por exemplo, v1.0\_cell1\_1 e v1.0\_cell1\_2). Cada snapshot é uma versão de uma process application (de uma perspectiva de gestão pura do ciclo de vida), mas o conteúdo e a função são os mesmos.

Quando instala uma Process Application num conjunto, é executada uma sincronização automática dos nós.

## Considerações de arquivo de versões para toolkits Process Designer

Lembre-se de que esses snapshots da process application são tipicamente tirados quando está pronto para testar ou instalar. Snapshots de Toolkit, no entanto, são normalmente tirados quando esse toolkit está pronto a ser utilizado pela process applications. Depois, se quiser actualizar o Toolkit, deve tirar outro Snapshot da "sugestão" quando estiver pronto, e os proprietários das Process Applications e Toolkits podem decidir se querem passar para o novo Snapshot.

## Arquivo de versões de módulos e bibliotecas

Se um módulo ou biblioteca estiverem numa process application ou num toolkit, este obtém o ciclo de vida da process application ou do toolkit (versões, snapshots, controlos e outros). Os nomes do módulo e biblioteca devem ser únicos no âmbito de uma Process Application ou Toolkit.

Este tópico descreve o arquivo de versões de módulos e bibliotecas que são utilizadas com Process Application. Tenha em atenção, porém, que se implementar módulos directamente de IBM Integration Designer para o Process Server, pode continuar a seguir o procedimento de atribuir números de versão a módulos durante a implementação, como descrito em "Criar versões arquivadas de módulos e bibliotecas".

Um módulo ou biblioteca associado com IBM Process Center deve ter as respectivas bibliotecas dependentes na mesma Process Application ou num Toolkit dependente.

A tabela seguinte lista as selecções que pode efectuar no editor de dependência em IBM Integration Designer quando uma biblioteca é associada a uma Process Application ou Toolkit:

Tabela 36. Dependências para Módulo, Process App ou Toolkit e Bibliotecas globais

Âmbito da biblioteca	Descrição	Pode depender de . . .
Módulo	Existe uma cópia desta biblioteca no servidor para cada módulo o utilizar.	Uma biblioteca de âmbito de módulo pode depender de todos os tipos de bibliotecas.
Process App ou Toolkit	A biblioteca é partilhada entre todos os módulos no âmbito da Process Application ou Toolkit. Esta definição entra em vigor caso a implementação seja efectuada através do IBM Process Center. Se a implementação ocorrer fora do IBM Process Center, a biblioteca é copiada para cada módulo. <b>Nota:</b> As bibliotecas criadas na versão 8 do IBM Integration Designer têm um nível de partilha de <b>Process App or Toolkit (Process App ou Toolkit)</b> por predefinição.	Uma biblioteca de este tipo apenas pode depender de bibliotecas globais.
Global	A biblioteca é partilhada entre todos os módulos em execução.	Uma biblioteca global apenas pode depender de outras bibliotecas globais. <b>Nota:</b> Deve configurar uma biblioteca partilhada em WebSphere de modo a implementar a biblioteca global. Consulte “Dependências de módulo e biblioteca” para obter mais informações.

## Módulos e bibliotecas associados a Process Applications ou Toolkits

Não é necessário criar versões de módulos e bibliotecas associados a Process Applications ou Toolkits.

Não é necessário criar versões dos módulos e bibliotecas associados a uma Process Application ou um Toolkit. Na realidade, não é possível criar uma versão de um módulo ou de uma biblioteca associada a uma Process Application ou a um Toolkit no editor de dependências. Os módulos e as bibliotecas associados a uma Process Application ou um Toolkit utilizam Snapshots, uma função no Process Center, para atingir o mesmo resultado que uma versão.

As bibliotecas associadas a uma Process Application ou a um Toolkit não terão um número de versão necessário na secção Bibliotecas do editor de dependências porque não é necessária nenhuma versão.

## Convenções de nomenclatura

É utilizada uma convenção de nomenclatura para diferenciar as diferentes versões de uma process application à medida que percorre o ciclo de vida de actualização, implementação, co-implementação, não implementação e arquivo.

Esta secção proporciona-lhe as convenções que são utilizadas para identificar, exclusivamente, versões de uma Process Application.

Um *contexto de versão* é uma combinação de acrónimos que descrevem, exclusivamente, uma Process Application ou Toolkit. Cada tipo de acrónimo tem uma convenção de nomenclatura. O acrónimo é limitado a um comprimento máximo de sete caracteres do conjunto de caracteres [A-Z0-9\_], excepto para o acrónimo do Snapshot, que também pode incluir um ponto final.

- O acrónimo da Process Application é criado quando a Process Application é criada. Pode ter um comprimento máximo de sete caracteres.
- O acrónimo do Snapshot é automaticamente criado quando o Snapshot é criado. Pode ter um comprimento máximo de sete caracteres.

Se o nome do Snapshot cumprir os critérios de um acrónimo de Snapshot válido, o nome e o acrónimo do Snapshot serão o mesmo.

**Nota:** Quando utilizar a função de encaminhamento adequada à versão do componente do fluxo de mediação, atribua um nome ao seu Snapshot para que esteja em conformidade com o esquema da <versão>.<edição>.<modificação> (por exemplo, 1.0.0). Devido ao acrónimo do Snapshot ser limitado a sete caracteres, os valores dos dígitos são limitados a um máximo de cinco dígitos no total (cinco dígitos mais dois pontos finais). Assim, deve ser tido cuidado quando os campos dos dígitos são aumentados, pois tudo além dos primeiros sete caracteres é truncado.

Por exemplo, o nome do Snapshot 11.22.33 resulta num acrónimo de Snapshot 11.22.3.

- O acrónimo da sugestão é criado automaticamente a partir do primeiro carácter de cada palavra do nome da sugestão. Por exemplo, uma nova sugestão criada com o nome **A minha nova sugestão** resultaria num valor de acrónimo de **AMNS**.

O nome e acrónimo predefinidos da sugestão são **Principal**. A implementação num servidor IBM Process Center inclui o acrónimo de sugestão no contexto de arquivo de versões caso o acrónimo da versão não seja **Principal**.

Uma definição de um processo empresarial numa Process Application é normalmente identificada pelo acrónimo do nome da Process Application, pelo acrónimo do Snapshot e pelo nome da definição do processo empresarial. Seleccione nomes exclusivos para as suas definições de processo empresarial, sempre que possível. Quando existirem nomes duplicados, poderá encontrar os seguintes problemas:

- Poderá ser incapaz de expor as definições do processo empresarial como serviços da Web sem algum tipo de mediação.
- Também poderá ser incapaz de chamar uma definição de processo empresarial criada no IBM Process Designer a partir de um processo BPEL criado no IBM Integration Designer.

O contexto da versão varia consoante o modo como a Process Application é implementada.

## Convenções de nomenclatura para implementações do servidor do Process Center

No servidor IBM Process Center, pode implementar um Snapshot de uma Process Application e um Snapshot de um Toolkit. Adicionalmente, pode implementar a sugestão de uma Process Application ou a sugestão de um Toolkit. (Uma *sugestão* é a versão funcional actual da sua Process Application ou Toolkit.) O contexto da versão varia, consoante o tipo de implementação.

Para Process Application, a sugestão da Process Application ou o Snapshot específico da Process Application é usado para identificar, exclusivamente, a versão.

Os toolkits podem ser implementados com uma ou mais aplicações, mas o ciclo de vida de cada toolkit está ligado ao ciclo de vida da process application. Cada Process Application tem a sua própria cópia do Toolkit ou Toolkits dependentes implementados no servidor. Um Toolkit implementado não é partilhado entre Process Application.

Se a sugestão associada à Process Application for denominada algo que não a predefinição de **Principal**, o acrónimo da sugestão também faz parte do contexto da versão.

Para obter mais informações, consulte a secção “Exemplos” na página 33, mais à frente neste tópico.

## Snapshots da Process Application

Para implementações do Snapshot da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo da sugestão da Process Application (se for utilizada outra sugestão que não **Principal**)
- Acrónimo do snapshot da Process Application

## Toolkits autónomos

Para implementações de Snapshot de Toolkit, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome de Toolkit
- Acrónimo da sugestão de Toolkit (se for utilizada outra sugestão que não **Principal**)
- Acrónimo do Snapshot do Toolkit

## Sugestões

Sugestões da Process Application utilizadas durante os testes iterativos no Process Designer. Apenas podem ser implementadas nos servidores do Process Center.

Para implementações da sugestão da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo da sugestão da Process Application (se for utilizada outra sugestão que não **Principal**)
- "Sugestão"

As sugestões de Toolkit também são utilizadas para testes iterativos no Process Designer. Não são implementadas num servidor de produção.

Para implementações da sugestão do Toolkit, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome de Toolkit
- Acrónimo da sugestão de Toolkit (se for utilizada outra sugestão que não **Principal**)
- "Sugestão"

## Exemplos

Os recursos devem ser exclusivamente designados e identificados externamente utilizando o contexto da versão.

- A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados. Neste exemplo, uma sugestão da Process Application utiliza o nome predefinido da sugestão (**Principal**):

*Tabela 37. Sugestão da Process Application com nome predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Principal</b>
Acrónimo da sugestão da Process Application	"" (quando a pista é <b>Principal</b> )
Snapshot da Process Application	
Acrónimo do snapshot da Process Application	<b>Sugestão</b>

Quaisquer módulos SCA associados a esta sugestão da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 38. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-Sugestão-M1	PA1-Sugestão-M1.ear
M2	PA1-Sugestão-M2	PA1-Sugestão-M2.ear



- A tabela seguinte mostra um exemplo de uma sugestão de Process Application que utiliza um nome de sugestão não predefinido:

*Tabela 39. Sugestão da Process Application com nome não predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Sugestão1</b>
Acrónimo da sugestão da Process Application	<b>T1</b>
Snapshot da Process Application	
Acrónimo do snapshot da Process Application	<b>Sugestão</b>

Quaisquer módulos SCA associados a esta sugestão da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 40. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-T1-Sugestão-M1	PA1-T1-Sugestão-M1.ear
M2	PA1-T1-Sugestão-M2	PA1-T1-Sugestão-M2.ear

As convenções de nomenclatura semelhantes aplicam-se às implementações de sugestões de Toolkit e de Snapshots avançadas. Estas também podem ser aplicadas em Snapshots avançados instaladas no Process Server.

- A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados. Neste exemplo, um snapshot da Process Application utiliza o nome predefinido da sugestão (**Principal**):

*Tabela 41. Snapshot da Process Application com nome predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Principal</b>
Acrónimo da sugestão da Process Application	"" (quando a pista é <b>Principal</b> )
Snapshot da Process Application	<b>Process Shapshot V1</b>
Acrónimo do snapshot da Process Application	<b>PSV1</b>

Quaisquer módulos SCA associados a este snapshot da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

*Tabela 42. Módulos SCA e ficheiros EAR adequados à versão*

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- A tabela seguinte mostra um exemplo de um snapshot de Process Application que utiliza um nome de sugestão não predefinido:

*Tabela 43. Snapshot da Process Application com nome não predefinido de sugestão*

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>

Tabela 43. Snapshot da Process Application com nome não predefinido de sugestão (continuação)

Tipo de nome	Exemplo
Acrónimo do nome da Process Application	<b>PA1</b>
Sugestão da Process Application	<b>Sugestão1</b>
Acrónimo da sugestão da Process Application	<b>T1</b>
Snapshot da Process Application	<b>Process Snapshot V1</b>
Acrónimo do snapshot da Process Application	<b>PSV1</b>

Quaisquer módulos SCA associados a este snapshot da Process Application incluem o contexto da versão, como mostrado na tabela seguinte:

Tabela 44. Módulos SCA e ficheiros EAR adequados à versão

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

## Convenções de nomenclatura para implementações do Process Server

No Process Server, pode implementar o Snapshot de uma Process Application. O acrónimo do Snapshot da Process Application é utilizado para identificar, exclusivamente, a versão.

Para implementações do Snapshot da Process Application, o contexto da versão é uma combinação dos seguintes itens:

- Acrónimo do nome da Process Application
- Acrónimo do Snapshot da Process Application

Os recursos devem ser exclusivamente designados e identificados externamente utilizando o contexto da versão. A tabela seguinte mostra um exemplo de nomes que são exclusivamente identificados:

Tabela 45. Exemplo de nomes e acrónimos

Tipo de nome	Exemplo
Nome de Process Application	<b>Process Application 1</b>
Acrónimo do nome da Process Application	<b>PA1</b>
Snapshot da Process Application	<b>1.0.0</b>
Acrónimo do Snapshot da Process Application	<b>1.0.0</b>

Um recurso, como um módulo ou biblioteca, tem o contexto da versão como parte da sua identidade.

A tabela seguinte mostra um exemplo de dois módulos e de como os ficheiros EAR associados incluem o contexto da versão:

Tabela 46. Módulos SCA e ficheiros EAR adequados à versão

Nome do módulo SCA	Nome adequado à versão	Nome da aplicação/EAR adequado à versão
M1	PA1-1.0.0-M1	PA1-1.0.0-M1.ear
M2	PA1-1.0.0-M2	PA1-1.0.0-M2.ear

A tabela seguinte mostra um exemplo de duas bibliotecas de âmbito da Process Application e de como os ficheiros JAR associados incluem o contexto da versão:

Tabela 47. Bibliotecas de âmbito do processo e ficheiros JAR adequados à versão

Nome da biblioteca de âmbito da Process Application SCA	Nome adequado à versão	Nome JAR adequado à versão
Lib1	PA1-1.0.0-Lib1	PA1-1.0.0-Lib1.jar
Lib2	PA1-1.0.0-Lib2	PA1-1.0.0-Lib2.jar

## Associações adequadas à versão

As Process Applications podem conter módulos SCA que incluam associações de importação e de exportação. Quando co-implementar as aplicações, a associação para cada versão da aplicação deve ser única. Algumas associações são automaticamente actualizadas durante a implementação para garantir a exclusividade entre versões. Noutros casos, tem de actualizar a associação após a implementação para garantir a respectiva exclusividade.

Uma associação *adequada à versão* é agrupada numa versão específica de uma Process Application, o que garante a respectiva exclusividade entre Process Applications. As seguintes secções descrevem as associações que são automaticamente actualizadas para serem adequadas à versão e quaisquer acções que necessite de tomar, no tempo de execução, quando uma associação não for adequada à versão. Para mais informações sobre questões a ter em conta quando cria módulos, consulte “Considerações ao utilizar associações”.

### SCA

O nome do destino de uma associação SCA é automaticamente alterado para ser adequado à versão durante a implementação, caso as associações de importação e de exportação do módulo sejam definidas no mesmo âmbito da Process Application.

Se as associações não forem definidas no mesmo âmbito da Process Application, é registada uma mensagem informativa. Deve modificar a associação de importação após a implementação para alterar o endereço do destino final. Pode utilizar a consola administrativa para alterar o endereço do destino final.

### Serviço da Web (JAX-WS ou JAX-RPC)

O nome do endereço do destino final de uma associação de serviço da Web é automaticamente alterado para ser adequado à versão, durante a implementação, caso todas as seguintes condições forem verdadeiras:

- Seguiu a convenção de nomenclatura predefinida para o endereço:  
**http://ip:port/ModuleNameWeb/sca/ExportName**
- O endereço do destino final é SOAP/HTTP.
- As associações de importação e exportação do módulo são definidas no mesmo âmbito da Process Application.

Se estas condições não forem verdadeiras, é registada uma mensagem informativa. A acção que adoptar, de seguida, dependerá do modo como estiver a implementar a sua Process Application:

- Se estiver a co-implementar a sua Process Application, deve alterar o nome, manualmente, ao URL de destino final de SOAP/HTTP ou à fila de destino de SOAP/JMS, para que seja único entre as versões da Process Application. Pode utilizar a consola administrativa após a implementação para alterar o endereço do destino final.
- Se apenas estiver a implementar uma versão única da Process Application, pode ignorar esta mensagem

Para a co-implementação do Snapshot da associação do serviço da Web de SOAP/ JMS, a acção que tomar depende de como estiver a implementar a sua Process Application:

- Se a importação e exportação do destino estiverem na mesma Process Application, execute os seguintes passos antes de publicar a Process Application para Process Center e crie o Snapshot:
  1. Altere o URL do destino final da exportação. Certifique-se de que o destino e o fornecedor de ligação são únicos.
  2. Altere o URL do destino final da importação para que seja igual ao que especificou para a exportação no passo anterior.
- Se a importação e exportação do destino estiverem em diferentes Process Applications, execute os seguintes passos:
  1. Altere o URL do destino final da exportação. Certifique-se de que o destino e o fornecedor de ligação são únicos.
  2. Publique a Process Application para Process Center.
  3. Crie o Snapshot.
  4. Implemente a Process Application para o Process Server.
  5. Utilize a consola administrativa da WebSphere para alterar o URL do destino final da importação correspondente para que seja igual ao que especificou para a exportação.

## HTTP

O nome do endereço do URL do destino final de uma associação HTTP é automaticamente alterado para ser adequado à versão, durante a implementação, caso todas as seguintes condições forem verdadeiras:

- Seguiu a convenção de nomenclatura predefinida para o endereço:  
**`http(s)://ip:port/ModuleNameWeb/contextPathinExport`**
- As associações de importação e exportação do módulo são definidas no mesmo âmbito da Process Application.

Se estas condições não forem verdadeiras, é registada uma mensagem informativa. A acção que adoptar, de seguida, dependerá do modo como estiver a implementar a sua Process Application:

- Se estiver a co-implementar a sua Process Application, deve alterar o nome, manualmente, ao URL de destino final para que seja único entre as versões da Process Application. Pode utilizar a consola administrativa após a implementação para alterar o endereço do destino final.
- Se apenas estiver a implementar uma versão única da Process Application, pode ignorar esta mensagem

## JMS e JMS genérico

Os JMS gerados pelo sistema e as associações de JMS genéricos são automaticamente adequados à versão.

**Nota:** Para JMS definidos pelo utilizador e associações de JMS genéricos, não ocorre qualquer alteração automática do nome durante a implementação para permitir às associações serem adequadas à versão. Se a associação for definida pelo utilizador, deve alterar o nome aos seguintes atributos, para que sejam únicos entre as versões da Process Application:

- Configuração do destino final
- Fila de destino de recepção
- Nome da porta de recepção (se definido)

Defina o destino de Enviar correspondente se alterar o destino final do módulo de destino.

## MQ/JMS e MQ

Não ocorre nenhuma alteração automática do nome durante a implementação para permitir às associações de tipo MQ/JMS ou MQ serem adequadas à versão.

Deve alterar o nome aos seguintes atributos, para que sejam únicos entre as versões da Process Application:

- Configuração do destino final
- Fila de destino de recepção

Defina o destino de Enviar correspondente se alterar o destino final do módulo de destino.

## EJB

Não ocorre nenhuma alteração automática do nome durante a implementação para permitir às associações de tipo EJB serem adequadas à versão.

Deve alterar o nome do atributo JNDI para que seja único entre as versões da Process Application.

Tenha em atenção que as aplicações do cliente também necessitam de ser actualizadas para utilizarem os novos nomes JDNI.

## EIS

O nome de um adaptador de recurso é automaticamente alterado para ser adequado à versão, durante a implementação, desde que o nome de recurso predefinido (*ModuleNameApp:Adapter Description*) não tenha sido modificado.

Se o nome de recurso predefinido tiver sido modificado, os nomes do adaptador de recurso devem ser únicos entre as versões da Process Application.

Se os nomes do adaptador de recurso não forem únicos, é registada uma mensagem informativa durante a implementação, para o alertar. Pode alterar, manualmente, os nomes dos adaptadores de recurso, após a implementação, utilizando a consola administrativa.

## Chamada para execução dinâmica adequada à versão

Pode configurar os componentes do fluxo de mediação para que encaminhem mensagens para os destinos finais que estejam dinamicamente determinados no tempo de execução. Quando criar o módulo de mediação, configura a procura do destino final para utilizar o encaminhamento adequado à versão.

Se utilizar o estilo IBM\_VRM (*<version>.<release>.<modification>*) para o Snapshot, pode exportar o ficheiro EAR da Process Application para WebSphere Service Registry and Repository (WSRR). Quando cria o módulo de mediação, configura, de seguida, a procura do destino final para utilizar o encaminhamento adequado à versão. Por exemplo, seleccione **Return endpoint matching latest compatible version of SCA module-based services (Repor destino final para corresponder à versão mais recente compatível dos serviços com base em módulo SCA)** a partir do **Match Policy field (campo Política de correspondência)** e selecciona **SCA** para **Binding Type (Tipo de associação)**.

As versões futuras da Process Application são implementadas no servidor e publicadas para WSRR e a procura do destino final do módulo de mediação chama, dinamicamente, a versão compatível mais recente do destino final de serviço.

Tenha em atenção que, como alternativa, pode definir o destino em SMOHeader e o valor pode ser transportado pela mensagem de pedido.

## Implementar Process Application com projectos e módulos Java

As Process Application podem conter módulos Java EE e projectos de Java personalizados. Quando co-implementar as aplicações, o módulo Java EE para cada versão da aplicação deve ser único.

Tenha em atenção que os módulos Java EE e os projectos de Java personalizados são implementados num servidor caso sejam implementados com um módulo SCA que tenha uma dependência declarada sobre estes. Se não seleccionar **Deploy with module (Implementar com módulo)** (que é a predefinição) quando declarar a dependência, necessita de implementar o módulo ou projecto manualmente.

## Implementar Process Applications com regras empresariais e selectores

Se estiver a implementar versões múltiplas de uma Process Application que inclua uma regra de negócio ou componente de selector, tenha em atenção ao modo como os metadados são utilizados pelas versões.

Os metadados dinâmicos para uma regra de negócio ou componente de selector são definidos no tempo de execução pelo nome do componente, espaço de nomes de destino do componente e tipo de componente. Se duas ou mais versões de uma Process Application que contenham uma regra de negócio ou selector forem implementadas para o mesmo ambiente em tempo de execução, partilharão a mesma lógica de regra (regra de negócio) ou metadados de encaminhamento (selector).

Para permitir a utilização, por parte de cada versão da regra de negócio ou componente de selector da Process Application, dos respectivos metadados dinâmicos (lógica de regra ou encaminhamento), reconstrua o espaço de nomes de destino para que seja único para cada versão da Process Application.

---

## Objectos de configuração

Pode utilizar os comandos AdminConfig (wsadmin) na ferramenta de administração da linha de comandos do WebSphere (wsadmin) para aceder e modificar propriedades de bases de dados e segurança no IBM Business Process Manager.

O termo *configuration object* refere um objecto que é acessido através dos comandos AdminConfig wsadmin. Para obter mais informações, consulte Security configuration properties.

---

## Arquitectura de implementação

A arquitectura de implementação do IBM Business Process Manager consiste em processos de software denominados servidores, unidades topológicas referenciadas como nós e células e o repositório de configuração utilizado para armazenar informações de configuração.

### Células

No IBM Business Process Manager, as *células* são agrupamentos lógicos de um ou mais nós numa rede distribuída.

Uma célula é um conceito de configuração, uma forma para os administradores associarem logicamente nós, uns com os outros. Os administradores definem os nós que formam uma célula, de acordo com o critério específico indicado para cada ambiente organizacional.

Os dados de configuração administrativa são guardados em ficheiros XML. Uma célula retém os ficheiros de configuração principais para cada um dos servidores em cada um dos nós da célula. Cada nó e servidor possuem também ficheiros de configuração local próprios. As alterações efectuadas a um nó local ou a um ficheiro de configuração do servidor, são temporárias, caso o servidor faça parte da célula. Enquanto estiverem em vigor, as alterações locais substituem as configurações das células. As alterações ao servidor principal e ficheiros de configuração de nó principais efectuadas ao nível da célula, substituem quaisquer alterações temporárias ao nó durante a sincronização dos documentos de configuração da célula com os nós. A sincronização durante eventos designados, tal como quando o servidor é iniciado.

## Servidores

Os servidores fornecem a funcionalidade nuclear do IBM Business Process Manager. Os Process Servers expandem ou aumentam capacidade de um servidor de aplicações para gerir módulos da Arquitectura de Componente de Serviço (SCA, Service Component Architecture). Outros servidores (gestores de implementação e agentes de nós), são utilizados para gerir Process Servers.

Um Process Server pode ser um *servidor autónomo* ou um *servidor gerido*. Um servidor gerido pode opcionalmente ser membro de um *conjunto*. Um conjunto de servidores geridos, conjunto de unidades de servidores e outros middleware é denominado um *ambiente de implementação*. Num ambiente de implementação, cada um dos servidores geridos ou conjuntos de unidades é configurado para uma função específica no ambiente de implementação (por exemplo, sistema central de destino, sistema central do módulo de aplicações ou servidor Common Event Infrastructure). Um servidor autónomo é configurado para fornecer todas as funções requeridas.

Os servidores fornecem o ambiente em tempo real para módulos SCA, para os recursos que são utilizados por esses módulos (origens de dados, especificações de activação e destinos JMS) e para recursos fornecidos pela IBM (destinos de mensagens, contentores Business Process Choreographer e servidores Common Event Infrastructure).

Um *agente do nó* é um agente administrativo que representa um nó para o sistema e gere os servidores nesse nó. Os agentes de nós supervisionam os servidores num sistema central e encaminham pedidos administrativos para os servidores. O agente do nó é criado quando um nó está federado a um gestor de implementação.

Um *gestor de implementação* é um agente administrativo que fornece uma vista de gestão centralizada para múltiplos servidores e conjuntos de unidades.

Um servidor autónomo é definido por um perfil autónomo; um gestor de implementação é definido por um perfil de gestor de implementação; os servidores geridos são criados num *nó gerido*, que é definido por um perfil de nós geridos.

### Servidores autónomos

Um servidor autónomo fornece um ambiente para implementação de módulos SCA num processo de servidor. Este processo de servidor inclui, mas não se limita a, uma consola administrativa, um destino de implementação, um suporte de serviço de mensagem, o gestor de regras de processo empresarial e um servidor Common Event Infrastructure (infraestrutura de evento comum).

Um servidor autónomo é simples de configurar e apresenta uma consola Quick Start a partir da qual poderá iniciar e parar o servidor, abrir a galeria de exemplos e a consola administrativa. Se instalar os exemplos do IBM Business Process Manager e, em seguida, abrir a galeria de exemplos, é aplicada uma solução de exemplo no servidor autónomo. É possível explorar os recursos utilizados para este exemplo na consola administrativa.

É possível implementar soluções próprias a um servidor autónomo, mas um servidor autónomo não pode fornecer a capacidade, escalabilidade ou robustez requerida num ambiente de produção. É melhor utilizar um ambiente de implementação de rede para o ambiente de produção.

É possível começar com um servidor autónomo e posteriormente incluí-lo num ambiente de implementação de rede, federando-o à célula do gestor de implementação, *desde que não tenham sido federados quaisquer outros nós a essa célula*. Não é possível federar múltiplos servidores autónomos numa célula. Para federar o servidor autónomo, utilize a consola administrativa do gestor de implementação ou o comando **addNode**. O servidor autónomo não deverá estar em execução ao federá-lo utilizando o comando **addNode**.

Um servidor autónomo é definido por um perfil do servidor autónomo.



## Conjuntos

Os conjuntos de unidades são grupos de servidores geridos em conjunto e participam na gestão da carga de trabalho.

Um conjunto pode conter nós ou servidores de aplicações individuais. Um nó é normalmente um sistema informático físico com um endereço de IP de sistema central distinto que executa um ou mais servidores de aplicações. Os conjuntos de unidades podem ser agrupados sob a configuração de uma célula, que associa logicamente muitos servidores e conjuntos de unidades com configurações e aplicações diferentes entre eles, dependendo do critério do administrador e dos requisitos de cada ambiente organizacional.

Os conjuntos são responsáveis pelo equilíbrio da carga de trabalho entre os servidores. Os servidores que fazem parte de um conjunto são denominados membros do conjunto. Durante a instalação de uma aplicação num conjunto de unidades, a aplicação é automaticamente instalada em cada um dos membros do conjunto de unidades.

Como cada membro de conjunto de unidades contém as mesmas aplicações, é possível distribuir tarefas cliente de acordo com as capacidades de cada máquina atribuindo cargas a cada um dos servidores.

A atribuição de cargas aos servidores num conjunto melhora o desempenho e redireccionamento em caso de falha. As tarefas são atribuídas aos servidores que possuem a capacidade para executar as operações de tarefas. Se um dos servidores não estiver disponível para executar a tarefa, esta é atribuída a outro membro do conjunto. Esta capacidade de reatribuição tem vantagens óbvias sobre a execução de um único servidor de aplicações que pode ficar sobrecarregado caso sejam efectuados demasiados pedidos.

## Perfis

Um perfil define um ambiente em tempo de execução único, com ficheiros de comando, ficheiros de configuração e ficheiros de registo separados. Os perfis definem três tipos diferentes de ambientes em sistemas IBM Business Process Manager: servidor autónomo, gestor de implementação e nó gerido.

Ao utilizar perfis, é possível ter mais do que um ambiente em tempo de execução no sistema, sem ter de instalar múltiplas cópias dos ficheiros binários do IBM Business Process Manager.

Utilize o Utilitários de linha de comandos BPMConfig para criar perfis do IBM BPM. O utilitário da linha de comandos **manageprofiles** ou a respectiva interface gráfica do utilizador, a Ferramenta de Gestão de Perfis (PMT, Profile Management Tool), pode ser utilizada como uma forma alternativa para criar um gestor de implementação ou os perfis de nós geridos. O PMT já não é suportado para a criação de perfis autónomos.

**Nota:** nas plataformas distribuídas, cada perfil tem um nome único. Em z/OS, todos os perfis são nomeados como “predefinidos”; não pode mudar o nome, editar, copiar ou eliminar perfis em z/OS.

## Tipos de perfis

Os seguintes tipos de perfis do IBM BPM estão disponíveis com o IBM Business Process Manager V8.5:

### Perfil autónomo do IBM BPM

O perfil autónomo define os servidores autónomos com capacidades e funcionalidades específicas para as configurações do IBM BPM Express. Pode criar o perfil autónomo através do modelo do perfil BPM/BpmServer, que só é suportado no IBM BPM Express.

### Perfil do gestor de implementação do IBM BPM

O perfil do gestor de implementação define um gestor de implementação que fornece uma interface administrativa para um grupo lógico de servidores numa ou mais estações. Pode criar o perfil do gestor de implementação através do modelo do perfil BPM/BpmDmgr, que só é suportado no IBM BPM Standard e no IBM BPM Advanced.

## Perfil do nó gerido do IBM BPM

O perfil do nó gerido define um nó gerido quando o nó é federado para um gestor de implementação. Pode criar um perfil de nó gerido através do modelo de perfil BPM/BpmNode, que só é suportado no IBM BPM Standard e no IBM BPM Advanced.

Os tipos de perfil específicos estão disponíveis com cada configuração do IBM BPM.

Tabela 48. Tipos de perfis disponíveis do IBM BPM

Configuração do IBM BPM	Tipos de perfis		
	Autónomas	Gestor de implementação	Nó gerido
IBM BPMEExpress	Sim	Não	Não
IBM BPM Standard	Não	Sim	Sim
IBM BPM Advanced	Não	Sim	Sim
Ambiente de Teste de Unidade (UTE, Unit Test Environment) para o IBM Integration Designer	Sim	Opcional	Opcional

## O directório do perfil

Todos os perfis no sistema possuem um directório próprio que contém todos os ficheiros correspondentes. A localização do directório do perfil é especificada durante a criação do perfil. Por predefinição, é no directório `profiles` que o IBM Business Process Manager é instalado. Por exemplo, o perfil `Dmgr` está localizado em `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

## O perfil predefinido

O primeiro perfil criado numa instalação do IBM Business Process Manager é o *perfil predefinido*. O perfil predefinido é o destino predefinido para comandos emitidos a partir do directório `bin` no directório onde foi instalado o IBM Business Process Manager. Se existir apenas um perfil no sistema, todos os comandos operam nesse perfil. Se for criado outro perfil, poderá torná-lo predefinido.

**Nota:** O perfil predefinido não é necessariamente um perfil cujo nome é "default".

## Aumentar perfis

Se já tiver criado um perfil de gestor de implementação, um perfil de nós geridos ou um perfil de servidor autónomo para o WebSphere Application Server Network Deployment, pode *aumentá-lo* para suportar o IBM Business Process Manager para além da função existente. Para aumentar um perfil, é necessário instalar primeiro o IBM Business Process Manager. Em seguida, utilize o utilitário da linha de comandos **manageprofiles** para aumentar um perfil autónomo ou utilize a Ferramenta de Gestão de Perfis (PMT, Profile Management Tool) ou o utilitário da linha de comandos **manageprofiles** para aumentar um gestor de implementação ou um perfil de nós geridos.

**Importante:** Num ambiente de implementação da rede, tem de aumentar primeiro o perfil do gestor de implementação e, em seguida aumentar os perfis de nós geridos.

**Restrição:** Não pode aumentar um perfil de gestor de implementação ou autónomo onde o registo de utilizadores VMM da WebSphere predefinido foi alterado, por exemplo, para utilizar LDAP.

## Gestores de implementação

Um gestor de implementação é um servidor que gere operações para um grupo lógico ou célula de outros servidores. O gestor de implementação é a localização central para administrar os servidores e conjuntos.

Quando é criado um ambiente de implementação, o perfil do agente de implementação é o primeiro perfil a ser criado. Cada ambiente de implementação que cria tem uma consola de Quick Start, a partir da qual poderá iniciar e parar o gestor de implementação e iniciar a respectiva consola administrativa. Pode utilizar a consola administrativa do gestor de implementação para gerir os servidores e conjuntos de unidades da célula. Isto inclui configurar servidores e conjuntos de unidades, adicionar servidores a conjuntos de unidades, iniciar a parar servidores e conjuntos de unidades e, implementar módulos SCA.

Embora o gestor de implementação seja um tipo de servidor, não é possível implementar módulos no próprio gestor de implementação.

## Nós

Um *nó* é um agrupamento lógico de servidores geridos.

Um nó corresponde normalmente a um sistema informático lógico ou físico com um endereço de IP do sistema central distinto. Os nós não podem abranger múltiplos computadores. Os nomes dos nós são normalmente idênticos ao nome do sistema central do computador.

Os nós da topologia de implementação de rede podem ser geridos ou não geridos. Um nó gerido possui um processo de agente do nó que gera a configuração e os servidores. Os nós não geridos não possuem um agente do nó.

### Nós geridos

Um *nó gerido* é um nó federado a um gestor de implementação, contém um agente de nó e pode conter servidores geridos. Num nó gerido, é possível configurar e executar servidores geridos.

Os servidores configurados num nó gerido constituem os recursos do ambiente de implementação. Estes servidores são criados, configurados, iniciados, interrompidos, geridos e eliminados utilizando a consola administrativa do gestor de implementação.

Um nó gerido possui um agente de nó que gere todos os servidores de um nó.

Quando um nó é federado, é criado automaticamente um processo de agente do nó. Este agente do nó deverá estar em execução para ser possível gerir a configuração do perfil. Por exemplo, quando são executadas as tarefas seguintes:

- Iniciar e parar processos no servidor.
- Sincronizar dados de configuração no gestor de implementação com a cópia no nó.

No entanto, o agente do nó não necessita de estar em execução para que as aplicações possam ser executadas ou para configurar recursos no nó.

Um nó gerido poderá conter um ou mais servidores, geridos por um gestor de implementação. É possível implementar soluções nos servidores de um nó gerido, mas o nó gerido não possui uma galeria de aplicações de exemplo. O nó gerido é definido por um perfil de nó gerido e possui uma consola Quick Start.

### Nós não geridos

Um nó não gerido não possui um agente do nó para gerir os servidores.

Nós não gerido na topologia Implementação de rede pode possuir definições de servidor tais como servidores da Web, mas não definições de application server. Os nós não geridos nunca podem ser

federados. Isto é, um agente do nó nunca pode ser adicionado a um nó não gerido. Um servidor autónomo é outro tipo de nó não gerido. O gestor de implementação não pode gerir este servidor autónomo porque é desconhecido para a célula. Um servidor autónomo pode ser federado. Quando é federado, é criado automaticamente um agente do nó. O nó torna-se um nó gerido na célula.

## Agentes de nós

Os agentes de nós são agentes administrativos que encaminham pedidos administrativos para os servidores.

Um agente do nó é um servidor que é executado em todos os sistemas centrais que participa na configuração da Implementação de Rede (Network Deployment). Trata-se de um agente puramente administrativo e não está envolvido nas funções das aplicações de servidor. Um agente do nó aloja também outras funções administrativas importantes, tais como serviços de transferência de ficheiros, sincronização da configuração e supervisão do desempenho.

## Considerações de nome para perfis, nós, servidores, sistemas centrais e células

Este tópico discute termos reservados e questões a considerar ao dar um nome a um perfil, nó, servidor, sistema central e célula (se tal for aplicável). Este tópico aplica-se a plataformas distribuídas.

### Considerações de nomes de perfil

O nome de perfil pode ser qualquer nome único com as seguintes restrições. Não utilizar qualquer dos seguintes caracteres ao dar um nome a um perfil:

- Espaços
- Caracteres especiais que não são permitidos no nome de um directório no sistema operativo, tal como \*, & ou ?.
- Barras (/) ou barras invertidas (\)

São permitidos caracteres de duplo byte.

**Windows** **Considerações sobre caminho do directório:** o caminho do directório de instalação deverá ser inferior ou igual a 60 caracteres. O número de caracteres do *caminho\_directório\_perfis\nome\_perfil* directório deverá ser inferior ou igual a 80 caracteres.

**Nota:** Utilize a convenção de nomenclatura de caminho abreviada quando criar um perfil num ambiente Windows para evitar a limitação do comprimento do caminho de 255 caracteres do Windows.

### Considerações de nomes de nó, servidor, sistema central e célula

**Nomes reservados:** evitar utilizar nomes reservados como valores de campos. A utilização de nomes reservados pode provocar resultados imprevisíveis. As palavras seguintes são reservadas:

- células
- nós
- servidores
- conjuntos
- aplicações
- implementações

**Descrição dos campos nas páginas Nós e Sistemas Centrais e Nó, Sistema Central e Nomes de Células:** Utilize as directrizes de nomenclatura adequadas quando cria perfis.

- Perfis do servidor autónomo

- Perfis do gestor de implementação
- Perfis do nó gerido

Tabela 49. directrizes de nomenclatura para perfis de servidores autónomos

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i> Nó <i>NodeNumber</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li> </ul>	Evitar a utilização dos nomes reservados.	Seleccionar qualquer nome pretendido. Se for planeada a criação de mais do que um servidor no sistema, deverá ser utilizado um nome único para ajudar a organizar a instalação.
Nome do servidor	<p>Linux      UNIX</p> <p>Windows      servidor1</p>	Utilize um nome único para o servidor.	Nome lógico para o servidor.
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows      O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede.	Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.

Tabela 50. directrizes de nomenclatura para perfis de gestores de implementação

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i> Gestor de Célula <i>Node Number</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li> </ul>	Utilize um nome único para o gestor de implementação. Evitar a utilização dos nomes reservados.	O nome é utilizado para administração na célula do gestor de implementação.
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows      O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede. Evitar a utilização dos nomes reservados.	Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.

Tabela 50. directrizes de nomenclatura para perfis de gestores de implementação (continuação)

Nome do campo	Valor predefinido	Restrições	Descrição
Nome da célula	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i></p> <p>Célula <i>CellNumber</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>CellNumber</i> é um número sequencial com início em 01.</li> </ul>	<p>Utilize um nome único para a célula do gestor de implementação. O nome da célula deverá ser único em todas as ocorrências em que o produto está em execução na mesma estação de trabalho física ou conjunto de unidades de estações de trabalho, tal como o Sysplex. Além disso, um nome de célula deverá ser único em qualquer circunstâncias em que seja necessária conectividade de rede entre entidades, seja entre as células ou a partir de um cliente que tenha de comunicar com cada uma das células. Os nomes das células também têm de ser únicos se os espaços de nomes correspondentes vão ser federados. Caso contrário, podem ocorrer sintomas de excepção, tal como o <code>javax.naming.NameNotFoundException</code>, em que será necessário criar células com nomes exclusivos.</p>	<p>Todos os nós federados tornam-se membros da célula do gestor de implementação, à qual foi dado o nome na página Nó, Sistema Central e Nomes de Célula da Ferramenta de Gestão de Perfis (Profile Management Tool).</p>

Tabela 51. Directrizes de nomenclatura para perfis de nós geridos

Nome do campo	Valor predefinido	Restrições	Descrição
Nome do nó	<p>Linux      UNIX</p> <p>Windows      <i>shortHostName</i></p> <p>Nó <i>NodeNumber</i> onde:</p> <ul style="list-style-type: none"> <li>• <i>shortHost Name</i> é o nome abreviado do sistema central.</li> <li>• <i>NodeNumber</i> é um número sequencial com início em 01.</li> </ul>	<p>Evitar a utilização dos nomes reservados.</p> <p>Utilize um nome único na célula do gestor de implementação.</p>	<p>O nome é utilizado para administração na célula do gestor de implementação à qual é adicionado o perfil do nó gerido. Utilize um nome único na célula do gestor de implementação.</p>
Nome do sistema central	<p>Linux      UNIX</p> <p>Windows      O formato longo do nome do servidor de nomes do domínio (DNS, domain name server).</p>	<p>Utilize um nome do sistema central totalmente qualificado que é endereçável através da rede.</p>	<p>Utilize o nome do DNS actual ou o endereço de IP da estação de trabalho para activar as comunicações. Consulte as informações adicionais sobre o nome do sistema central na tabela seguinte.</p>

### Considerações sobre o nome do sistema central:

O nome do sistema central é o nome da rede da estação de trabalho física em que o nó está instalado. O nome do sistema central deverá ser resolvido para o nó de uma rede física no servidor. Quando existem múltiplos cartões de rede no servidor, o nome do sistema central ou o endereço de IP deverá ser resolvido para um dos cartões de rede. Os nós remotos utilizam o nome do sistema central para ligar e para comunicar com este nó.

IBM Business Process Manager é compatível com os Protocolos da Internet versão 4 (IPv4) e versão 6 (IPv6). Poderá utilizar qualquer dos formatos sempre que for necessário introduzir endereços de IP na consola administrativa ou em outro local. Observe que, se o IPv6 estiver implementado no sistema será necessário introduzir o endereço de IP no formato IPv6 e reciprocamente, se o IPv6 ainda não estiver disponível, deverá introduzir o endereço IP no formato IPv4. Para mais informações sobre IPv6 consulte a seguinte descrição: IPv6.

As directrizes seguintes podem ajudar a determinar o nome do sistema central apropriado para a estação de trabalho:

- Selecione um nome do sistema central que possam ser contactado por outras estações de trabalho na rede.
- Não utilizar o identificador genérico, sistema central local (localhost), para este valor.
- Não proceder à instalação de IBM Business Process Manager produtos num servidor com um nome do sistema central que utilize caracteres do conjunto de caracteres de duplo byte (DBCS). A utilização de caracteres DBCS no nome do sistema central, não é suportada.
- Evite utilizar o carácter de sublinhado ( \_ ) nos nomes dos servidores. Os padrões da Internet ditam que os nomes dos domínios estão em conformidade com os requisitos de nome do sistema central descritos no Internet Official Protocol Standards RFC 952 e RFC 1123. Os nomes do domínio devem conter apenas letras (maiúsculas ou minúsculas) e dígitos. Os nomes dos domínios podem também conter caracteres de traço ( - ) desde que os traços não estejam no fim do nome. Os caracteres de sublinhado ( \_ ) não são suportados no nome do sistema central. Se tiver instalado IBM Business Process Manager num servidor com um carácter de sublinhado no nome do servidor, terá de aceder ao servidor através do respectivo IP até que seja mudado o nome.

Caso sejam definidos nós coexistindo no mesmo computador com endereços de IP únicos, deverá ser definido cada um dos endereços de IP numa tabela de procura num servidor de nomes de domínio (DNS). Os ficheiros de configuração para servidores não fornecem a resolução do nome do domínio para múltiplos endereços IP numa estação de trabalho com um único endereço de rede.

O valor especificado para o nome do sistema central é utilizado como valor da propriedade hostName nos documentos de configuração. Especifique o valor do nome do sistema central num dos formatos seguintes:

- cadeia de nome do sistema central dos servidores de nomes de domínio (DNS), como xmachine.manhattan.ibm.com
- A cadeia predefinida do nome do sistema central do DNS, tal como xmachine
- Endereços de IP numéricos, como 127.1.255.3

O nome do sistema central do DNS totalmente qualificado tem a vantagem de ser totalmente inequívoco e flexível. Existe a flexibilidade de alterar o endereço de IP actual para o sistema central sem ter de alterar a configuração do servidor. Este valor para o nome do sistema central é particularmente útil se planejar alterar o endereço de IP frequentemente se utilizar o protocolo DHCP para atribuir endereços de IP. Uma desvantagem deste formato é a dependência do DNS. Se o DNS não estiver disponível, então a conectividade fica comprometida.

O nome curto do sistema central também é resolvido de forma dinâmica. O formato de nome curto tem a capacidade acrescida de poder ser redefinido no ficheiro dos sistemas centrais locais para que o sistema possa executar o servidor mesmo quando está desligado da rede. Para ser possível executar desligado, deverá definir o nome curto como 127.0.0.1 (circuito fechado local) no ficheiro dos sistemas centrais. A



desvantagem do formato de nome curto e a dependência do DNS para o acesso remoto. Se o DNS não estiver disponível, então a conectividade fica comprometida.

Um endereço de IP numérico tem a vantagem de não necessitar da resolução de nomes através do DNS. Um nó remoto pode ligar-se ao nó nomeado com um endereço de IP numérico sem a necessidade do DNS estar disponível. A desvantagem deste formato é o facto do endereço de IP numérico ser fixo. A definição da propriedade hostName (nome do sistema central) terá de ser alterada nos documentos de configuração sempre que o endereço de IP da estação de trabalho for alterado. Por esta razão, não deverá utilizar um endereço de IP numérico se utilizar DHCP ou se alterar regularmente o endereço de IP. Outra desvantagem deste formato é não poder utilizar o nó se o sistema central estiver desligado da rede.

---

## BPMN 2.0

As definições de processo empresarial IBM Business Process Manager suportam a subclasse executável comum da classe de conformidade de modelação de processos BPMN 2.0, que lida com modelos que pode executar.

O BPMN (Business Process Model and Notation) é a norma fundamental dos processos no IBM Process Designer e IBM Process Center. Os diagramas do BPD (Business Process Definition) baseiam-se na especificação do BPMN. Este tópico apresenta alguns dos modos de aplicação do BPMN 2.0 no IBM Business Process Manager. Para obter informações detalhadas sobre o BPMN, consulte a página de Especificação do BPMN em <http://www.bpmn.org/>.

O IBM Business Process Manager suporta os seguintes tipos de tarefas do BPMN 2.0:

- Nenhuma (tarefa abstracta na especificação do BPMN 2.0)
- Tarefa do sistema (tarefa do serviço na especificação do BPMN 2.0)
- Tarefa de utilizador
- Script
- Tarefa de decisão (tarefa de regra de negócio na especificação do BPMN 2.0)

Os eventos de mensagens intermédias do IBM BPM fornecem funções semelhantes às tarefas de envio e recepção do BPMN.

## Notação do BPMN 2.0

Iniciar em V7.5.1, os ícones de tarefa Process Designer BPMN 2.0 nos diagramas BPD são recolhidos numa paleta simplificada e apresentados em diagramas em processamento. Os ícones mostram se a sua actividade é uma tarefa de sistema, uma tarefa de utilizador, uma tarefa de decisão, um script ou um processo de ligação. As actividades em modelos criados em versões anteriores também apresentam os tipos de tarefas BPMN 2.0 apropriados e os ícones de tarefa quando os visualiza na versão 7.5.1 ou posterior.

## Actividades e tarefas

Existem algumas alterações de terminologia das versões anteriores do Process Designer. Algumas dessas alterações envolvem tipos de actividades cujo nome foi alterado.

- As actividades do serviço (automáticas) são agora tarefas do sistema.
- As actividades do serviço (tarefas) numa pista sem ser do sistema são agora tarefas do utilizador.
- As actividades do serviço (tarefas) numa pista do sistema são agora tarefas de decisão se referenciarem um serviço de decisão.
- As actividades do serviço (tarefas) numa pista do sistema são agora tarefas do sistema se referenciarem qualquer tipo de serviço diferente de uma decisão.
- As actividades de Javascript são agora tarefas de script.
- As actividades de processos imbricados são agora processos ligados.

- As actividades externas de versões anteriores do Process Designer estão disponíveis como implementações externas de tarefas do utilizador ou do sistema.

## Portas de ligação

Não existem alterações de notação às portas de ligação de versões anteriores. No entanto, existem três alterações de terminologia. A porta de ligação de decisão é agora a *porta de ligação exclusiva*, a divisão simples ou a porta de ligação de junção é agora a *porta de ligação paralela* e a divisão condicional ou porta de ligação de junção é agora a *porta de ligação inclusiva*.

Também existe um novo tipo de porta de ligação, a *porta de ligação de eventos*. Uma porta de ligação de eventos representa um ponto de desvio num processo em que os caminhos alternativos que seguem a porta de ligação se baseiam nos eventos ocorridos, em vez de na avaliação de expressões que utilizam dados do processo (como numa porta de ligação exclusiva ou inclusiva). Um evento específico, normalmente a recepção de uma mensagem, determina o caminho que será seguido.

## Eventos de não interrupção

O BPMN 2.0 adicionou notação para eventos de não interrupção. Por predefinição, um evento limite interrompe a actividade anexada. Quando o evento é activado, a actividade pára e o símbolo continua o fluxo de sequência de saída do evento. Se o evento for definido como sendo de não interrupção, quando o evento é activado, a actividade anexa continua a ser executada em paralelo e é criado um novo símbolo que é passado pelo fluxo de sequência de saída do evento. O limite do evento é alterado para uma linha tracejada para eventos de não interrupção.

Os eventos intermédios anexados a actividades são eventos intermédios de interrupção se fecharem as actividades que lhes são anexas ou eventos intermédios de não interrupção se não fecharem as actividades que lhe são anexas.

## Evento inicial

A especificação do BPMN permite aos modelos de processo omitir símbolos de eventos de início e fim. O Process Designer requer que os modelos de processo utilizem eventos de início e paragem.

Existem vários tipos de eventos de início disponíveis no Process Designer:

### processos

- nenhum
- mensagem
- ad hoc

### subprocessos

- nenhum

### subprocessos do evento

- erro
- mensagem
- temporizador

Pode alterar o tipo de um evento de início editando as propriedades do evento. Pode possuir vários eventos de início de mensagem num processo, mas só pode utilizar um evento que não seja de início.

## Eventos de fim

Estão disponíveis quatro tipos de eventos de fim: *mensagem*, *término*, *erro* e *nenhum*. Pode alterar o tipo de um evento de fim.

Quando um processo ascendente chama um processo descendente e este último executa uma acção de evento terminar, o processo descendente pára e o processo ascendente continua para os passos seguintes.

## Subprocessos

A especificação do BPMN define dois tipos de subprocessos, incorporados e reutilizáveis. Pode reactivar ambos os tipos no Process Designer. Os subprocessos incorporados são denominados simplesmente *subprocessos* no Process Designer e são novos na versão 7.5.1. O subprocesso reutilizável do BPMN é denominado *processo ligado* no Process Designer.

O processo contém um subprocesso que é uma forma de agrupar passos de processo para reduzir a complexidade e os ecos parasitas do diagrama. Os subprocessos contraem vários passos numa actividade. O subprocesso só pode ser visualizado pelo processo no qual é definido. Um subprocesso existe no âmbito do respectivo chamador e tem acesso a todas as variáveis nesse ambiente. Não existe nenhum parâmetro a ser transmitido de e para o subprocesso incorporado.

Separado do subprocesso e do processo ligado, Process Designer possui um subprocesso de evento, que é um subprocesso especializado utilizado para tratamento de eventos. Não está ligado a outras actividades através do fluxo de sequência e ocorre apenas se o evento de início for activado.

## Processos ligados

Um subprocesso reutilizável do BPMN é denominado *processo ligado* no Process Designer. É um processo criado fora do processo actual que pode ser chamado pelo processo actual. É reutilizável, porque as outras definições do processo também podem chamar este processo. O processo ligado define os parâmetros de entrada e saída e não tem acesso ao âmbito ou ambiente do chamador. O processo ligado é semelhante ao processo imbricado disponível nas versões anteriores; não existe qualquer alteração no comportamento da actividade. Os processos imbricados anteriores passam a processos ligados. O processo ligado é semelhante a um subprocesso com um pequeno limite e está realçado na janela do Inspector.

## Ciclos

O BPMN fornece o conceito de uma actividade que pode ser repetida. A actividade pode ser atómica, o que significa que a actividade é repetida ou pode ser um subprocesso, encapsulando um conjunto de passos repetidos. Se expandir a actividade repetida, poderá ver as actividades contidas executadas repetidamente. A condição é sempre avaliada no início de cada iteração do ciclo. Não existe capacidade de avaliação no final de cada iteração do ciclo.

IBM Business Process Manager possui um *ciclo de várias ocorrências*, que é executado um determinado número de vezes com as actividades contidas executadas em sequência ou em paralelo.

## Importar processos não BPMN

É possível importar modelos criados no IBM WebSphere Business Modeler e utilizá-los no Process Designer. Para mais informações sobre a importação BPMN 2.0, consulte Mapeamento de IBM WebSphere Business Modeler elementos para IBM Business Process Manager construção. Também pode importar modelos BPMN 2.0 criados em WebSphere Business Compass da IBM, Rational Software Architect, ou outros ambientes de modelação.

---

## Definições de Processo Empresarial (BPDs)

Para modelar um processo no IBM Process Designer, tem de criar uma definição de processo empresarial (BPD). A definição de business process pode ser baseado num modelo BPMN importado.

Uma BPD é um modelo reutilizável de um processo, que define aquilo que é comum a todas as instâncias do tempo de execução desse modelo de processo. Uma BPD tem de conter um evento de

início, um evento de conclusão, pelo menos uma pista, e uma ou mais actividades. Consulte "Convenções de nomenclatura do Process Designer da IBM" nas ligações relacionadas para obter mais detalhes sobre as limitações de caracteres aplicáveis a BPDs.

Uma Definição de Processo Empresarial (BPD) tem de incluir uma pista para cada sistema ou grupo de utilizadores que participe num processo. Os controlos podem ser de participante ou de sistema. No entanto, é possível criar uma BPD que agrupe as actividades de um grupo e de um sistema numa única pista, se preferir. Consulte "Criar uma definição de processo empresarial (BPD)" nas ligações relacionadas para obter informações sobre como criar uma BPD.

É possível nomear qualquer pessoa ou grupo específico como responsável pelas actividades que ocorrem numa pista de participante. Cada pista que cria é atribuída ao grupo Todos os Utilizadores por predefinição. É possível utilizar este grupo predefinido para executar e testar a BPD no Inspector. O grupo Todos os utilizadores inclui todos os utilizadores que são membros do grupo de segurança **tw\_allusers**, que é um grupo de segurança especial que inclui automaticamente todos os utilizadores no sistema.

Uma pista de sistema contém actividades geridas por um sistema IBM Process Center específico. Cada actividade requer uma implementação, que irá definir a actividade e as propriedades da tarefa. Durante a implementação, um programador cria um serviço ou escreve o JavaScript necessário para concluir as actividades de uma pista de sistema. Consulte "Compreender os tipos de serviço" nas ligações relacionadas para obter mais informações sobre os serviços.

Para cada BPD que criar, terá de declarar variáveis para capturar dados de negócio que são passados de actividade para actividade no processo. Consulte "Gerir e mapear variáveis" nas ligações relacionadas para aprender mais sobre a implementação de variáveis.

É possível também adicionar eventos a uma BPD. Os eventos no IBM BPM podem ser activados pela ultrapassagem de um prazo, uma excepção ou a chegada de uma mensagem. O activador que pretende determina o tipo de evento que escolhe implementar. Para obter informações mais detalhadas sobre os tipos de eventos disponíveis e respectivos activadores, consulte "Eventos de modelação".

---

## Associações

No núcleo de uma arquitectura orientada para o serviço está o conceito de um *serviço*, uma unidade de funcionalidade realizada por uma interacção entre dispositivos de processamento. Uma *exportação* define a interface externa (ou ponto de acesso) de um módulo, de forma que os componentes da área de comunicações partilhada (SCA, Service Component Architecture) dentro do módulo possam oferecer os seus serviços a clientes externos. Uma *importação* define uma interface para serviços fora de um módulo, para que os serviços possam ser chamados dentro do módulo. Utiliza *associações* específicas do protocolo com importações e exportações para especificar os meios de transportar os dados para dentro ou para fora do módulo.

## Exportações

Os clientes externos podem chamar componentes SCA num módulo de integração através de uma variedade de protocolos (tais como HTTP, JMS, MQ e RMI/IIOP) com dados numa variedade de formatos (tais como XML, CSV, COBOL e JavaBeans). As exportações são componentes que recebem estes pedidos de fontes externas e que, em seguida, chamam IBM Business Process Manager componentes utilizando o modelo de programação SCA.

Por exemplo, na imagem seguinte, uma exportação recebe um pedido através do protocolo de HTTP de uma aplicação cliente. Os dados são transformados num objecto de negócio, o formato utilizado pelo componente SCA. O componente é, em seguida, chamado com o objecto de dados.

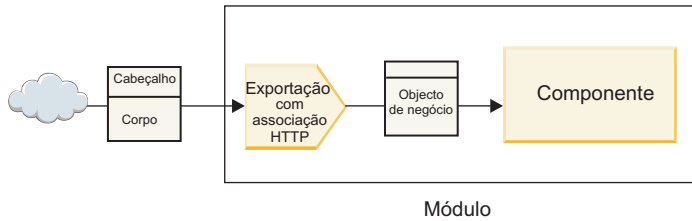


Figura 46. Uma exportação com associação HTTP

## Importações

Um componente SCA pode querer chamar um serviço externo não SCA que espera dados num formato diferente. Uma importação é utilizada pelo componente SCA para chamar o serviço externo, utilizando o modelo de programação SCA. A importação chama, em seguida, o serviço destino na forma esperada pelo serviço.

Por exemplo, na seguinte imagem é enviado um pedido de um componente SCA, pela importação, para um serviço externo. O objecto de negócio que é o formato utilizado pelo componente SCA, é transformado no formato esperado pelo serviço e o serviço é chamado.

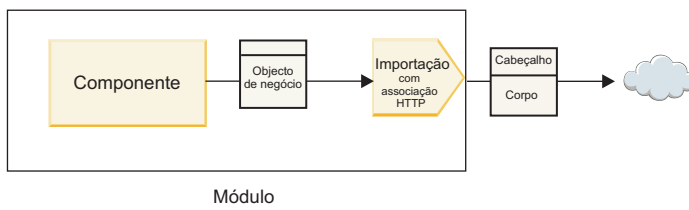


Figura 47. Uma importação com associação de HTTP

## Lista de associações

Utiliza Integration Designer para gerar uma associação para uma importação ou exportação e para configurar a associação. Os tipos de associações que estão disponíveis encontram-se na lista seguinte.

- SCA

A associação SCA, que é a predefinição, permite que o serviço comunique com serviços noutros módulos SCA. Utilize uma importação com uma associação SCA para aceder a um serviço noutro módulo SCA. Utilize uma exportação com uma associação SCA para oferecer um serviço a outros módulos SCA.

- Serviço da Web

Um serviço da web permite-lhe aceder a um serviço externo utilizando mensagens SOAP inter-operacionais e qualidades de serviço. Também pode utilizar associações do serviço da web para incluir anexos como parte da mensagem SOAP.

A associação do serviço da web pode utilizar um protocolo de transporte de SOAP/HTTP (SOAP acima de HTTP) ou SOAP/JMS (SOAP acima de JMS). Independentemente do transporte (HTTP ou JMS) utilizado para transmitir as mensagens SOAP, as associações do serviço da web processam interações de pedido/resposta de forma síncrona.

- HTTP

A associação de HTTP permite aceder a um serviço externo utilizando o protocolo HTTP, onde são utilizadas mensagens não SOAP ou onde é necessário o acesso directo de HTTP. Esta associação é utilizada quando está a trabalhar com serviços da web que se baseiam no modelo HTTP (ou seja, serviços que utilizam as conhecidas operações de interface HTTP, como GET, PUT, DELETE, etc.).

- Empresa JavaBeans (EJB)

As associações EJB permitem que os componentes SCA interajam com serviços fornecidos pela lógica de negócio Java EE a ser executada num servidor Java EE.

- EIS

A associação do Enterprise Information System (EIS), quando utilizada com um adaptador de recurso JCA, permite-lhe aceder a serviços num sistema de informação empresarial ou tornar os serviços disponíveis no EIS.

- Associações JMS

As associações de serviço de mensagens Java (JMS), JMS genérico e WebSphere MQ JMS (MQ JMS) são utilizadas para interações com sistemas de serviço de mensagens onde a comunicação assíncrona através de filas de mensagens é crítica quanto à fiabilidade.

Uma exportação com uma das associações JMS observa uma fila para a chegada de uma mensagem e envia de forma assíncrona a resposta, se existente, à fila de respostas. Uma importação com uma das associações JMS cria e envia uma mensagem à fila JMS e observa uma fila para a chegada da resposta, se existente.

- JMS

A associação JMS permite aceder ao fornecedor de JMS integrado no WebSphere.

- JMS genérico

A associação de JMS genérico permite aceder a um fornecedor de sistemas de serviço de mensagens que não pertence à IBM.

- MQ JMS

A associação MQ JMS permite aceder ao subconjunto JMS de um sistema de serviço de mensagens WebSphere MQ. Esta associação é utilizada quando o subconjunto JMS de funções for suficiente para a aplicação do utilizador.

- MQ

A associação WebSphere MQ permite comunicar com aplicações nativas MQ, trazendo-as ao contexto de arquitectura orientada para o serviço e fornecendo acesso a informação de cabeçalho específica de MQ. Utilize esta associação quando precisar de utilizar funções nativas de MQ.

## Exportar e importar descrição geral de associação

Uma exportação permite efectuar serviços num módulo de integração disponível a clientes externos e uma importação permite que os seus componentes da área de comunicações partilhada (SCA) num módulo de integração chamem serviços externos. A associação associada à exportação e importação especifica a relação entre mensagens de protocolo e objectos de negócio. Também especifica a forma como as operações e falhas são seleccionadas.

## Fluxo de informações através de uma exportação

Uma exportação recebe um pedido que se destina ao componente ao qual a exportação está ligada, através de um transporte específico determinado pela associação associada (por exemplo, HTTP).

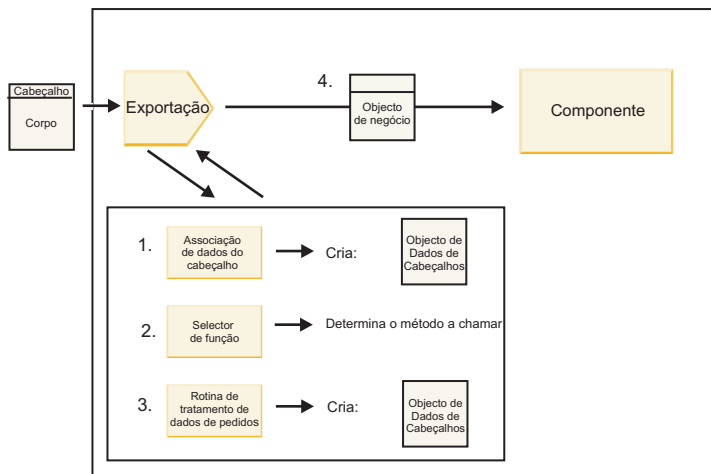


Figura 48. Fluxo de um pedido através da exportação para um componente

Quando a exportação recebe o pedido, aparece a seguinte sequência de eventos:

1. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho transforma o cabeçalho do protocolo num objecto de dados de cabeçalho.
2. O selector de função determina o nome do método nativo a partir da mensagem de protocolo. O nome do método nativo é mapeado pela configuração de exportação para o nome de uma operação na interface da exportação.
3. A rotina de tratamento de dados ou associação de dados de pedido no método transforma o pedido num objecto de negócio de pedido.
4. A exportação chama o método de componente com o objecto de negócio de pedido.
  - A associação de exportação HTTP, a associação de exportação do serviço da web, e a associação de exportação EJB chamam o componente da área de comunicações partilhada (SCA) de forma síncrona.
  - As associações de exportação JMS, Generic JMS, MQ JMS e WebSphere MQ chamam o componente SCA de forma síncrona.

Tenha em atenção que uma exportação pode propagar os cabeçalhos e as propriedades de utilizador que recebe através do protocolo, se a propagação de contexto estiver activada. Os componentes que estão ligados à exportação podem aceder a estes cabeçalhos e às propriedades de utilizador. Consulte o tópico “Propagation (Propagação)” no Information Center do WebSphere Integration Developer para obter mais informações.

Se esta for uma operação bidireccional, o componente devolve uma resposta.



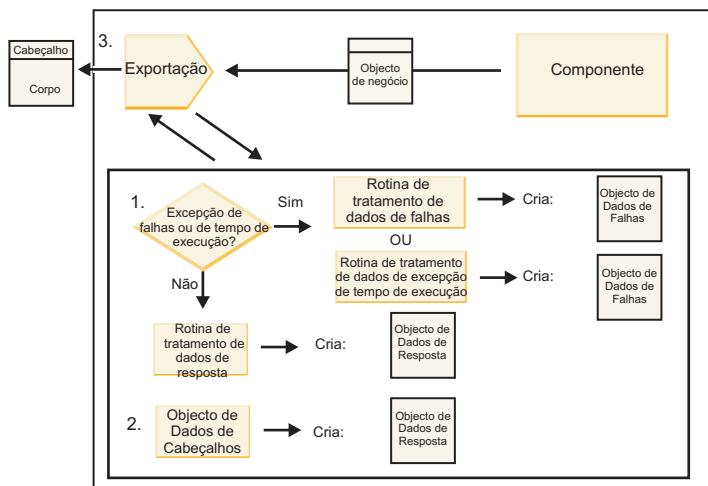


Figura 49. Fluxo de uma resposta de volta pela exportação

Aparece a seguinte sequência de passos:

1. Se uma mensagem de resposta normal for recebida pela associação de exportação, a rotina de tratamento de dados ou a associação de dados no método transforma o objecto de negócio numa resposta.

Se a resposta for uma falha, a rotina de tratamento de dados ou associação de dados de falha no método transforma a falha numa resposta de falha.

Apenas para associações de exportação HTTP, se a resposta for uma excepção de tempo de execução, a rotina de tratamento de dados de excepção de tempo de execução, se estiver configurada, é chamada.

2. Apenas para associações WebSphere, a associação de dados de cabeçalho transforma os objectos de dados de cabeçalho em cabeçalhos de protocolo.
3. A exportação envia a resposta de serviço através do transporte.

## Fluxo de informações através de uma importação

Os componentes enviam pedidos a serviços fora do módulo, utilizando uma importação. O pedido é enviado através de um transporte específico determinado pela associação associada.

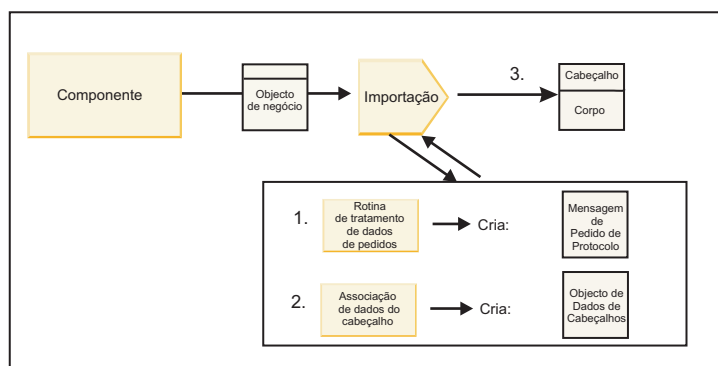


Figura 50. Fluxo de um componente através da importação para um serviço

O componente chama a importação com um objecto de negócio de pedido.

**Nota:**

- A associação de importação de HTTP, a associação de importação do serviço da web e a associação de importação de EJB devem ser chamadas de forma síncrona pelo componente de chamada.
- As associações de importação JMS, Generic JMS, MQ JMS e WebSphere MQ devem ser chamadas de forma assíncrona.

Depois do componente chamar a importação, aparece a seguinte sequência de eventos:

1. A rotina de tratamento de dados ou associação de dados de pedido no método transformam o objecto de negócio de pedido numa mensagem de pedido de protocolo.
2. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho no método define o objecto de negócio de cabeçalho no cabeçalho do protocolo.
3. A importação chama o serviço com o pedido de serviço através do transporte.

Se esta for uma operação bidireccional, o serviço devolve uma resposta e é apresentada a seguinte sequência de passos:

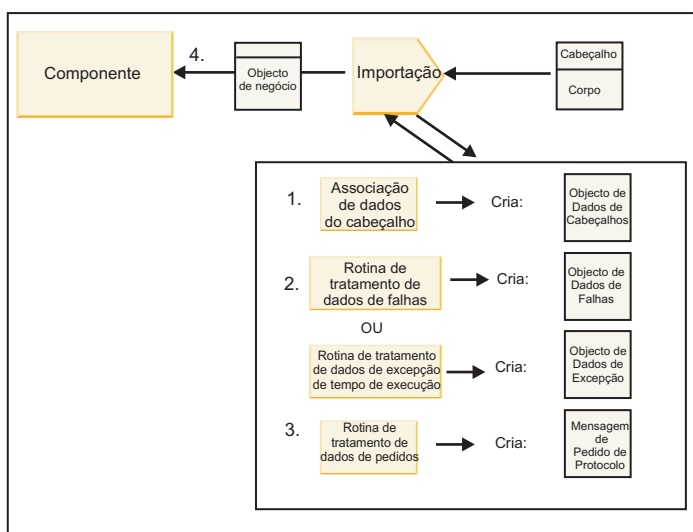


Figura 51. Fluxo de uma resposta de volta pela importação

1. Apenas para associações WebSphere MQ, a associação de dados de cabeçalho transforma o cabeçalho do protocolo num objecto de dados de cabeçalho.
2. É determinado se a resposta é uma falha.
  - Se a resposta for uma falha, o selector de falhas inspeciona a falha para determinar em que falha WSDL é mapeada. A rotina de tratamento de dados de falhas no método transforma, assim, a falha numa resposta de falha.
  - Se a resposta for uma excepção de tempo de execução, a rotina de tratamento de dados da excepção de tempo de execução, se estiver configurada, é chamada.
3. A rotina de tratamento de dados de resposta ou associação no método transforma a resposta num objecto de negócio de resposta.
4. A importação devolve o objecto de negócio de resposta ao componente.

## Exportar e importar configuração de associação

Um dos aspectos fundamentais de exportar e importar associações é a transformação do formato de dados que indica de que forma os dados são mapeados (anular a serialização) de um formato de ligação nativo para um objecto de negócio ou de que forma é mapeado (serializado) de um objecto de negócio para um formato de ligação nativo. Para associações associadas a exportações, também pode especificar

um selector de função para indicar qual a operação a executar nos dados. Para associações associadas a exportações e importações, pode indicar de que forma devem ser processadas as falhas que ocorrem durante o processamento.

Além disso, pode especificar informações sobre associações específicas de transporte. Por exemplo, para uma associação HTML especifica o URL do ponto final. Para a associação HTTP, as informações específicas de transporte encontram-se descritas nos tópicos “Gerar uma associação de importação HTTP” e “Gerar uma associação de exportação HTTP”. Também é possível encontrar informações sobre outras associações no Information Center.

## Transformação de formato de dados em importações e exportações

Quando uma associação de exportação ou importação é configurada em IBM Integration Designer, uma das propriedades de configuração que especifica é o formato de dados utilizado pela associação.

- Para associações de exportação, nas quais uma aplicação de cliente envia pedidos e recebe respostas de um componente SCA, indique o formato dos dados nativos. Dependendo do formato, o sistema selecciona a rotina de tratamento de dados apropriada para transformar os dados nativos num objecto de negócio (que é utilizado pelo componente SCA) e para transformar inversamente o objecto de negócio em dados nativos (que é a resposta à aplicação de cliente).
- Para associações de importação nas quais um componente SCA envia pedidos e recebe respostas de um serviço fora do módulo, indique o formato de dados dos dados nativos. Dependendo do formato, o sistema selecciona a rotina de tratamento de dados apropriada ou associação de dados para transformar o objecto de negócio em dados nativos ou vice-versa.

IBM Business Process Manager fornece um conjunto de formatos de dados predefinidos e rotinas de tratamento de dados correspondentes ou associações de dados que suportem os formatos. Também pode criar as suas próprias rotinas de tratamento de dados personalizadas e registar o formato de dados para essas rotinas de tratamento de dados. Para mais informações, consulte o tópico “desenvolver rotinas de tratamento de dados” no Information Center IBM Integration Designer.

- *Rotinas de tratamento de dados* são neutros de protocolo e transformam dados de um formato para outro. No IBM Business Process Manager, as rotinas de tratamento de dados transformam, normalmente, os dados nativos (como XML, CSV e COBOL) num objecto de negócio e um objecto de negócio em dados nativos. Pelo facto de serem neutras de protocolo, pode reutilizar a mesma rotina de tratamento de dados com uma variedade de associações de exportação e importação. Por exemplo, pode utilizar a mesma rotina de tratamento de dados XML com uma associação de exportação ou importação HTTP ou com uma associação de exportação ou importação JMS.
- *As associações de dados* também transformam dados nativos num objecto de negócio (e vice-versa), mas são específicas de protocolo. Por exemplo, uma associação de dados HTTP pode ser utilizada com uma associação de exportação ou importação HTTP apenas. Ao contrário das rotinas de tratamento de dados, uma associação de dados HTTP não pode ser reutilizada com uma associação de exportação ou importação MQ.

**Nota:** Três associações de dados HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão obsoletas a partir da IBM Business Process Manager versão 7.0. Utilize rotinas de tratamento de dados sempre que possível.

Como referido anteriormente, é possível criar rotinas de tratamento de dados personalizadas, se necessário. Também pode criar associações de dados personalizadas; no entanto, recomenda-se que crie rotinas de tratamento de dados personalizadas uma vez que podem ser utilizadas em múltiplas associações.

### Rotinas de tratamento de dados:

As rotinas de tratamento de dados são configuradas contra associações de exportação e importação para transformar dados de um formato para outro numa forma neutra a protocolo. Várias rotinas de tratamento de dados são fornecidas como parte do produto, mas também pode criar a sua própria rotina

de tratamento de dados, se necessário. Pode associar uma rotina de tratamento de dados a uma associação de exportação ou importação em um de dois níveis: pode associá-la a todas as operações na interface da exportação ou importação ou pode associá-la a uma operação específica para o pedido ou resposta.

### Rotinas de tratamento de dados predefinidas

Utilize o IBM Integration Designer para especificar a rotina de tratamento de dados que pretende utilizar.

As rotinas de tratamento de dados que estão predefinidas para a sua utilização estão listadas na seguinte tabela que também descreve de que forma cada rotina de tratamento de dados transforma dados internos e externos.

**Nota:** Excepto onde notado, estas rotinas de tratamento de dados podem ser utilizadas com associações JMS, Generic JMS, MQ JMS, WebSphere MQ, e HTTP.

Consulte o tópico “Data handlers (Rotinas de tratamento de dados)” no Information Center do Integration Designer para obter informações mais detalhadas.

Tabela 52. Rotinas de tratamento de dados predefinidas

Rotina de tratamento de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
ATOM	Analisa ATOM feeds num objecto de negócio ATOM feed.	Dispõe em série um objecto de negócio ATOM feed para ATOM feeds.
Delimitado	Analisa dados delimitados num objecto de negócio.	Dispõe em série um objecto de negócio para dados delimitados, incluindo CSV.
Largura fixa	Analisa dados de largura fixa no objecto de negócio.	Dispõe em série um objecto de negócio para dados de largura fixa.
Processado por WTX	Delega a conversão do formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX deriva da rotina de tratamento de dados.	Delega a conversão do formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX deriva da rotina de tratamento de dados.
Processado por WTX Invoker	Delega a transformação de formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX é fornecido pelo utilizador.	Delega a transformação de formato de dados ao WebSphere Transformation Extender (WTX). O nome de mapa WTX é fornecido pelo utilizador.
JAXB	Dispõe em série beans de Java para um objecto de negócio utilizando as regras de mapeamento definidas pela Arquitectura Java para especificação de associação XML (JAXB).	Anula a serialização de um objecto de negócio para beans de Java utilizando as regras de mapeamento definidas pela especificação JAXB.
JAXWS <b>Nota:</b> A rotina de tratamento de dados JAXWS pode ser utilizada apenas com a associação EJB.	Utilizada por uma associação EJB para transformar um objecto Java de resposta ou objecto Java de excepção para um objecto de negócio de resposta, utilizando as regras de mapeamento definidas pela especificação Java API para serviços da Web XML (JAX-WS).	Utilizada por uma associação EJB para transformar um objecto de negócio em parâmetros de método Java de saída, utilizando as regras de mapeamento definidas pela especificação JAX-WS.
JSON	Analisa dados JSON num objecto de negócio.	Dispõe em série um objecto de negócio para dados JSON.

Tabela 52. Rotinas de tratamento de dados predefinidas (continuação)

Rotina de tratamento de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Corpo nativo	Analisa os bytes nativos, texto, mapa, sequência ou objecto num dos cinco objectos de negócio (texto, bytes, mapa, sequência ou objecto).	Transforma os cinco objectos de negócio base em byte, texto, mapa, sequência ou objecto.
SOAP	Analisa a mensagem SOAP (e o cabeçalho) num objecto de negócio.	Dispõe em série um objecto de negócio para uma mensagem SOAP.
XML	Analisa dados XML num objecto de negócio.	Dispõe em série um objecto de negócio para dados XML.
UTF8XMLDataHandler	Analisa dados XML codificados em UTF-8 num objecto de negócio.	Dispõe em série um objecto de negócio em dados XML codificados em UTF-8 ao enviar uma mensagem.

### Criar uma rotina de tratamento de dados

Informações detalhadas sobre como criar uma rotina de tratamento de dados podem ser encontradas no tópico “Developing data handlers (Desenvolver rotinas de tratamento de dados)” no Information Center Integration Designer.

### Associações de dados:

As associações de dados estão configuradas contra associações de exportação e importação para transformar dados de um formato para outro. As associações de dados são específicas para um protocolo. São fornecidas várias associações de dados como parte do produto, mas também pode criar a sua própria associação de dados, se necessário. Pode associar uma associação de dados com uma exportação ou importação a um de dois níveis-pode associá-la a todas as operações na interface da exportação ou importação ou pode associá-la com uma operação específica para o pedido ou resposta.

Utilize IBM Integration Designer para especificar qual a associação de dados que pretende utilizar ou para criar a sua própria associação de dados. Pode encontrar um debate sobre a criação de associações de dados na secção de “descrição geral de associações JMS, MQ JMS e JMS genérico” do Information Center IBM Integration Designer.

### Associações JMS

A seguinte tabela lista as associações de dados que podem ser utilizadas com:

- Associações JMS
- Associações genéricas JMS
- Associações JMS do WebSphere MQ

A tabela também inclui uma descrição das tarefas executadas pelas associações de dados.

Tabela 53. Associações de dados predefinidas para associações JMS

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Objecto Java serializado	Transforma o objecto serializado Java num objecto de negócio (que está correlacionado com o tipo de entrada ou saída no WSDL).	Dispõe em série um objecto de negócio para o objecto disposto em série Java na mensagem de objecto JMS.

Tabela 53. Associações de dados predefinidas para associações JMS (continuação)

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Bytes reiniciados ciclicamente	Extrai os bytes da mensagem de bytes JMS de chegada e reinicia-os ciclicamente no objecto de negócio JMSBytesBody.	Extrai os bytes do objecto de negócio JMSBytesBody e reinicia-os ciclicamente na mensagem de bytes JMS de saída.
Entrada de mapa reiniciada ciclicamente	Extrai o nome, valor e informação de tipo para cada entrada na mensagem de mapa JMS de entrada e cria uma lista de objectos de negócio MapEntry. Em seguida, reinicia ciclicamente a lista no objecto de negócio JMSMapBody.	Extrai o nome, valor e informação de tipo da lista MapEntry no objecto de negócio JMSMapBody e cria as entradas correspondentes na mensagem de correlação JMS de saída.
Objecto reiniciado ciclicamente	Extrai o objecto da mensagem de objecto JMS de entrada e reinicia-o ciclicamente no objecto de negócio JMSObjectBody.	Extrai o objecto do objecto de negócio JMSObjectBody e reinicia-o ciclicamente na mensagem de objecto JMS de saída.
Texto reiniciado ciclicamente	Extrai o texto da mensagem de texto JMS de entrada e reinicia-o ciclicamente no objecto de negócio JMSTextBody.	Extrai o texto do objecto de negócio JMSTextBody e reinicia-o ciclicamente na mensagem de texto JMS de saída.

## Associações do WebSphere MQ

A seguinte tabela lista as associações de dados que podem ser utilizadas com o WebSphere MQ e descreve as tarefas que as associações de dados executam.

Tabela 54. Associações de dados predefinidas para associações WebSphere MQ

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Objecto Java serializado	Transforma o objecto Java serializado da mensagem recebida num objecto de negócio (que está mapeado como o tipo de entrada ou saída no WSDL).	Transforma um objecto de negócio num objecto serializado Java na mensagem de saída.
Bytes reiniciados ciclicamente	Extrai os bytes da mensagem de bytes MQ não estruturada e reinicia-os ciclicamente no objecto de negócio JMSBytesBody.	Extrai os bytes de um objecto de negócio JMSBytesBody e reinicia ciclicamente os bytes na mensagem de bytes MQ não estruturada de saída.
Texto reiniciado ciclicamente	Extrai o texto da mensagem de texto MQ não estruturada e reinicia-o ciclicamente no objecto de negócio JMSTextBody.	Extrai o texto de um objecto de negócio JMSTextBody e reinicia-o ciclicamente numa mensagem de texto MQ não estruturada.
Entrada sequencial de reinício cíclico	Extrai o nome e informação de tipo para cada entrada na mensagem sequencial JMS de entrada e cria uma lista dos objectos de negócio StreamEntry. Em seguida, reinicia ciclicamente a lista no objecto de negócio JMSStreamBody.	Extrai o nome e informação de tipo da lista StreamEntry no objecto de negócio JMSStreamBody e cria entradas correspondentes na mensagem de saída JMSStream.

Para além das associações de dados enunciadas na Tabela 25 na página 60, o WebSphere MQ também utiliza associações de dados de cabeçalho. Para mais informações, consulte o Information Center IBM Integration Designer.

### Associações HTTP

A seguinte tabela lista as associações de dados que podem ser utilizadas com HTTP e descreve as tarefas que as associações de dados podem executar.

Tabela 55. associações de dados predefinidas para associações HTTP

Associação de dados	Dados nativos para objecto de negócio	Objecto de negócio para dados nativos
Bytes reiniciados ciclicamente	Extrai os bytes do corpo da mensagem HTTP de entrada e reinicia-os ciclicamente no objecto de negócio HTTPBytes.	Extrai os bytes do objecto de negócio HTTPBytes e inclui-os no corpo da mensagem HTTP de saída.
Texto reiniciado ciclicamente	Extrai o texto do corpo da mensagem HTTP de entrada e reinicia-o ciclicamente no objecto de negócio HTTPText.	Extrai o texto do objecto de negócio HTTPText e inclui-o na mensagem HTTP de saída.

### Selectores de função em associações de exportação

Um selector de função é utilizado para indicar qual a operação a executar nos dados para uma mensagem de pedido. Os selectores de função estão configurados como parte de uma associação de exportação.

Considere uma exportação SCA que expõe uma interface. A interface contém duas operações-Criar e Actualizar. A exportação tem uma associação JMS que lê de uma fila.

Quando chega uma mensagem à fila, a exportação é passada aos dados associados, mas que operação da interface da exportação deve ser chamada no componente ligado? A operação é determinada pelo selector de função e a configuração de associação de exportação.

O selector de função devolve o nome de função nativo (o nome de função no sistema cliente que enviou a mensagem). O nome de função nativo é depois mapeado para a operação ou nome de função na interface associada à exportação. Por exemplo, na imagem seguinte, o selector de função devolve o nome de função nativo (CRT) da mensagem recebida, o nome de função nativo é mapeado para a operação Criar e o objecto de negócio é enviado ao componente SCA com a operação Criar.

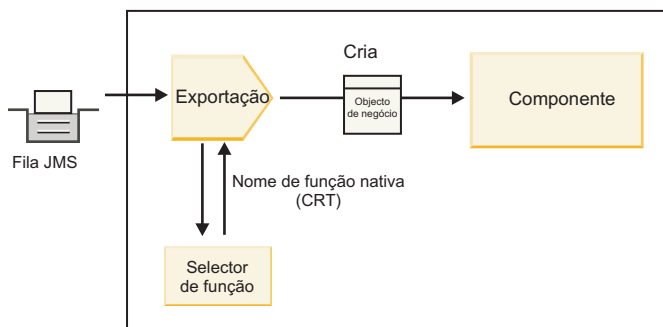


Figura 52. O selector de função

Se a interface tiver apenas uma operação, não há necessidade de especificar um selector de função.

Vários selectores de função pré-emballados encontram-se disponíveis e listados nas seguintes secções.



## Associações JMS

A seguinte tabela lista os selectores de função que podem ser utilizados com:

- Associações JMS
- Associações genéricas JMS
- Associações JMS do WebSphere MQ

Tabela 56. Selectores de função predefinidos para associações JMS

Selector de função	Descrição
Selector de função JMS para associações de dados JMS simples	Utiliza a propriedade JMSType da mensagem para seleccionar o nome de operação.
Selector de função de propriedade de cabeçalho JMS	Devolve o valor da propriedade de cadeia JMS TargetFunctionName do cabeçalho.
Selector de função de porta de ligação de serviço JMS	Determina se o pedido é uma operação unidireccional ou bidireccional ao examinar a propriedade JMSReplyTo definida pelo cliente.

## Associações WebSphere MQ

A seguinte tabela lista os selectores de função que podem ser utilizados com associações WebSphere MQ.

Tabela 57. Selectores de função predefinidos para associações WebSphere MQ

Selector de função	Descrição
Selector de função MQ handleMessage	Devolve handleMessage como um valor que é mapeado utilizando as associações de método de exportação para o nome de uma operação na interface.
MQ utiliza o selector de função predefinido JMS	Lê a operação nativa a partir da propriedade TargetFunctionName da pasta de um cabeçalho MQRFH2.
MQ utiliza o formato de corpo de mensagem como função nativa	Encontra o campo Formatar do último cabeçalho e devolve esse campo como cadeia.
Selector de função do tipo MQ	Cria um método na associação de exportação ao obter um URL contendo as propriedades Msd, Set, Type (Tipo) e Format (Formatar) no cabeçalho MQRFH2.
Selector de função de porta de ligação de serviço MQ	Utiliza a propriedade MsgType no cabeçalho MQMD para determinar o nome da operação.

## Associações HTTP

A seguinte tabela lista os selectores de função que podem ser utilizados com associações HTTP.

Tabela 58. Selectores de função predefinidos para associações HTTP

Selector de função	Descrição
Selector de função HTTP baseado no cabeçalho TargetFunctionName	Utiliza a propriedade de cabeçalho HTTP TargetFunctionName do cliente para determinar qual a operação a chamar no tempo de execução da exportação.
Selector de função HTTP baseado no método URL e HTTP	Utiliza o caminho relativo do URL juntamente com o método HTTP do cliente para determinar a operação nativa definida na exportação.
O selector de função de porta de ligação de serviço HTTP baseado no URL com um nome de operação	Determina o método para chamar com base no URL se "operationMode = oneWay" foi anexado ao pedido URL.

**Nota:** Também pode criar o seu próprio selector de função, utilizando o IBM Integration Designer. São fornecidas informações sobre como criar um selector de funções no Information Center do IBM Integration Designer. Por exemplo, uma descrição da criação de um selector de função para associações WebSphere MQ podem ser encontradas em “Descrição geral dos selectores de função MQ”.

## **Processamento de falhas**

Pode configurar as associações de importação e exportação para processar falhas (por exemplo, excepções de negócio) que ocorrem durante o processamento ao especificar rotinas de tratamento de dados de falha. Pode definir uma rotina de tratamento de dados em três níveis-pode associar uma rotina de tratamento de dados de falha com uma falha, com uma operação ou para todas as operações com uma associação.

Uma rotina de tratamento de dados de falha processa dados de falha e transforma-os no formato correcto para serem enviados pela associação de exportação ou importação.

- Para uma associação de exportação, a rotina de tratamento de dados de falha transforma o objecto de negócio enviado pelo componente numa mensagem de resposta que pode ser utilizada pela aplicação do cliente.
- Para uma associação de importação, a rotina de tratamento de dados de falha transforma os dados de falha ou mensagem de resposta enviados por um serviço num objecto de negócio de excepção pode ser utilizada pelo componente SCA.

Para associações de importação, a associação chama o selector de falha, que determina se a mensagem de resposta é uma resposta normal, uma falha de negócio ou uma excepção de tempo de execução.

Pode especificar uma rotina de tratamento de dados de falha para uma determinada falha, para uma operação e para todas as operações com uma associação.

- Se a rotina de tratamento de dados de falha estiver definida em todos os três níveis, a rotina de tratamento de dados associada a uma determinada falha é chamada.
- Se as rotinas de tratamento de dados de falha estiverem definidas nos níveis de operação e associação, a rotina de tratamento de dados associada à operação é chamada.

São utilizados editores no IBM Integration Designer para especificar processamento de falhas. O editor de interface é utilizado para indicar se vai haver uma falha na operação. Depois de ser gerada uma associação com esta interface, o editor na vista de propriedades permite configurar a forma como a falha será processada. Para mais informações, consulte o tópico “Selector de falhas” nas informações do Information Center do IBM Integration Designer.

### **De que forma são processadas as falhas nas associações de exportação:**

Quando há ocorrência de uma falha durante o processamento do pedido de uma aplicação de cliente, a associação de exportação pode devolver a informação de falha ao cliente. Pode configurar a associação de exportação para especificar de que forma a falha deve ser processada e devolvida ao cliente.

Configure a associação de exportação utilizando IBM Integration Designer.

Durante o processamento do pedido, o cliente chama uma exportação com um pedido e a exportação chama o componente SCA. Durante o processamento do pedido, o componente SCA pode devolver uma resposta de negócio ou pode lançar uma excepção de negócio de serviço ou uma excepção de tempo de execução de serviço. Quando isto ocorre, a associação de exportação transforma a excepção numa mensagem de falha e envia-a ao cliente, como apresentado na seguinte imagem e descrito nas secções que se seguem.

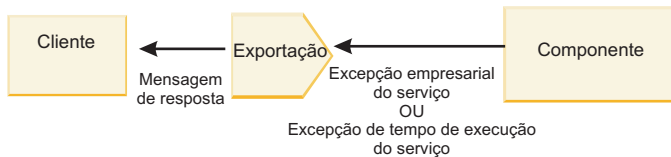


Figura 53. De que forma a informação de falha é enviada do componente através da associação de exportação ao cliente

Pode criar uma rotina de tratamento de dados personalizada ou associação de dados para processar falhas.

## Falhas de negócio

Falhas de negócio são falhas de negócio ou excepções que ocorrem durante o processamento.

Considere a seguinte interface que tem uma operação createCustomer. Esta operação tem duas falhas de negócio definidas: CustomerAlreadyExists e MissingCustomerId.

Operações

Operações e os respectivos parâmetros

	Nome	Tipo
createCustomer		
Entrada(s)	entrada	CustomerInfo
Saída(s)	Saída	CustomerInfo
Falha	Cliente já existe	Customer Already ExistsBO
Falha	MissingCustomerId	MissingCustomerIdBO

Figura 54. Interface com duas falhas

Neste exemplo, se um cliente envia um pedido para criar um cliente (para este componente SCA) e esse cliente já existir, o componente lança uma falha CustomerAlreadyExists para a exportação. A exportação precisa de propagar esta falha de negócio de volta para o cliente de chamada. Para esse efeito, utiliza a rotina de tratamento de dados que está definida na associação de exportação.

Quando uma falha de negócio é recebida pela associação de exportação, ocorre o seguinte processamento:

1. A associação determina qual a rotina de tratamento de dados a chamar para processar a falha. Se a excepção de negócio de serviço contiver o nome da falha, é chamada a rotina de tratamento de dados que está definida na falha. Se a excepção de negócio de serviço não contiver o nome da falha, o nome da falha é derivado ao corresponder os tipos de falha.
2. A associação chama a rotina de tratamento de dados com o objecto de dados da excepção de negócio de serviço.
3. A rotina de tratamento de dados falhada transforma o objecto de dados de falha numa mensagem de resposta e devolve-a à associação de exportação.
4. A exportação devolve a mensagem de resposta ao cliente.

Se a excepção de negócio de serviço contiver o nome da falha, é chamada a rotina de tratamento de dados que está definida na falha. Se a excepção de negócio de serviço não contiver o nome da falha, o nome da falha é derivado ao corresponder os tipos de falha.

## Excepções de tempo de execução

Uma excepção de tempo de execução é uma excepção que ocorre na aplicação SCA durante o processamento de um pedido que não corresponde à falha de negócio. Ao contrário das falhas de negócio, as excepções de tempo de execução não estão definidas na interface.

Em determinados cenários pode querer propagar estas excepções de tempo de execução pela aplicação de cliente de forma que a aplicação do cliente possa tomar a acção adequada.

Por exemplo, se um cliente enviar um pedido (ao componente SCA) para criar um cliente e um erro de autorização ocorrer durante o processamento do pedido, o componente lança uma excepção de tempo de execução. Esta excepção de tempo de execução tem de ser propagada de volta ao cliente de chamada para que possa tomar a acção adequada quanto à autorização. Isto é alcançado através da rotina de tratamento de dados de excepção de tempo de execução configurada na associação de exportação.

**Nota:** Pode configurar uma rotina de tratamento de dados de excepção de tempo de execução apenas em associações HTTP.

O processamento de uma excepção de tempo de execução é semelhante ao processamento de uma falha de negócio. Se tiver sido definida uma rotina de tratamento de dados de excepção de tempo de execução, ocorre o seguinte processamento:

1. A associação de exportação chama a rotina de tratamento de dados adequada com a excepção de tempo de execução de serviço.
2. A rotina de tratamento de dados transforma o objecto de dados de falha numa mensagem de resposta e devolve-a à associação de exportação.
3. A exportação devolve a mensagem de resposta ao cliente.

O processamento de falhas e o processamento de excepções de tempo de execução são opcionais. Se não pretender propagar falhas ou excepções de tempo de execução pelo cliente de chamada, não configure a rotina de tratamento de dados de falha ou rotina de tratamento de dados de excepção de tempo de execução.

### Como são processadas falhas em associações de importação:

Um componente utiliza uma importação para enviar um pedido a um serviço fora do módulo. Quando ocorre uma falha durante o processamento do pedido, o serviço devolve a falha à associação de importação. Pode configurar a associação de importação para especificar de que forma a falha deve ser processada e devolvida ao componente.

Pode configurar a associação de importação utilizando IBM Integration Designer. É possível especificar uma rotina de tratamento de dados de falha (ou associação de dados) e especifica também um selector de falhas.

### Rotinas de tratamento de dados de falhas

O serviço que processa o pedido envia à associação de importação informações de falha em forma de uma excepção ou mensagem de resposta que contém os dados de falha.

A associação de importação transforma a excepção de serviço ou mensagem de resposta numa excepção de negócio de serviço ou excepção de tempo de execução de serviço, como indicado na imagem seguinte e descrito nas secções seguintes.

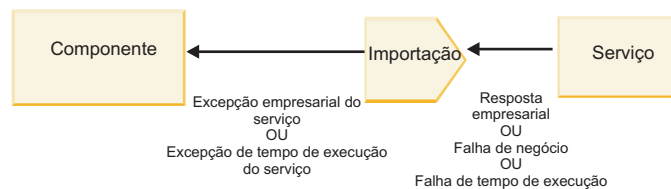


Figura 55. De que forma são enviadas as informações de falha do serviço através da importação para o componente

Pode criar uma rotina de tratamento de dados personalizada ou associação de dados para processar falhas.

### Selectores de falha

Quando configura uma associação de importação, pode especificar um selector de falhas. O selector de falhas determina se a resposta de importação é uma resposta efectiva, uma excepção de negócio ou uma falha de tempo de execução. Determina também, do corpo de resposta ou cabeçalho, o nome de falha nativo que é mapeado pela configuração de associação para o nome de uma falha na interface associada.

Dois tipos de selectores de falhas pré-embalados estão disponíveis para utilização com JMS, MQ JMS, JMS Genérico JMS, WebSphere MQ e importações HTTP:

Tabela 59. Selectores de falhas pré-embalados

Tipo de selector de falhas	Descrição
Com base no cabeçalho	Determina se uma mensagem de resposta é uma falha de negócio, uma excepção de tempo de execução ou uma mensagem normal com base nos cabeçalhos na mensagem de resposta recebida.
SOAP	Determina se a mensagem SOAP de resposta é uma resposta normal, uma falha de negócio ou uma excepção de tempo de execução.

O que se segue mostra exemplos de selectores de falhas com base no cabeçalho e do selector de falhas SOAP.

- Selector de falhas com base no cabeçalho

Se uma aplicação pretender indicar que a mensagem recebida é uma falha de negócio, têm de existir dois cabeçalhos na mensagem recebida para falhas de negócio, que são apresentados da seguinte forma:

Nome do cabeçalho = FaultType, Valor do cabeçalho = Business  
Nome do cabeçalho = FaultName, Valor do cabeçalho = <nome de falha nativo definido pelo utilizador>

Se uma aplicação pretender indicar que a mensagem de resposta recebida é uma excepção de tempo de execução, tem de existir um cabeçalho na mensagem de entrada, que é mostrada da seguinte forma:

Header name = FaultType, Header value = Runtime

- Selector de falhas SOAP

Uma falha de negócio pode ser enviada como parte da mensagem SOAP com o seguinte cabeçalho SOAP personalizado. "CustomerAlreadyExists" é o nome da falha neste caso.

```
<ibmSoap:BusinessFaultName  
xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists  
</ibmSoap:BusinessFaultName>
```

O selector de falhas é opcional. Se não especificar um selector de falhas, a associação de importação não pode determinar o tipo de resposta. Assim, a associação processa-o como uma resposta de negócio e chama a rotina de tratamento de dados ou associação de dados.

É possível criar um selector de falhas personalizado. Os passos para criar um selector de falhas personalizado são fornecidos no tópico "Developing a custom fault selector (Desenvolver um selector de falhas personalizado)" do Information Center do IBM Integration Designer.

### Falhas de negócio

Uma falha de negócio pode ocorrer quando existe um erro no processamento de um pedido. Por exemplo, se enviar um pedido para criar um cliente e esse cliente já existe, o serviço envia uma excepção de negócio à associação de importação.

Quando uma excepção de negócio é recebida pela associação, os passos de processamento dependem de um selector de falhas ter sido configurado para a associação.

- Se não tiver sido configurado nenhum selector de falhas, a associação chama a rotina de tratamento de dados de resposta ou associação de dados.
- Se tiver sido configurado um selector de falhas, ocorre o seguinte processamento:
  1. A associação de importação chama o selector de falhas para determinar se a resposta é uma falha de negócio, uma resposta de negócio ou uma falha de tempo de execução.
  2. Se a resposta for uma falha de negócio, a associação de importação chama o selector de falhas para fornecer o nome de falha nativo.
  3. A associação de importação determina a falha WSDL correspondente ao nome de falha nativo devolvido pelo selector de falhas.
  4. A associação de importação determina a rotina de tratamento de dados de falha que está configurada para esta falha WSDL.
  5. A associação de importação chama esta rotina de tratamento de dados de falha com os dados de falha.
  6. A rotina de tratamento de dados de falha transforma os dados de falha num objecto de dados de devolve-o à associação de importação.
  7. A associação de importação constrói um objecto de excepção de negócio de serviço com o objecto de dados e o nome da falha.
  8. A importação devolve a excepção de negócio de serviço ao componente.

### **Excepções de tempo de execução**

Uma excepção de tempo de execução pode ocorrer quando existe um problema em comunicar com o serviço. O processamento de uma exception de tempo de execução é semelhante ao processamento de uma excepção de negócio. Se tiver sido configurado um selector de falhas, ocorre o seguinte processamento:

1. A associação de importação chama a rotina de tratamento de dados de excepção de tempo de execução apropriada com os dados de excepção.
2. A rotina de tratamento de dados de excepção de tempo de execução transforma os dados de excepção num objecto de excepção de tempo de execução de serviço e devolve-o à associação de importação.
3. A importação devolve o objecto de excepção de tempo de execução ao componente.

## **Interoperacionalidade entre módulos SCA e serviços Open SCA**

O IBM WebSphere Application Server V7.0 Feature Pack for Service Component Architecture (SCA) fornece um modelo de programação simples mas poderoso para construir aplicações baseadas nas especificações Open SCA. Os módulos SCA de IBM Business Process Manager utilizam associações de importação e exportação para interoperar com serviços Open SCA desenvolvidos num ambiente Rational Application Developer e alojados pelo WebSphere Application Server Feature Pack for Service Component Architecture.

Uma aplicação SCA chama uma aplicação Open SCA através de uma associação de importação. Uma aplicação SCA recebe uma aplicação Open SCA através de uma associação de exportação. Uma lista de associações suportadas é apresentada na “Chamar serviços através de associações interoperáveis” na página 69.

### **Chamar Open SCA a partir de módulos SCA**

As aplicações SCA desenvolvidas com o IBM Integration Designer podem chamar aplicações Open SCA desenvolvidas num ambiente Rational Application Developer. Esta secção fornece um exemplo de chamada de um serviço Open SCA a partir de um módulo SCA utilizando uma associação de importação

SCA.

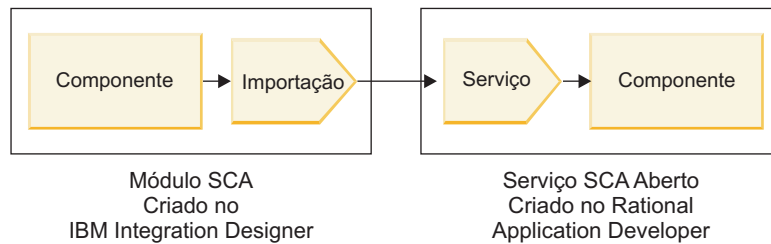


Figura 56. Componente em módulo SCA a chamar serviço Open SCA

Não é necessária qualquer configuração especial para chamar um serviço Open SCA.

Para ligar a um serviço Open SCA através de uma associação de importação SCA, fornece o nome do componente e o nome de serviço do serviço Open SCA na associação de importação.

1. Para obter o nome do componente de destino e serviço do composto Open SCA, execute os seguintes passos:
  - a. Certifique-se que o separador **Properties (Propriedades)** está aberto ao fazer clique em **Window (Janela) > Show View (Mostrar vista) > Properties (Propriedades)**.
  - b. Abra o editor de composto através de duplo clique no diagrama que contém o componente e serviço. Por exemplo, para um componente chamado **customer (cliente)**, o diagrama composto é **customer.composite\_diagram**.
  - c. Faça clique no componente de destino.
  - d. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome do componente de destino.
  - e. Faça clique no ícone de serviço associado ao componente.
  - f. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome do serviço.
2. Para configurar a importação IBM Business Process Manager para ligar ao serviço Open SCA, efectue os seguintes passos:
  - a. No IBM Integration Designer, navegue para o separador **Properties (Propriedades)** da importação SCA que pretende ligar ao serviço Open SCA.
  - b. No campo **Module name (Nome do módulo)**, introduza o nome do componente do passo 1d na página 68.
  - c. No campo **Export name (Nome da exportação)**, introduza o nome do serviço do passo 1f na página 68.
  - d. Guarde o seu trabalho ao premir Ctrl+S.

## Chamar módulos SCA de serviços Open SCA

Aplicações SCA abertas desenvolvidas num ambiente Rational Application Developer podem chamar aplicações SCA desenvolvidas com o IBM Integration Designer. Esta secção fornece um exemplo de chamada de um módulo SCA (através de uma associação de exportação SCA) de um serviço Open SCA.



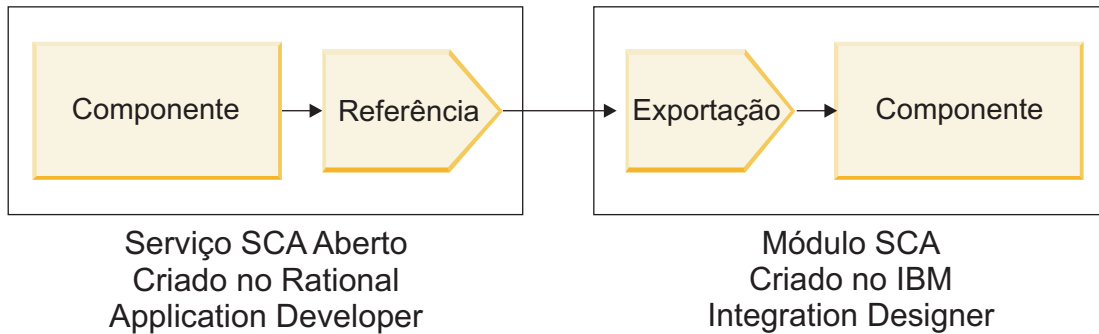


Figura 57. Componente de chamada de serviço Open SCA em módulo SCA

Para ligar a um componente SCA através de uma associação de referência Open SCA, fornece o nome do módulo e nome de exportação.

1. Para obter o nome do módulo e exportação de destino, execute os seguintes passos:
  - a. No IBM Integration Designer, abra o módulo no editor de montagem através de duplo clique no módulo.
  - b. Faça clique na exportação.
  - c. No campo **Name (Nome)** do separador **Properties (Propriedades)**, note o nome da exportação.
2. Configure a referência Open SCA que pretende ligar ao módulo IBM Business Process Manager e exporte:
  - a. No Rational Application Developer, abra o editor de composto através de duplo clique no diagrama de composto que contém o componente e serviço.
  - b. Faça clique no ícone de referência da referência do componente para apresentar as propriedades de referência no separador **Properties (Propriedades)**.
  - c. Faça clique no separador **Binding (Associação)** no lado esquerdo da página.
  - d. Faça clique em **Bindings (Associação)** e, em seguida, faça clique em **Add (Adicionar)**.
  - e. Selecione a associação SCA.
  - f. No campo **Uri**, introduza o nome do módulo IBM Business Process Manager, seguido de uma barra ("/"), seguida do nome de exportação (que determinou no passo 1c na página 69).
  - g. Faça clique em **OK**.
  - h. Guarde o seu trabalho ao premir Ctrl+S.

## Chamar serviços através de associações interoperáveis

As seguintes associações são suportadas para interoperacionalidade com um serviço Open SCA.

- Associação SCA

No IBM Business Process Manager, quando um módulo SCA chama um serviço Open SCA, são suportados os seguintes estilos de chamada para execução:

- Assíncrono (unidireccional)
- Síncrono (pedido/resposta)

A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).

Note que a associação SCA suporta propagação de contexto de transacção e segurança.

- Associação de serviço da Web (JAX-WS - Web service) com o protocolo SOAP1.1/HTTP ou o protocolo SOAP1.2/HTTP

A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).

Para além disso, são suportadas as seguintes qualidades de serviço:

- Web Services Atomic Transaction
- Web Services Security
- Associação EJB

Uma interface Java é utilizada para definir a interacção entre um módulo SCA e um serviço Open SCA quando a associação EJB é utilizada.

Note que a associação EJB suporta propagação de contexto de transacção e segurança.

- Associações JMS

A interface de importação SCA e a interface de serviço Open SCA têm de utilizar uma interface WSDL compatível com interoperacionalidade de serviços da Web (WS-I - web services interoperability).

São suportados os seguintes fornecedores JMS:

- WebSphere Platform Messaging (Associação JMS)
- WebSphere MQ (Associação JMS MQ)

**Nota:** Os gráficos comerciais não são interoperáveis em quaisquer associações SCA e, por esse motivo, não são suportados em interfaces utilizadas para interoperar com o WebSphere Application Server Feature Pack for Service Component Architecture.

## Tipos de associação

Utilize *associações* específicas de protocolo com importações e exportações para especificar os meios de transportar dados para dentro ou para fora de um módulo.

### Seleccionar associações apropriadas

Quando está a criar uma aplicação, necessita de saber como seleccionar a associação mais adequada às necessidades da aplicação.

As associações disponíveis em IBM Integration Designer proporcionam uma gama de opções. Nesta lista, pode determinar que tipo de associação poderá ser mais adequada às necessidades da aplicação.

Tenha em consideração uma associação de *Arquitetura do Componente de Serviço (SCA, Service Component Architecture)* quando estes factores forem aplicáveis:

- Todos os serviços estão contidos em módulos, ou seja, não existem serviços externos.
- Pretende separar a função em módulos SCA diferentes que interajam directamente uns com os outros.
- Os módulos estão estreitamente agrupados.

Tenha em consideração uma associação de *Serviço da web* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço externo através da Internet ou de fornecer um serviço através da Internet.
- Os serviços estão vagamente agrupados.
- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.
- O protocolo do serviço externo a que está a aceder ou do serviço que pretende fornecer é o SOAP/HTTP ou SOAP/JMS.

Tenha em consideração uma associação de *HTTP* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço externo através da Internet ou disponibilizar um serviço através da Internet e está a trabalhar com outros serviços da Web, tal como OBTER, COLOCAR, e ELIMINAR.
- Os serviços estão vagamente agrupados.
- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.

Tenha em consideração uma associação de *Enterprise JavaBeans (EJB)* quando estes factores forem aplicáveis:

- A associação destina-se a um serviço importado, que é um EJB ou que necessita de ser acedido por clientes EJB.
- O serviço importado é vagamente agrupado.
- As interações EJB declaráveis não são necessárias.
- Prefere-se a comunicação síncrona, ou seja, um pedido de um serviço pode esperar por uma resposta de outro.

Tenha em consideração uma associação de *Enterprise Information Systems (EIS)* quando estes factores forem aplicáveis:

- Necessita de aceder a um serviço num sistema EIS utilizando um adaptador de recurso.
- Prefere-se a transmissão de dados síncrona à assíncrona.

Tenha em consideração uma associação de *Java Message Service (JMS)* quando estes factores forem aplicáveis:

**Importante:** Existem vários tipos de associações JMS. Se pretender trocar mensagens SOAP com JMS, tenha em consideração a associação de serviço da web com o protocolo SOAP/JMS. Consulte “Associações do serviço da Web” na página 72.

- Necessita de aceder a um sistema de serviço de mensagens.
- Os serviços estão vagamente agrupados.
- Prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *Serviço de Mensagem Java (JMS) Genérico* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens que de um fornecedor que não pertença à IBM.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *Fila de Mensagens (MQ, Message Queue)* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens do WebSphere MQ e necessita de utilizar as funções nativas de MQ.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.

Tenha em consideração uma associação de *MQ JMS* quando estes factores forem aplicáveis:

- Necessita de aceder a um sistema de serviço de mensagens WebSphere MQ mas pode fazê-lo num contexto JMS, ou seja, o subconjunto de funções JMS é suficiente para a sua aplicação.
- Os serviços estão vagamente agrupados.
- A fiabilidade é mais importante que o desempenho, ou seja, prefere-se a transmissão de dados assíncrona à síncrona.

## **Associações SCA**

Uma associação da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) permite que um serviço comunique com outros serviços noutros módulos. Uma importação com associação SCA permite-lhe aceder a um serviço noutro módulo SCA. Uma exportação com uma associação SCA permite-lhe oferecer um serviço a outros módulos.

Utilize IBM Integration Designer para criar e configurar associações SACA para importações e exportações nos módulos SCA.

Se os módulos estiverem a ser executados no mesmo servidor ou implementados no mesmo conjunto de unidades, uma associação SCA é a associação mais fácil e rápida a utilizar.

Depois do módulo que contém a associação SCA ser implementado no servidor, pode utilizar a consola administrativa para ver informações sobre a associação ou, no caso de uma associação de importação, alterar as propriedades seleccionadas da associação.

## Associações do serviço da Web

Uma associação do serviço da Web é o meio de transmitir mensagens desde um componente de Arquitectura do Componente de Serviço (SCA) para um serviço da web (e vice-versa).

### Descrição geral de associações de serviço da Web: Advanced

Uma importação de serviço da web permite-lhe contactar um serviço da web externo através dos seus componentes de Arquitectura do Componente de Serviço (SCA). Uma associação de exportação de serviço da web permite-lhe expor os seus componentes SCA a clientes como serviços da web.

Com uma associação de serviço da web, acede a serviços externos utilizando mensagens SOAP interoperáveis e qualidades de serviço (QoS).

É possível utilizar o Integration Designer para criar e configurar associações de serviços da Web em importações e exportações em módulos SCA. Estão disponíveis os seguintes tipos de associações de serviços da Web:

- SOAP1.2/HTTP e SOAP1.1/HTTP

Estas associações são baseadas em Java API para Serviços da Web XML (JAX-WS), uma programação Java API para criar serviços da web.

- Utilize SOAP1.2/HTTP se o seu serviço da web for compatível com a especificação SOAP 1.2.
- Utilize SOAP1.1/HTTP se o seu serviço da web for compatível com a especificação SOAP 1.1.

**Importante:** Quando implementa uma aplicação com uma associação de serviço da web (JAX-WS), o servidor de destino não pode ter a opção **Start components as needed (Iniciar componentes consoante for necessário)** seleccionada. Consulte “Verificar a configuração do servidor” na página 81 para obter mais detalhes.

Quando selecciona uma destas associações, é possível enviar anexos com as suas mensagens SOAP.

As associações de serviço da web funcionam com mensagens SOAP padrão. Porém, utilizando uma das associações de serviço da web, é possível personalizar o modo como as mensagens SOAP são analisadas ou escritas. Por exemplo, é possível processar elementos não padronizados em mensagens SOAP ou aplicar processamento adicional à mensagem SOAP. Quando configura a associação, especifica uma rotina de tratamento de dados que executa este processamento na mensagem SOAP.

É possível utilizar conjuntos de políticas com uma associação de serviço da web (JAX-WS). Um conjunto de políticas é um conjunto de tipos de políticas, com cada um a fornecer uma qualidade de serviço (QoS). Por exemplo, o conjunto de políticas WSAddressing fornece um modo neutro de transporte para tratar, uniformemente, serviços da web e mensagens. Utiliza Integration Designer para seleccionar o conjunto de políticas para a associação.

**Nota:** Se deseja utilizar um conjunto de políticas de Linguagem de Marcação para Asserções de Segurança (SAML), deve executar algumas configurações adicionais, como descrito em “Importar definições de políticas SAML” na página 78.

- SOAP1.1/HTTP

Utilize esta associação se desejar criar serviços da web que utilizem uma mensagem de codificação SOAP baseada em Java API para RPC com base em XML (JAX-RPC).

- SOAP1.1/JMS

utilize esta associação para enviar ou receber mensagens SOAP utilizando um destino de Serviço de Mensagem (JMS) Java.

Independentemente do transporte (HTTP ou JMS) utilizado para transmitir a mensagem SOAP, as associações de serviço da web processam sempre interações de pedido/resposta de modo síncrono. O módulo que estabelece a chamada para execução no fornecedor de serviço é bloqueado até ser recebida uma resposta do fornecedor. Consulte “Chamada para execução síncrona” para obter mais informações sobre este estilo de chamada para execução.

**Importante:** As seguintes combinações de associações de serviço da web não podem ser utilizadas em exportações no mesmo módulo. Se necessitar de expor componentes utilizando mais do que uma destas associações de exportação, necessita de ter cada uma num módulo separado e, de seguida, ligar esses módulos aos seus componentes utilizando a associação SCA:

- SOAP 1.1/JMS e SOAP 1.1/HTTP utilizando JAX-RPC
- SOAP 1.1/HTTP utilizando JAX-RPC e SOAP 1.1/HTTP utilizando JAX-WS
- SOAP 1.1/HTTP utilizando JAX-RPC e SOAP 1.2/HTTP utilizando JAX-WS

Após o módulo SCA que contém a associação de serviço da web ser implementado no servidor, é possível utilizar a consola administrativa para ver informação sobre a associação ou para alterar as propriedades seleccionadas da associação.

**Nota:** Os serviços da Web permitem às aplicações interoperar utilizando descrições padrão de serviços e formatos padrão para as mensagens que trocam. Por exemplo, as associações de serviço da web de importação e exportação podem interoperar com serviços implementados utilizando Web Services Enhancements (WSE) Versão 3.5 e Windows Communication Foundation (WCF) Versão 3.5 para Microsoft .NET. Quando interoperar com tais serviços, deve assegurar que:

- O ficheiro Web Services Description Language (WSDL) utilizado para aceder à exportação de serviço da web inclui um valor de acção SOAP não vazio para cada operação na interface.
- O cliente do serviço da web define o cabeçalho SOAPAction ou o cabeçalho wsa:Action quando envia mensagens para uma exportação de serviço da web.

### Propagação de cabeçalho SOAP: Advanced

Quando processar mensagens SOAP, poderá necessitar de aceder a informações de certos cabeçalhos SOAP nas mensagens que são recebidas, assegurar que as mensagens com cabeçalhos SOAP são enviadas com valores específicos ou permitir aos cabeçalhos SOAP atravessarem um módulo.

Quando configura uma associação de serviço da web no Integration Designer, é possível indicar que deseja transportar cabeçalhos SOAP para serem propagados.

- Quando são recebidos pedidos numa exportação ou quando são recebidas respostas numa importação, a informação de cabeçalho SOAP pode ser acedida, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.
- Quando os pedidos são enviados de uma exportação ou quando as respostas são enviadas de uma importação, os cabeçalhos SOAP podem ser incluídos nessas mensagens.

O formato e presença dos cabeçalhos SOAP propagados podem ser afectados pelos conjuntos de políticas configuradas para a importação ou exportação, como explicado na Tabela 31 na página 75.

Para configurar a propagação de cabeçalhos SOAP para uma importação ou exportação, seleccione (da vista de Propriedades do Integration Designer) o separador **Propagate Protocol Header (Propagar cabeçalho de protocolo)** e seleccione as opções de que necessitar.

## Cabeçalho de endereçamento WS

O cabeçalho de endereçamento WS pode ser propagado pela associação de serviço da web (JAX-WS).

Quando propagar o cabeçalho de endereçamento WS, tenha em atenção a seguinte informação:

- Se activar a propagação para o cabeçalho de endereçamento WS, o cabeçalho será propagado para o módulo nas seguintes circunstâncias:
  - Quando os pedidos são recebidos numa exportação
  - Quando as respostas são recebidas numa importação
- O cabeçalho de endereçamento WS não é propagado para as mensagens de saída de IBM Business Process Manager (isto é, o cabeçalho não é propagado quando os pedidos são enviados de uma importação ou quando as respostas são enviadas da exportação).

## Cabeçalho de segurança WS

O cabeçalho de segurança WS pode ser propagado pela associação de serviço da web (JAX-WS) e pela associação de serviço da web (JAX-RPC).

A especificação da Segurança WS dos serviços da web descreve melhorias ao serviço de mensagens SOAP para proporcionar qualidade de protecção através de integridade de mensagem, confidencialidade de mensagem e autenticação de mensagem única. Estes mecanismos podem ser utilizados para acomodar uma ampla variedade de modelos de segurança e tecnologias de encriptação.

Quando propagar o cabeçalho Segurança WS, tenha em atenção a seguinte informação:

- Se activar a propagação para o cabeçalho de segurança WS, o cabeçalho será propagado através do módulo nas seguintes circunstâncias:
  - Quando os pedidos são recebidos numa exportação
  - Quando os pedidos são enviados de uma importação
  - Quando as respostas são recebidas numa importação
- O cabeçalho *não* será, por predefinição, propagado quando as respostas são enviadas da exportação. Porém, se definir a propriedade JVM **WSSECURITY.ECHO.ENABLED** para **true**, o cabeçalho será propagado quando as respostas forem enviadas da exportação. Neste caso, se o cabeçalho de segurança WS no caminho de pedido não for modificado, os cabeçalhos de segurança WS poderão ser automaticamente passados dos pedidos para as respostas.
- O formato exacto da mensagem SOAP enviada de uma importação para um pedido ou de uma exportação para uma resposta poderá não corresponder exactamente à mensagem SOAP que foi originalmente recebida. Por este motivo, assume-se que qualquer assinatura digital se torna inválida. Se for necessária uma assinatura digital nas mensagens que são enviadas, deve ser estabelecida utilizando o conjunto de políticas de segurança apropriado e os cabeçalhos de segurança WS relacionados com a assinatura digital nas mensagens recebidas devem ser removidos do módulo.

Para propagar o cabeçalho de segurança WS, deve incluir o esquema de segurança WS com o módulo de aplicação. Consulte “Incluindo o esquema de segurança WS num módulo de aplicação” na página 75 para obter o procedimento para incluir o esquema.

## Como os cabeçalhos são propagados

O modo como os cabeçalhos são propagados varia de acordo com a definição da política de segurança sobre associação de importação e exportação, como mostrado na Tabela 31 na página 75:



Tabela 60. Como são passados os cabeçalhos de segurança

	Associação de exportação sem política de segurança	Associação de exportação com política de segurança
Associação de importação sem política de segurança	Os cabeçalhos de segurança são passados tal como estão pelo módulo. Não são descodificados.  Os cabeçalhos são enviados para o exterior no mesmo formato em que são recebidos.  A assinatura digital pode tornar-se inválida.	Os cabeçalhos de segurança são descodificados e passados através do módulo com verificação e autenticação de assinatura.  Os cabeçalhos descodificados são enviados para o exterior.  A assinatura digital pode tornar-se inválida.
Associação de importação com política de segurança	Os cabeçalhos de segurança são passados tal como estão pelo módulo. Não são descodificados.  Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.	Os cabeçalhos de segurança são descodificados e passados através do módulo com verificação e autenticação de assinatura.  Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.

Configure os conjuntos de políticas apropriados nas associações de exportação e importação, pois isto isola o autor do pedido de serviço de alterações à configuração ou os requisitos de QoS do fornecedor de serviços. Ter cabeçalhos SOAP padrão visíveis num módulo pode, depois, ser utilizado para influenciar o processamento (por exemplo, registo e rastreio) no módulo. Propagar cabeçalhos SOAP através de um módulo a partir de uma mensagem recebida para uma mensagem enviada significa que as vantagens de isolamento do módulo são reduzidas.

Os cabeçalhos padrão, como cabeçalhos de segurança WS, não devem ser propagados num pedido para uma importação ou num pedido para uma exportação quando a importação ou exportação tiver um conjunto de políticas associado que normalmente resultaria na criação de tais cabeçalhos. Caso contrário, ocorre um erro devido à duplicação dos cabeçalhos. Em alternativa, os cabeçalhos devem ser explicitamente removidos ou a associação da importação ou exportação deve ser configurada para impedir a propagação de cabeçalhos de protocolo.

### Aceder a cabeçalhos SOAP

Quando é recebida uma mensagem, que contém cabeçalhos SOAP, de uma importação ou exportação de serviço da web, os cabeçalhos são colocados na secção de cabeçalhos do objecto de mensagem de serviço (SMO). É possível aceder à informação do cabeçalho, como descrito em “Aceder à informação do cabeçalho SOAP no SMO”.

### Incluindo o esquema de segurança WS num módulo de aplicação

O seguinte procedimento destaca os passos para incluir o esquema no módulo de aplicação:

- Se o computador no qual o Integration Designer estiver em execução tiver acesso à Internet, execute os passos seguintes:
  1. Na perspectiva de Business Integration, seleccione **Dependencies (Dependências)** para o seu projecto.
  2. Expanda **Predefined Resources (Recursos predefinidos)** e seleccione **WS-Security 1.0 schema files (ficheiros do esquema de segurança WS 1.0)** ou **WS-Security 1.1 schema files (ficheiros do esquema de segurança 1.1)** para importar o esquema para o seu módulo.
  3. Limpe e reconstrua o projecto.



- Se um computador no qual o Integration Designer estiver em execução não tiver acesso à Internet, pode transferir o esquema para um segundo computador que tenha acesso à Internet. Pode, de seguida, copiá-lo para o computador em que Integration Designer está a ser executado.
  1. A partir do computador com acesso à Internet, transfira o esquema remoto:
    - a. Faça clique em **File (Ficheiro) > Import (Importação) > Business Integration > WSDL and XSD (WSDL e XSD)**.
    - b. Seleccione **Remote WSDL (WSDL remota)** ou **XSD File (Ficheiro XSD)**.
    - c. Importe os seguintes esquemas:
      - `http://www.w3.org/2003/05/soap-envelope/`
      - `http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd`
      - `http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd`
  2. Copie os esquemas para o computador sem acesso à Internet.
  3. A partir do computador sem acesso à Internet, importe o esquema:
    - a. Faça clique em **File (Ficheiro) > Import (Importação) > Business Integration > WSDL and XSD (WSDL e XSD)**.
    - b. Seleccione **Local WSDL (WSDL local)** ou **XSD File (Ficheiro XSD)**.
  4. Altere as localizações do esquema para `oasis-wss-wssecurity-secext-1.1.xsd`:
    - a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd`.
    - b. Altere:
 

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='http://www.w3.org/2003/05/soap-envelope/' />
```

 para:
 

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='../w3/_2003/_05/soap_envelope.xsd' />
```
    - c. Altere:
 

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd' />
```

 para:
 

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd' />
```
  5. Altere a localização do esquema para `oasis-200401-wss-wssecurity-secext-1.0.xsd`:
    - a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-200401-wss-wssecurity-secext-1.0.xsd`.
    - b. Altere:
 

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd"/>
```

 para:
 

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="../w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd"/>
```
  6. Limpe e reconstrua o projecto.

## Propagação de cabeçalho de transporte: Advanced

Quando processar mensagens SOAP, poderá necessitar de aceder a informações de certos cabeçalhos de transporte nas mensagens que são recebidas, assegurar que as mensagens com cabeçalhos de transporte são enviadas com valores específicos ou permitir aos cabeçalhos de transporte atravessarem um módulo.

Quando configura uma associação de serviço da web no Integration Designer, é possível indicar que deseja transportar cabeçalhos para serem propagados.

- Quando são recebidos pedidos numa exportação ou quando são recebidas respostas numa importação, a informação de cabeçalho de transporte pode ser acedida, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que estes cabeçalhos sejam modificados.
- Quando as respostas são enviadas de uma exportação ou quando os pedidos são enviados de uma importação, os cabeçalhos de transporte podem ser incluídos nessas mensagens.

### Especificar a propagação de cabeçalhos

Para configurar a propagação de cabeçalhos de transporte para uma importação ou exportação, execute os seguintes passos:

1. A partir da vista de Propriedades do Integration Designer, seleccione **Binding (Associação) > Propagation (Propagação)**.
2. Defina a opção de propagação de cabeçalho de transporte de que necessita.

**Nota:** A propagação de cabeçalho de transporte está desactivada por predefinição e apenas pode ser implementada num ambiente em tempo de execução de Versão 7.0.0.3 (ou posterior). Tenha também em atenção que, para a Versão 7.0.0.3, a propagação de cabeçalho de transporte está limitada a cabeçalhos de transporte de HTTP apenas.

Se activar a propagação de cabeçalhos de transporte, os cabeçalhos serão propagados através de um módulo de mensagens recebidas e, se não remover, explicitamente, os cabeçalhos, estes serão utilizados em chamadas para execução subsequentes na mesma sessão.

**Nota:** Os cabeçalhos de transporte não podem ser propagados quando estiver a utilizar a associação de serviço da web (JAX-RPC).

### Aceder à informação de cabeçalho

Quando a propagação de cabeçalho de transporte é activada para mensagens recebidas, todos os cabeçalhos de transporte (incluindo cabeçalhos definidos por cliente) são visíveis no objecto de mensagem de serviço (SMO). Pode definir os cabeçalhos para valores diferentes ou criar novos. Repare, porém, que não existe verificação ou validação dos valores que definir e quaisquer cabeçalhos incorrectos poderão causar problemas ao tempo de execução do serviço da web.

Tenha em conta a seguinte informação sobre a definição de cabeçalhos de HTTP:

- Quaisquer alterações aos cabeçalhos que estejam reservados para o motor do serviço da web não serão respeitadas na mensagem de saída. Por exemplo, os cabeçalhos de versão ou método HTTP, tipo de conteúdo, comprimento de conteúdo e SOAPAction estão reservados para o motor de serviço da web.
- Se o valor de cabeçalho for um número, o número (em vez de cadeia) deve ser directamente definido. Por exemplo, utilize **Max-Forwards = 5** (em vez de **Max-Forwards = Max-Forwards: 5**) e **Age = 300** (em vez de **Age = Age: 300**).
- Se a mensagem de pedido for inferior a 32 KB de tamanho, o motor de serviço da web retira o cabeçalho Transferência-Codificação e, em alternativa, define o cabeçalho Conteúdo-Comprimento para o tamanho fixo da mensagem.
- O idioma do conteúdo é repostado por WAS.channel.http no caminho de resposta.
- Uma definição inválida para Actualização resulta num erro 500.
- Os seguintes cabeçalhos acrescentam o valor reservado pelo motor de serviço da web às definições do cliente:
  - Agente do utilizador
  - Controlo de Cache
  - Pragma
  - Aceitar

## – Ligação

É possível aceder à informação do cabeçalho de uma das seguintes maneiras:

- Utilizar uma primitiva de mediação para aceder a estruturas SMO  
Consulte as ligações “Informações Relacionadas” para encontrar informação sobre como utilizar primitivas de mediação.
- Utilizar o SPI de serviço de contexto

O seguinte código exemplo lê os cabeçalhos de transporte de HTTP do serviço de contexto:

```
HeadersType headerType = ContextService.INSTANCE.getHeaders();
HTTPHeaderType httpHeaderType = headerType.getHTTPHeader();
List HTTPHeader httpHeaders = httpHeaderType.getHeader();
if(httpHeaders!=null){
    for(HTTPHeader httpHeader: httpHeaders){
        String httpHeadername = httpHeader.getName();
        String httpHeaderValue = httpHeader.getValue();
    }
}
List PropertyType properties = headerType.getProperties();
if(properties!=null){
    for(PropertyType property: properties){
        String propertyName = property.getName();
        String propertyValue = property.getValue().toString();
    }
}
```

## Resolução de problemas

Se tiver problemas ao enviar os cabeçalhos revistos, pode interceptar a mensagem TCP/IP ao utilizar ferramentas como o Monitor TCP/IP no Integration Designer. Acede-se ao Monitor TCP/IP ao seleccionar **Run/Debug (Executar/Depurar) > TCP/IP Monitor (Monitor TCP/IP)** da página de Preferências.

Também pode ver os valores do cabeçalho usando o rastreio do motor JAX-WS: **org.apache.axis2.\*=all: com.ibm.ws.websvcs.\*=all:**

## Trabalhar com associações do serviço da web (JAX-WS): **Advanced**

Quando utilizar associações do serviço da web (JAX-WS) com as suas aplicações, pode adicionar uma qualidade de serviço (QoS) de Linguagem de Marcação para Asserções de Segurança (SAML) à associação. Deve, primeiro, utilizar a consola administrativa para importar o conjunto de políticas. Também pode utilizar a consola administrativa para se certificar de que o servidor está devidamente configurado para utilização com a associação do serviço da web (JAX-WS).

## Importar definições de políticas SAML: **Advanced**

A Security Assertion Markup Language (SAML) é um OASIS padrão, com base XML, para permutar informação relativa a identidade do utilizador e atributos de segurança. Quando configura uma ligação de serviço da Web (JAX-WS) no Integration Designer, pode especificar uma definição de políticas SAML. Primeiro, utilizar a consola administrativa de IBM Business Process Manager para disponibilizar as definições de política SAML, de modo a que possam ser importadas para Integration Designer.

As definições de política SAML estão, geralmente, localizadas no directório de configuração de perfis:

*profile\_root/config/templates/PolicySets*

Antes de iniciar este procedimento, verificar se os seguintes directórios (que contêm as definições de políticas) estão localizados no directório de configuração de perfis:

- predefinição SAML11 Bearer WSHTTPS
- predefinição SAML20 Bearer WSHTTPS
- predefinição SAML11 Bearer WSSecurity
- predefinição SAML20 Bearer WSSecurity
- predefinição SAML11 HoK Public WSSecurity
- predefinição SAML20 HoK Public WSSecurity
- predefinição SAML11 HoK Symmetric WSSecurity
- predefinição SAML20 HoK Symmetric WSSecurity
- predefinição Username WSHTTPS

Caso os directórios não estejam no directório de configuração de perfis, deverão ser copiados para esse directório, a partir da seguinte localização:

`app_server_root/profileTemplates/default/documents/config/templates/PolicySets`

Para importar as definições de política para a consola administrativa, seleccione as que pretende tornar disponíveis no Integration Designer e, em seguida, guarde um ficheiro .zip para cada uma dessas definições de política numa localização que seja acessível pelo Integration Designer.

1. Importar as definições de política, seguindo os passos que se seguem:
  - a. A partir da consola administrativa, fazer clique em **Services (Serviços) > Policy Sets (Definições de Política) > Application policy sets (Definições de política de Aplicação)**.
  - b. Fazer clique em **Import (Importar) > from Default Repository (do Repositório Predefinido)**.
  - c. Seleccionar as definições de política SAML predefinidas, e fazer clique em **OK (OK)**.
2. Exportar as definições de política para que possam ser utilizadas por Integration Designer:
  - a. A partir da página de definições de política de Aplicação, seleccionar a definição de política SAML para exportar, e fazer clique em **Export (Exportar)**.

**Nota:** Caso a página de definições de política de Aplicação não esteja a ser exibida, fazer clique em **Services (Serviços) > Policy Sets (Definições de Política) > Application Policy Sets (Definições de Política de Aplicação)** a partir da consola administrativa.

- b. Na página seguinte, fazer clique na ligação do ficheiro .ZIP para a definição de política.
- c. Na janela File Download (Transferir Ficheiro), fazer clique em **Save (Guardar)**, e indicar a localização que é acessível a partir de Integration Designer.
- d. Fazer clique em **Back (Retroceder)**.
- e. Concluir os passos 2a na página 79 concluídos 2d na página 79 para cada definição de política que deseja exportar.

As definições de política SAML são guardadas nos ficheiros .ZIP e estão prontas para serem importadas para Integration Designer.

Importar as definições de política para Integration Designer, conforme descrito no tópico “Definições de política”.

*Chamar serviços da Web que exigem autenticação base de HTTP:* **Advanced**

A autenticação base de HTTP utiliza um nome de utilizador e uma palavra-passe para autenticar um cliente do serviço para um destino final seguro. Poderá definir uma autenticação base de HTTP, enviando ou recebendo pedidos de serviços da Web.

O utilizador define a autenticação básica de HTTP para receber pedidos de serviço da Web através da configuração de associações de exportação de Serviços da Web Java API para XML (JAX-WS), como descrito em Criar e atribuir funções de segurança a exportações de serviços da Web.

A autenticação base de HTTP pode ser activada para pedidos de serviço da Web, que são enviados por uma associação de importações JWX-WS, num destes dois modos:

- Quando configurar a associação de importações de um módulo SCA, poderá seleccionar o conjunto de política de autenticação HTTP que foi fornecida, chamada BBPMHTTPBasicAuthentication (que é fornecida com a associação de importação do serviço da Web (JAX-WS) ou qualquer outro conjunto de política que inclua a política HTTPTransport.
- Aquando da construção do módulo SCA, poderá utilizar capacidades de mediação de fluxo para criar, de forma dinâmica, um novo cabeçalho de autenticação HTTP, e especificar a informação relativa ao nome de utilizador e à palavra-passe no cabeçalho.

**Nota:** O conjunto de políticas tem preferência sobre o valor especificado no cabeçalho. Caso deseje utilizar o conjunto de valores no cabeçalho de autenticação HTTP no tempo de execução, não deverá anexar um conjunto de políticas que inclua a política HTTPTransport. Não utilizar, especificamente, o conjunto de políticas BPMHTTPBasicAuthentication predefinido e, se definiu um conjunto de políticas, certifique-se de que ele não contém a política HTTPTransport.

Para obter mais informações sobre definições e associações de políticas de serviços da Web e sobre como estas são utilizadas, consulte Definições de políticas de serviços da Web do Information Center do WebSphere Application Server.

- Para utilizar o conjunto de políticas fornecidas, executar os passos que se seguem:
  1. Opcional: Na consola administrativa, crie uma associação de políticas gerais do cliente ou edite uma associação existente que inclui a política HTTPTransport com os valores de ID do utilizador e palavra-passe necessários.
  2. Em IBM Integration Designer, criar uma associação de importações de um serviço da Web (JAX-WS), e anexar o conjunto de políticas BPMHTTPBasicAuthentication.
  3. Executar *um* dos seguintes passos:
    - Em IBM Integration Designer, nas propriedades de associações de importações dos serviços da Web (JAX-WS), especificar o nome de uma associação de políticas de clientes gerais, já existente, que inclua a política HTTPTransport.
    - Após a implementação do módulo SCA, utilize a consola administrativa para seleccionar uma associação de políticas de clientes existente ou para criar uma associação de políticas de clientes nova e, em seguida, associe-a à associação de importação.
  4. Opcional: Na consola administrativa do Process Server, editar a associação de definições das políticas seleccionadas, para especificar o ID e palavra-passe exigidos.
- Para especificar o nome de utilizador e a palavra-passe no cabeçalho da autenticação HTTP, executar um dos seguintes passos:
  - Utilizar a primitiva de mediação HTTP Header Setter IBM Integration Designer para criar o cabeçalho de autenticação HTTP, e especifique o nome de utilizador e a palavra-passe.
  - Caso seja exigida uma lógica adicional, utilizar um código Java numa primitiva de mediação, personalizada (conforme exibido no exemplo seguinte) para:
    1. Criar um cabeçalho de autenticação HTTP.
    2. Especificar a informação relativa ao nome de utilizador e à palavra-passe.
    3. Adicionar o novo cabeçalho de autenticação HTTP ao HTTPControl.
    4. Retroceder o HTTPControl actualizado para o Context service (Serviço de contexto).

```
//Get the HeaderInfoType from contextService
ContextService contextService = (ContextService) ServiceManager.INSTANCE
    .locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
```

```

if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
//Get the HTTP header and HTTP Control from HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
//Create new HTTPAuthentication and set the HTTPCredentials
HTTPAuthentication authorization = factory.createHTTPAuthentication();
HTTPCredentials credentials = factory.createHTTPCredentials();
authorization.setAuthenticationType(HTTPAuthenticationType.BASIC_LITERAL);
credentials.setUserId("USERNAME");
credentials.setPassword("PASSWORD");
authorization.setCredentials(credentials);
cp.setAuthentication(authorization);
httpHeaderType.setControl(cp);
// Set header info back to the current execution context.
contextService.setHeaderInfo(headers);

```

Verificar a configuração do servidor: **Advanced**

Quando implementa uma aplicação com uma associação de serviço da Web (JAX-WS), tem de se certificar que o servidor no qual a aplicação está implementada não tem a opção **Start components as needed (Iniciar componentes como necessário)** seleccionada.

Poderá verificar se esta opção está seleccionada, executando os seguintes passos, a partir da consola administrativa:

1. Fazer clique em **Servers (Servidores) > Server types (Tipos de servidor) > WebSphere application servers (Servidores WebSphere Application)**.
2. Fazer clique no nome do servidor.
3. A partir do separador Configuração, determinar se foram seleccionados os **Start components as needed (Componentes de início necessários)**.
4. Executar um dos seguintes passos:
  - Caso estejam seleccionados os **Start components as needed (Componentes de início necessários)**, remover a marca, e fazer clique em **Apply (Aplicar)**.
  - Caso não estejam seleccionados os **Start components as needed (Componentes de início necessários)** fazer clique em **Cancel (Cancelar)**.

Anexos em mensagens SOAP: **Advanced**

É possível enviar e receber mensagens SOAP que incluam dados binários (como ficheiros PDF ou imagens JPEG) como anexos. Os anexos podem ser *referenciados* (ou seja, explicitamente representados como componentes da mensagem na interface de serviço) ou *não referenciados* (situação na qual ou números e tipos arbitrários dos anexos podem ser incluídos).

Um anexo referenciado pode ser representado numa das seguintes maneiras:

- Os anexos MTOM utilizam a codificação especificada do Mecanismo de Optimização de Transmissão de Mensagem SOAP (<http://www.w3.org/TR/soap12-mtom/>). Os anexos MTOM são activados através de uma opção de configuração nas associações de importação e exportação e são o modo recomendado para codificar anexos para novas aplicações.
- Como um elemento `ws:swaRef-typed` no esquema de mensagem



Os anexos definidos utilizando wsi:swaRef type estão em conformidade com a Web Services Interoperability Organization (WS-I) *Perfil de Anexos Versão 1.0* (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como os elementos da mensagem se relacionam com os componentes MIME.

- Como um componente da mensagem de nível superior, utilizando um tipo de esquema binário  
Os anexos representados como componentes de nível superior estão em conformidade com a especificação das *Mensagens SOAP com Anexos* (<http://www.w3.org/TR/SOAP-attachments> ).  
Os anexos representados como componentes da mensagem de nível superior também podem ser configurados para assegurar que as mensagens e documento WSDL produzidos pela associação estão em conformidade com a *WS-I Perfil de Anexos Versão 1.0* e *WS-I Perfil Básico Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Um anexo não referenciado é transportado numa mensagem SOAP sem qualquer representação no esquema da mensagem.

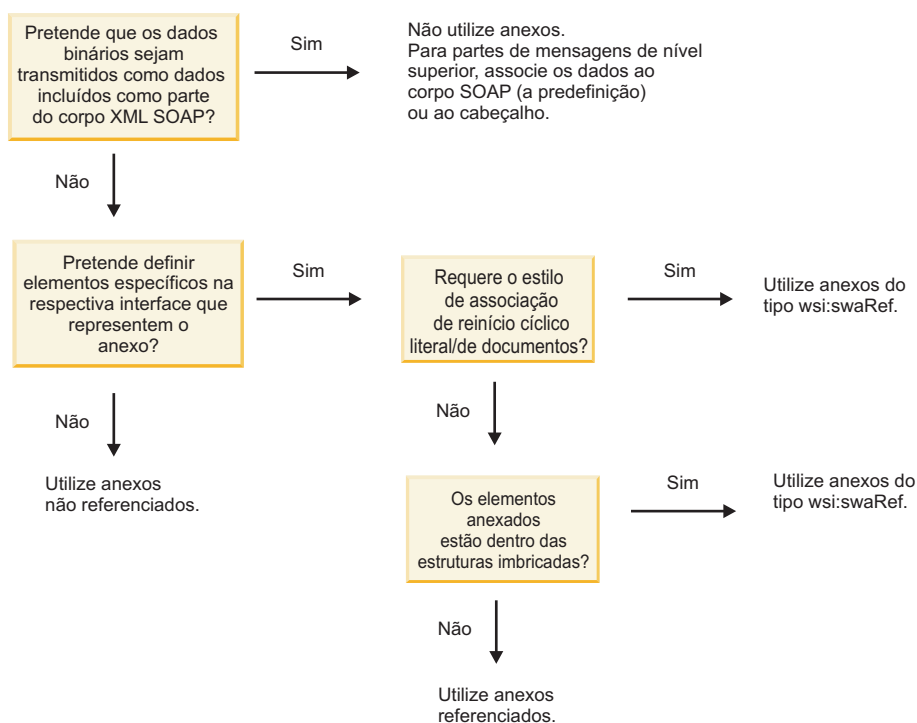
Em todos os casos, excepto anexos MTOM, a associação SOAP WSDL deve incluir uma associação MIME para anexos a serem utilizados, e o tamanho máximo dos anexos não deve ultrapassar 20 MB.

**Nota:** Para enviar ou receber mensagens SOAP com anexos, deve utilizar uma das associações de serviço da web baseadas em Java API para Serviços da Web XML (JAX-WS).

Como seleccionar o estilo de anexo apropriado: **Advanced**

Quando conceber um serviço que inclua dados binários, tenha em consideração o modo como esses dados binários são transportados nas mensagens SOAP que são enviadas e recebidas pelo serviço.

O Mecanismo de Optimização de Transmissão de Mensagens (MTOM) deve ser utilizado para anexos, caso a aplicação do serviço da Web ligada o suporte. Caso contrário, o seguinte diagrama mostra como são seleccionados outros estilos de anexo. Utilize o seguinte conjunto de questões para determinar o estilo de anexo apropriado:





*Anexos MTOM: partes de mensagens de nível superior:*

É possível enviar e receber mensagens de serviço da Web que incluem anexos Mecanismo de Optimização de Transmissão de Mensagens (MTOM, Message Transmission Optimization Mechanism) SOAP. Numa mensagem SOAP de várias partes MIME, o corpo SOAP é a primeira parte da mensagem, vindo o anexo ou anexos nas partes seguintes.

Ao enviar ou receber um anexo referenciado numa mensagem SOAP, os dados binários que compõem o anexo (que são, muitas vezes, extensos) são retidos separadamente do corpo da mensagem SOAP, para não terem de ser analisados como XML. Isto resulta num processamento mais eficiente do que se os dados binários fossem mantidos num elemento XML.

Segue-se um exemplo de uma mensagem SOAP MTOM:

```
... other transport headers (outros cabeçalhos de transporte)...
Content-Type: multipart/related; boundary=MIMEboundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812;
type="application/xop+xml"; start="
<0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>; start-info="text/xml"; charset=UTF-8

--MIMEboundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: application/xop+xml; charset=UTF-8; type="text/xml";
content-transfer-encoding: binary
content-id:
  <0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>

<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
      <sendImage xmlns="http://org.apache.axis2/jaxws/sample/mtom">
        <input>
          <imageData><xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
            href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/></imageData>
        </input>
      </sendImage>
    </soapenv:Body>
  </soapenv:Envelope>
--MIMEboundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: text/plain
content-transfer-encoding: binary
content-id:
  <1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org>

... binary data goes here ...
--MIMEboundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--
```

Repare que no exemplo MTOM, o tipo conteúdo para o envelope SOAP é **application/xop+xml** e que os dados binários são substituídos por um elemento **xop:Include** como se demonstra abaixo:

```
<xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/>
```

### Processamento interno de anexos referenciados

Quando um cliente passa uma mensagem SOAP com um anexo ao componente Service Component Architecture (SCA), a associação da exportação do serviço da web (JAX-WS) remove o anexo em primeiro lugar. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por fim, a associação define o binário do anexo no objecto empresarial.

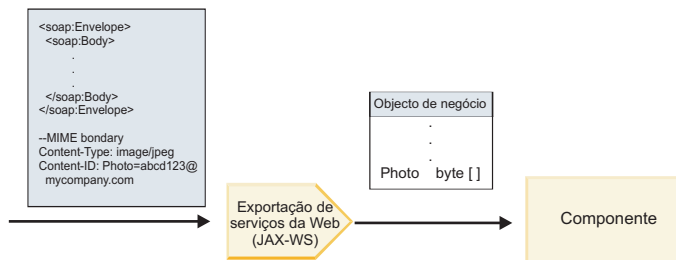


Figura 58. Forma como a associação da exportação do serviço da web (JAX-WS) processa uma mensagem SOAP com um anexo referenciado

### Atributos do anexo MTOM

- O MTOM suporta elementos de anexo dentro de estruturas imbricadas.
- O MTOM só está disponível para o tipo base64Binary.
- O MTOM suporta elementos de anexo dentro de estruturas imbricadas, o que significa que os **bodyPath** para anexos MTOM são as localizações **xpath** do elemento onde o anexo MTOM é retido. A lógica computacional do **bodyPath** segue estritamente o esquema para gerar a localização **xpath**, tal como demonstrado nos exemplos abaixo:
  - Para um tipo não matricial (**maxOccurs** é 1): /sendImage/input/imageData
  - Para um tipo matricial (**maxOccurs** > 1): /sendImage/input/imageData[1]
- Os tipos de anexo misturados não são suportados, o que significa que, se o MTOM for activado na associação de importação, será gerado o anexo MTOM. Se o MTOM for desactivado ou se o valor de configuração MTOM for mantido no valor predefinido na associação de exportação, a mensagem MTOM de entrada não será suportada.

Anexos referenciados: elementos do tipo swaRef: **Advanced**

É possível enviar e receber mensagens SOAP que incluam anexos representados na interface de serviço como elementos do tipo swaRef.

É definido um elemento do tipo swaRef no *Perfil de Anexos Versão 1.0* de Web Services Interoperability Organization (WS-I) (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como os elementos da mensagem estão relacionados com componentes MIME.

Na mensagem SOAP, o corpo SOAP contém um elemento do tipo swaRef que identifica o identificador de conteúdo do anexo.

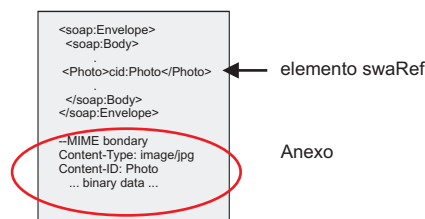


Figura 59. Uma mensagem SOAP com um elemento swaRef

A WSDL para esta mensagem SOAP contém um elemento do tipo swaRef, dentro de um componente da mensagem, que identifica o anexo.

```
<element name="sendPhoto">
  <complexType>
    <sequence>
```

```

        <element name="Photo" type="ws:swaRef"/>
    </sequence>
</complexType>
</element>

```

A WSDL também deve conter uma associação MIME que indica quais as mensagens multi-parte MIME que devem ser utilizadas.

**Nota:** A WSDL *não* inclui uma associação MIME para o elemento da mensagem do tipo swaRef específico, pois as associações MIME apenas se aplicam a componentes da mensagem de nível superior.

Os anexos representados como elementos do tipo swaRef apenas podem ser propagados através de componentes de fluxo de mediação. Caso um anexo deva ser acessado por ou propagado para outro tipo de componente, utilize um componente de fluxo de mediação para mover o anexo para uma localização que seja acessível por esse componente.

### Processamento interno de anexos

Utiliza Integration Designer para configurar uma associação de exportação para receber o anexo. Cria um módulo e a respectiva interface associada e operações, incluindo um elemento do tipo swaRef. Pode, em seguida, criar uma associação de serviço da web (JAX-WS).

**Nota:** Consulte o tópico “Working with attachments (Trabalhar com anexos)” no Information Center do Integration Designer para obter informações mais detalhadas.

Quando um cliente passa uma mensagem SOAP com um anexo SwaRef para um componente de Arquitectura do Componente de Serviço (SCA), a associação de exportação do serviço da web (JAX-WS) remove, primeiro, o anexo. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por último, a associação coloca o identificador de conteúdo no anexo no objecto de negócio.

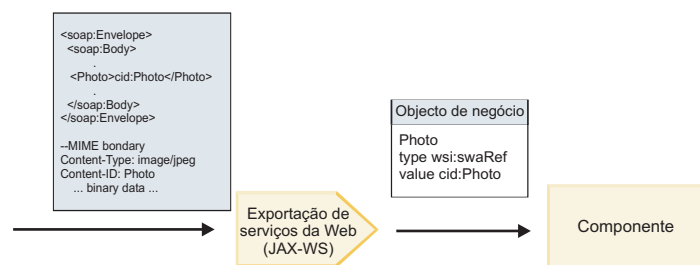


Figura 60. Como a associação de exportação do serviço da web (JAX-WS) processa uma mensagem SOAP com um anexo swaRef

### Aceder a metadados do anexo num componente de fluxo de mediação

Como apresentado na Figura 16 na página 86, quando os anexos swaRef são acessados por componentes, o identificador do conteúdo do anexo aparece como um elemento do tipo swaRef.

Cada anexo de uma mensagem SOAP também tem um elemento de **anexos** correspondente no SMO. Quando utilizar o tipo swaRef de WS-I, o elemento dos **anexos** inclui o tipo de conteúdo do anexo e identificador do conteúdo e também os dados binários concretos do anexo.

Para obter o valor de um anexo swaRef, é, assim, necessário obter o valor do elemento do tipo swaRef e, de seguida, localizar o elemento dos **anexos** com o valor correspondente do **identificador de conteúdo**. Tenha em consideração que o valor do **identificador de conteúdo** tem, geralmente, o prefixo **cid:** removido do valor de swaRef.

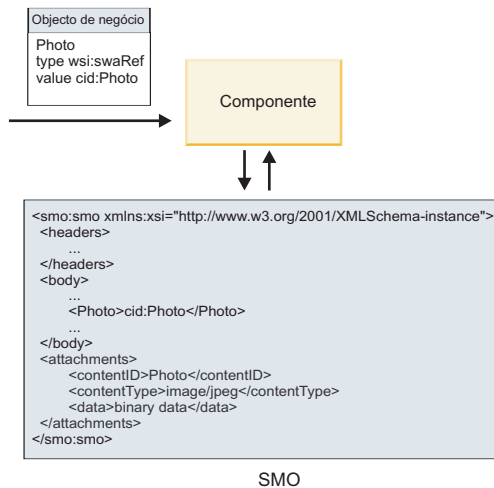


Figura 61. Como os anexos swaRef aparecem no SMO

### Processamento de saída

Utiliza Integration Designer para configurar uma associação de importação de serviço da web (JAX-WS) para chamar um serviço da web externo. A associação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define o anexo que será passado para o serviço da web.

Quando uma mensagem SCA é recebida por uma associação de importação de serviço da web (JAX-WS), os elementos de tipo swaRef são enviados como anexos caso a importação esteja ligada a um componente de fluxo de mediação e o elemento de tipo swaRef tenha um elemento do **anexo** correspondente.

Para processamento de saída, os elementos de tipo swaRef são sempre enviados com os respectivos valores do identificador de conteúdo; porém, o módulo de mediação deve assegurar que existe um elemento dos **anexos** correspondente em concordância com o valor do **identificador de conteúdo**.

**Nota:** Para ser compatível com o Perfil de Anexos de WS-I, o valor do **identificador de conteúdo** deve acompanhar a "codificação do componente do identificador de conteúdo", como descrito na secção 3.8 do *Perfil de Anexos 1.0 de WS-I*.

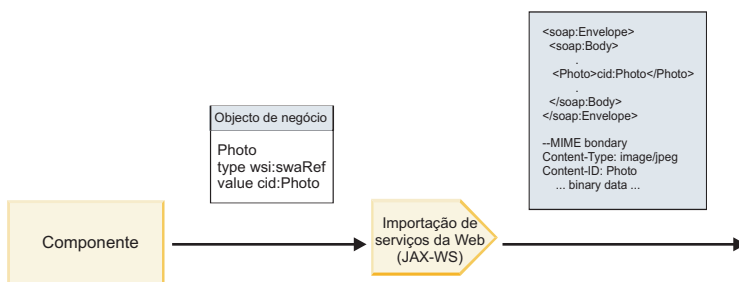


Figura 62. Como a associação de importação do serviço da web (JAX-WS) cria uma mensagem SOAP com um anexo swaRef

### Definir metadados de anexo num componente de fluxo de mediação

Se, no SMO, existir um valor de elemento do tipo swaRef e um elemento dos **anexos**, a associação prepara a mensagem SOAP (com o anexo) e envia-a para um destinatário.

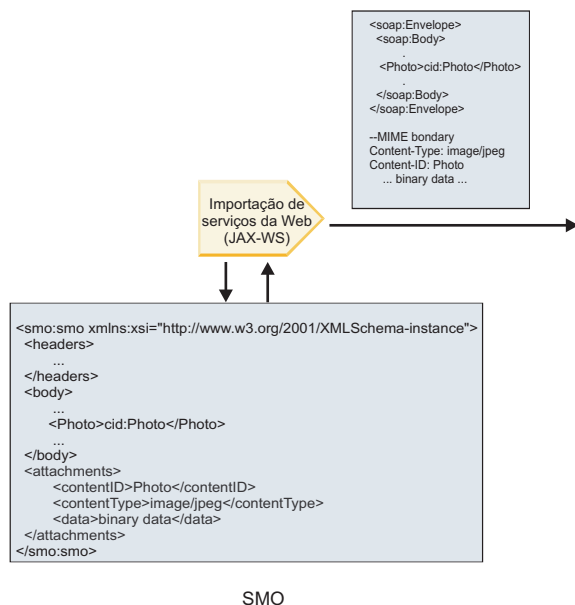


Figura 63. Como um anexo swaRef no SMO é acedido para criar a mensagem SOAP

O elemento dos **anexos** está presente no SMO apenas se um componente de fluxo de mediação estiver directamente ligado à importação ou exportação; não é passado por outros tipos de componente. Se os valores forem necessários num módulo que contenha outros tipos de componente, deve ser utilizado um componente de fluxo de mediação para copiar os valores para uma localização onde possam ser acedidos no módulo e utilizado um componente de fluxo de mediação para definir os valores correctos antes de uma chamada para execução através de uma importação de serviço da web.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.
- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

Anexos referenciados: componentes de mensagem de nível superior: **Advanced**

É possível enviar e receber mensagens SOAP que incluam anexos binários que sejam declarados como componentes na sua interface de serviço.

Numa mensagem SOAP de multi-parte MIME, o corpo SOAP é o primeiro componente da mensagem e o anexo ou anexos estão em componentes subsequentes.

Qual a vantagem de enviar ou receber um anexo referenciado numa mensagem SOAP? Os dados binários que constituem o anexo (que é, frequentemente, bastante grande) são mantidos em separado do corpo da mensagem SOAP para que não tenham de ser analisados como XML. Isto resulta num processamento mais eficiente do que se os dados binários fossem mantidos num elemento XML.

## Tipos de mensagens SOAP com anexos referenciados

A partir da Versão 7.0.0.3 de IBM Business Process Manager, pode seleccionar o modo como a mensagem SOAP é criada:

- **Mensagens compatíveis com WS-I**

O tempo de execução pode criar mensagens SOAP compatíveis com *Perfil de Anexos Versão 1.0* de WS-I e *Perfil Básico Versão 1.1* de WS-I. Numa mensagem SOAP compatível com estes perfis, apenas um componente de mensagem está associado ao corpo SOAP; para os associados como anexos, a codificação do componente do identificador de conteúdo (como descrito no *Perfil de Anexos Versão 1.0* de WS-I) é utilizada para relacionar o anexo com o componente da mensagem.

- **Mensagens não compatíveis com WS-I**

O tempo de execução pode criar mensagens SOAP não compatíveis com os perfis de WS-I mas que sejam compatíveis com as mensagens criadas na Versão 7.0 ou 7.0.0.2 de IBM Business Process Manager. As mensagens SOAP utilizam elementos de nível superior com o nome do componente da mensagem com um atributo **href** que retém o **content-id** do anexo, mas a codificação do elemento do identificador de conteúdo (como descrito no *Perfil de Anexos Versão 1.0* de WS-I) não é utilizada.

## Seleccionar compatibilidade de WS-I para exportações de serviço da web

Utiliza Integration Designer para configurar uma associação de exportação. Cria um módulo e respectiva interface e operações associadas. Pode, em seguida, criar uma associação de serviço da web (JAX-WS). A página Anexos referenciados apresenta todos os elementos binários da operação criada, seleccionando-se quais os componentes que serão anexos. Especifica, de seguida, na página Especificar a compatibilidade de WS-I AP 1.0 de Integration Designer, uma das seguintes escolhas:

- **Use WS-I AP 1.0 compliant SOAP message (Utilizar mensagem SOAP compatível com WS-I AP 1.0)**

Se seleccionar esta opção, também pode especificar qual o componente da mensagem que deve ser associado ao corpo SOAP.

**Nota:** Esta opção apenas pode ser utilizada quando o ficheiro WSDL correspondente também é compatível com WS-I.

Um ficheiro WSDL que seja criado por Integration Designer de Versão 7.0.0.3 é compatível com WS-I. Porém, se importar um ficheiro WSDL que não seja compatível com WS-I, não é possível seleccionar esta opção.

- **Use non WS-I AP 1.0 compliant SOAP message (Utilize mensagem SOAP não compatível com WS-I AP 1.0)**

Se seleccionar esta opção, a qual é a predefinição, o primeiro componente da mensagem é associado ao corpo SOAP.

**Nota:** Apenas os componentes de mensagem de nível superior (ou seja, elementos definidos no tipo de porta WSDL como componentes na mensagem de entrada ou de saída) que têm um tipo binário (base64Binary ou hexBinary) podem ser enviados ou recebidos como anexos referenciados. Consulte o tópico “Working with attachments (Trabalhar com anexos)” no Information Center do Integration Designer para obter informações mais detalhadas.

Para mensagens compatíveis com WS-I, o identificador de conteúdo que é criado na mensagem SOAP é uma concatenação dos seguintes elementos:

- O valor do atributo de **nome** do elemento **wSDL:part** referenciado por **mime:content**
- O carácter =
- Um valor globalmente único, como um UUID
- O carácter @
- Um nome de domínio válido

## Processamento interno de anexos referenciados

Quando um cliente passa uma mensagem SOAP com um anexo ao componente Service Component Architecture (SCA), a associação da exportação do serviço da web (JAX-WS) remove o anexo em primeiro lugar. Analisa, de seguida, o componente SOAP da mensagem e cria um objecto de negócio. Por fim, a associação define o binário do anexo no objecto empresarial.

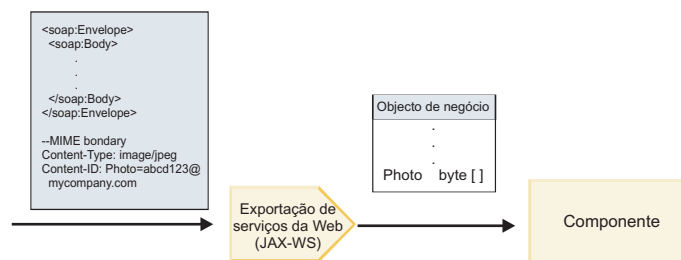


Figura 64. Como a associação de exportação do serviço da web (JAX-WS) processa uma mensagem SOAP compatível com WS-I com um anexo referenciado

## Aceder a metadados do anexo num componente de fluxo de mediação

Como mostrado na Figura 19 na página 89, quando os anexos referenciados são acedidos por componentes, os dados do anexo aparecem como uma matriz de bytes.

Cada anexo referenciado de uma mensagem SOAP também tem um elemento de **anexos** correspondente no SMO. O elemento dos **anexos** inclui o tipo de conteúdo do anexo e o caminho para o elemento do corpo da mensagem, onde se encontra o anexo.

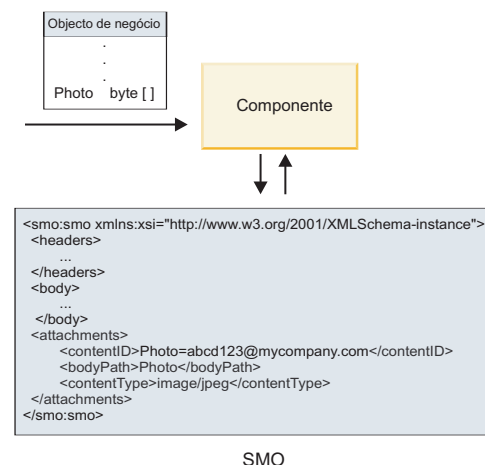


Figura 65. Como os anexos referenciados aparecem no SMO

**Importante:** O caminho para o elemento do corpo da mensagem não é automaticamente actualizado caso a mensagem seja transformada e o anexo deslocado. É possível utilizar um fluxo de mediação para actualizar o elemento dos **anexos** com um novo caminho (por exemplo, como parte da transformação ou utilizando um definidor de elemento da mensagem separado).

## Como são construídas as mensagens SOAP

Utiliza Integration Designer para configurar uma associação de importação de serviço da web (JAX-WS) para chamar um serviço da web externo. A associação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define quais os componentes da mensagem que



devem ser passados como anexos. É possível indicar, na página de Especificar a compatibilidade de WS-I AP 1.0 do Integration Designer, uma das seguintes opções:

- **Use WS-I AP 1.0 compliant SOAP message (Utilizar mensagem SOAP compatível com WS-I AP 1.0)**

Se seleccionar esta opção, também pode especificar qual o componente da mensagem que deve ser associado ao corpo SOAP; todos os outros são associados a anexos ou a cabeçalhos. As mensagens enviadas pela associação não incluem elementos no corpo SOAP que se refiram aos anexos; a relação é expressa através do identificador de conteúdo do anexo, incluindo o nome do componente da mensagem.

- **Use non WS-I AP 1.0 compliant SOAP message (Utilize mensagem SOAP não compatível com WS-I AP 1.0)**

Se seleccionar esta opção, a qual é a predefinição, o primeiro componente da mensagem é associado ao corpo SOAP; todos os outros são associados a anexos ou cabeçalhos. As mensagens enviadas pela associação incluem um ou mais elementos no corpo SOAP que se referem aos anexos através de um atributo **href**.

**Nota:** O componente que representa um anexo, como definido na WSDL, deve ser de tipo simples (base64Binary ou hexBinary). Se um componente for definido por um complexType, esse componente não pode ser associado como anexo.

### Processamento externo de anexos referenciados

A associação de importação utiliza informação no SMO para determinar como os componentes de mensagem binária de nível superior são enviados como anexos.

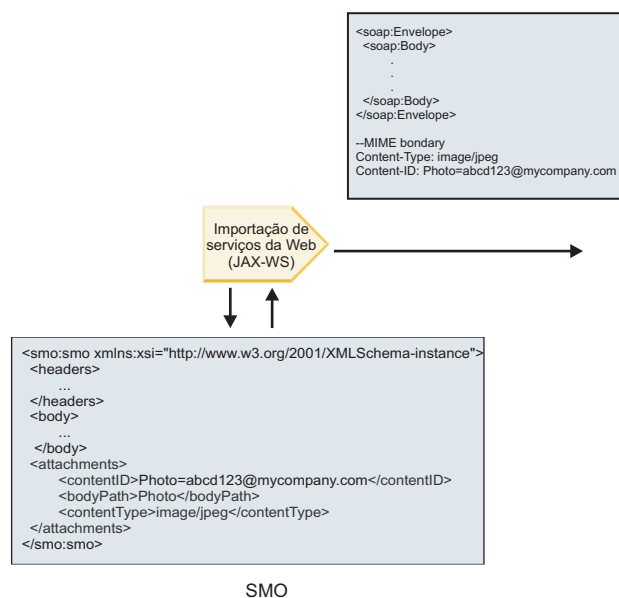


Figura 66. Como o anexo referenciado no SMO é acedido para criar a mensagem SOAP

O elemento dos **anexos** está presente no SMO apenas se um componente de fluxo de mediação estiver directamente ligado à importação ou exportação; não é passado por outros tipos de componente. Se os valores forem necessários num módulo que contenha outros tipos de componente, deve ser utilizado um componente de fluxo de mediação para copiar os valores para uma localização onde possam ser acedidos no módulo e utilizado um componente de fluxo de mediação para definir os valores correctos antes de uma chamada para execução através de uma importação de serviço da web.

A associação utiliza uma combinação das seguintes condições para determinar como (ou se) a mensagem é enviada:

- Se existe uma associação de WSDL MIME para o componente binário da mensagem de nível superior e, caso exista, como é definido o tipo de conteúdo
- Se existe um elemento dos **anexos** no SMO cujo valor de **bodyPath** faça referência a um componente binário de nível superior

### Como os anexos são criados quando um elemento do anexo existe no SMO

A tabela seguinte mostra como um anexo é criado e enviado caso o SMO contenha um elemento do **anexo** com um **bodyPath** que corresponda a um componente do nome da mensagem:

Tabela 61. Como é criado o anexo

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> <li>• Nenhum tipo de conteúdo definido para o componente da mensagem</li> <li>• Tipos múltiplos de conteúdo definidos</li> <li>• Tipo de conteúdo de carácter global definido</li> </ul>	Componente da mensagem enviado como um anexo.  O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.  O <b>Content-Type</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para <b>application/octet-stream</b> .
Presente com conteúdo individual, sem carácter global, para o componente da mensagem	Componente da mensagem enviado como um anexo.  O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.  O <b>tipo de conteúdo</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para o tipo definido no elemento do conteúdo de WSDL MIME.
Não presente	Componente da mensagem enviado como um anexo.  O <b>Content-Id</b> é definido para o valor no elemento do anexo caso esteja presente; caso contrário, é criado um.  O <b>Content-Type</b> é definido para o valor no elemento dos anexos, se presente; caso contrário, é definido para <b>application/octet-stream</b> . <b>Nota:</b> Enviar componentes da mensagem como anexos quando não assim definido na WSDL pode interromper a compatibilidade com o Perfil de Anexos de WS-I 1.0 e isto deve ser evitado, se possível.

### Como os anexos são criados quando não existe qualquer elemento do anexo no SMO

A tabela seguinte mostra como um anexo é criado e enviado caso o SMO não contenha um elemento do **anexo** com um **bodyPath** que corresponda a um componente do nome da mensagem:

Tabela 62. Como é criado o anexo

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> <li>• Nenhum tipo de conteúdo definido para o componente da mensagem</li> <li>• Tipos múltiplos de conteúdo definidos</li> <li>• Tipo de conteúdo de carácter global definido</li> </ul>	Componente da mensagem enviado como um anexo.  É criada o <b>identificador de conteúdo</b> .  O <b>tipo de conteúdo</b> é definido para <b>application/octet-stream</b> .

Tabela 62. Como é criado o anexo (continuação)

Estado da associação WSDL MIME para componente binário de nível superior da mensagem	Como a mensagem é criada e enviada
Presente com conteúdo individual, sem carácter global, para o componente da mensagem	Componente da mensagem enviado como um anexo. É criada o <b>identificador de conteúdo</b> . O <b>tipo de conteúdo</b> é definido para o tipo definido no elemento do conteúdo de WSDL MIME.
Não presente	Componente da mensagem não enviado como anexo.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.
- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

Anexos não referenciados: **Advanced**

Pode enviar e receber anexos *não referenciados* que não estejam declarados na interface de serviço.

Numa mensagem SOAP de multi-parte MIME, o corpo SOAP é o primeiro componente da mensagem e os anexos estão nos componentes subsequentes. Não é incluída qualquer referência ao anexo no corpo SOAP.

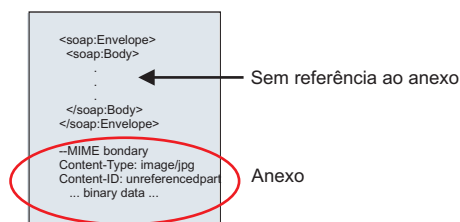


Figura 67. Uma mensagem SOAP com um anexo não referenciado

Pode enviar uma mensagem SOAP com um anexo não referenciado através de uma exportação de serviço da web para uma importação de serviço da web. A mensagem de saída, que é enviada para o serviço da web de destino, contém o anexo.

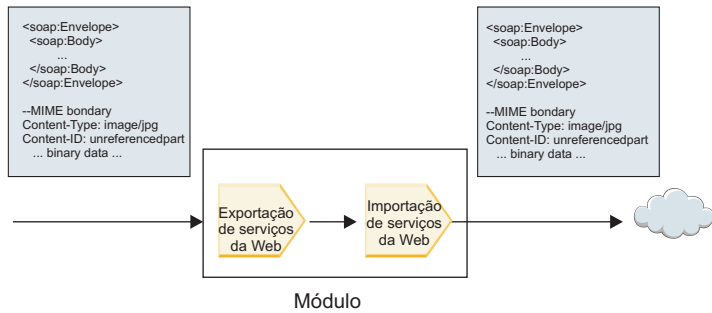


Figura 68. Um anexo a passar através de um módulo SCA

Na Figura 23 na página 93, a mensagem SOAP, com o anexo, passa sem modificação.

Também pode modificar a mensagem SOAP utilizando um componente de fluxo de mediação. Por exemplo, pode utilizar o componente de fluxo de mediação para extrair dados da mensagem SOAP (dados binários no corpo da mensagem, neste caso) e criar uma mensagem com anexos SOAP. Os dados são processados como componente do elemento dos anexos de um objecto de mensagem de serviço (SMO).

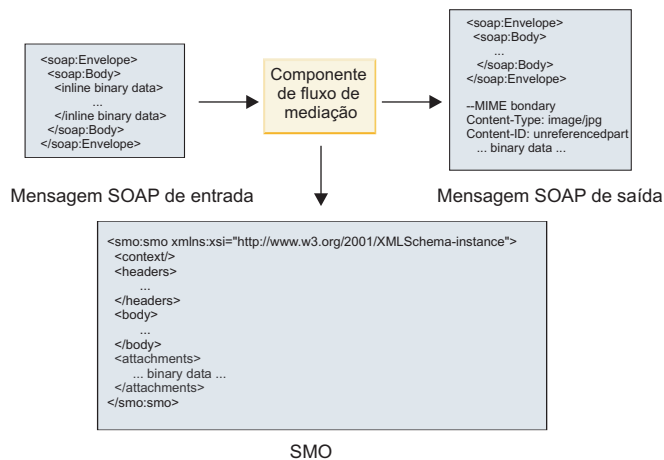


Figura 69. Uma mensagem processada por um componente de fluxo de mediação

Da mesma forma, o componente de fluxo de mediação pode transformar mensagens recebidas ao extrair e ao codificar o anexo e, de seguida, ao transmitir a mensagem sem anexos.

Em vez de extrair dados de uma mensagem SOAP recebida para formar uma mensagem com anexos SOAP, é possível obter os dados do anexo através de uma fonte externa, tal como uma base de dados.

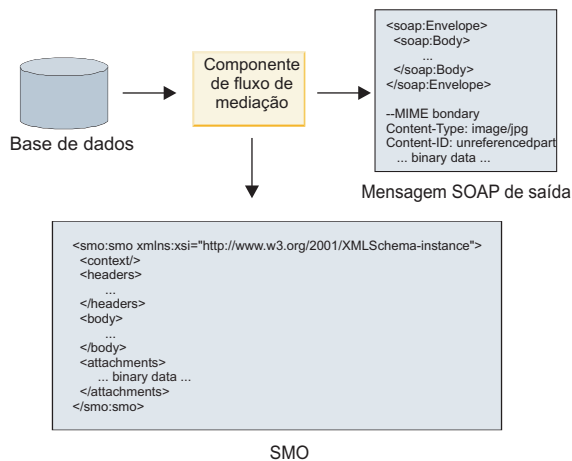


Figura 70. Um anexo obtido de uma base de dados e adicionado à mensagem SOAP

Da mesma forma, o componente de fluxo de mediação pode extrair o anexo de uma mensagem SOAP recebida e processar a mensagem (por exemplo, armazenar o anexo numa base de dados).

Os anexos não referenciados apenas podem ser propagados por componentes de fluxo de mediação. Caso um anexo deva ser acessado por ou propagado para outro tipo de componente, utilize um componente de fluxo de mediação para mover o anexo para uma localização que seja acessível por esse componente.

**Importante:** Como descrito em “representação XML de SMO,” a primitiva de mediação de mapeamento transforma as mensagens ao utilizar uma transformação XSLT 1.0. A transformação opera numa serialização XML do SMO. A primitiva de mediação de mapeamento permite que a raiz de serialização seja especificada e o elemento raiz do documento XML reflecte esta raiz.

Quando enviar mensagens SOAP com anexos, o elemento raiz que seleccionar determina como os anexos são propagados.

- Se utilizar “/corpo” como raiz do mapa XML, todos os anexos são propagados pelo mapa, por predefinição.
- Se utilizar “/” como raiz do mapa, é possível controlar a propagação dos anexos.

#### Utilização de associação de estilo de documento WSDL com mensagens multi-parte: Advanced

A organização de Web Services Interoperability Organization (WS-I) definiu um conjunto de regras relativamente a como os serviços da web devem ser descritos através de uma WSDL e a como as mensagens SOAP correspondentes devem ser formadas, de modo a assegurar a interoperabilidade.

Estas regras estão especificadas no *Perfil Básico Versão 1.1* de WS-I (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). Em específico, o Perfil Básico de WS-I 1.1 R2712 declara: "Uma associação documento-literal TEM DE ser convertida num ENVELOPE com um soap:Corpo cujo elemento descendente é uma ocorrência da declaração do elemento global referenciado pelo componente wsdl:componente da mensagem."

Isto significa que, quando utilizar uma associação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) definida com componentes múltiplos, apenas um desses componentes deve ser associado ao corpo SOAP de modo a ser compatível com o Perfil Básico 1.1 de WS-I.

Em acréscimo, o Perfil de Anexos 1.0 R2941 de WS-I declara: "Uma wsdl:binding numa DESCRIÇÃO DEVE associar todos os wsdl:part de uma wsdl:message no wsdl:portType, a que se refere, a um de soapbind:body, soapbind:header, soapbind:fault soapbind:headerfault ou mime:content."

Isto significa que, quando utilizar uma associação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) definida com componentes múltiplos, todos os componentes que não aquele selecionado para ser associado ao corpo SOAP devem ser associados como anexos ou cabeçalhos.

A seguinte abordagem é utilizada quando as descrições de WSDL são criadas para exportações com associações de serviço da web (JAX-WS e JAX-RPC) neste caso:

- É possível seleccionar qual o componente da mensagem que é associado ao corpo SOAP caso exista mais do que um elemento de tipo não binário. Se apenas existir um elemento de tipo não binário, esse elemento é automaticamente associado ao corpo SOAP.
- Para a associação JAX-WS, todos os outros componentes da mensagem de tipo "hexBinary" ou "base64Binary" são associados como anexos referenciados. Consulte "Anexos referenciados: componentes de mensagem de nível superior" na página 88.
- Todos os outros componentes são associados como cabeçalhos SOAP.

As associações de importação JAX-RPC e JAX-WS respeitam a associação SOAP num documento WSDL existente com mensagens de estilo de documento multi-parte, mesmo que associem os componentes múltiplos ao corpo SOAP; porém, não é possível criar clientes de serviço da web para tais documentos WSDL no Rational Application Developer.

**Nota:** A associação JAX-RPC não suporta anexos.

O padrão recomendado quando utilizar mensagens multi-parte com uma operação com associação SOAP de estilo de documento é, assim:

1. Utilizar o estilo moldado de documento/literal. Neste caso, as mensagens têm sempre um componente individual; porém, os anexos não podem ser referenciados (como descrito em "Anexos não referenciados" na página 93) ou do tipo swaRef (como descrito em "Anexos referenciados: elementos do tipo swaRef" na página 84) neste caso.
2. Utilizar o estilo RPC/literal. Neste caso, não existem restrições na associação WSDL em termos de número de componentes associados ao corpo SOAP; a mensagem SOAP resultante tem sempre um descendente único que representa a operação a ser chamada, com os componentes da mensagem a serem descendentes desse elemento.
3. Para a associação JAX-WS, tenha, no máximo, um componente da mensagem que não seja do tipo "hexBinary" ou "base64Binary", excepto se for aceitável associar os outros componentes não binários aos cabeçalhos SOAP.
4. Quaisquer outros casos estão sujeitos ao comportamento descrito.

**Nota:** Existem restrições adicionais quando utiliza mensagens SOAP que não são compatíveis com o *Perfil Básico Versão 1.1* de WS-I.

- O primeiro componente da mensagem deve ser não binário.
- Quando recebe mensagens SOAP de estilo de documento multi-parte com anexos referenciados, a associação JAX-WS espera que cada anexo referenciado seja representado por um elemento descendente do corpo SOAP com um valor de atributo href que identifica o anexo pelo respectivo identificador de conteúdo. A associação JAX-WS envia anexos referenciados para tais mensagens do mesmo modo. Este comportamento não é compatível com o Perfil Básico de WS-I.

Para assegurar que as suas mensagens são compatíveis com o Perfil Básico, siga a abordagem 1 na página 95 ou 2 na página 96 na lista anterior ou evite a utilização de anexos referenciados para tais mensagens e utilize anexos não referenciados ou do tipo swaRef.

## Associações HTTP

A associação HTTP está concebida para fornecer a conectividade da área de comunicações partilhada (SCA) ao HTTP. Como consequência, aplicações HTTP existentes ou desenvolvidas recentemente podem participar em ambientes de Arquitectura orientada para o serviço (SOA, Service Oriented Architecture).

HTTP é um protocolo amplamente utilizado para transferir informações na web. Quando está a trabalhar com uma aplicação externa que utiliza o protocolo HTTP, é necessária uma associação HTTP. A associação HTTP transforma os dados que são transferidos para uma mensagem num formato nativo para um objecto de negócio numa aplicação SCA. A associação HTTP também pode transformar dados que são transferidos como um objecto de negócio para um formato nativo esperado pela aplicação externa.

**Nota:** Se pretender interagir com clientes e serviços que utilizam o protocolo SOAP/HTTP dos serviços da Web, considere utilizar uma das associações de serviço da web que oferecem funcionalidade adicional referente ao processamento de qualidades de serviço padrão de serviços da web.

Alguns cenários comuns para utilizar a associação HTTP são descritos na lista seguinte:

- Serviços alojados em SCA podem chamar aplicações HTTP utilizando uma importação HTTP.
- Serviços alojados em SCA podem expor-se como aplicações activadas por HTTP, para que possam ser utilizadas por clientes HTTP, utilizando uma exportação HTTP.
- IBM Business Process Manager e Process Server podem comunicar entre eles em toda a infra-estrutura HTTP, conseqüentemente, os utilizadores conseguem gerir as suas comunicações de acordo com padrões empresariais.
- IBM Business Process Manager e Process Server podem agir como mediadores de comunicações HTTP, transformando e encaminhando mensagens, o que melhora a integração de aplicações utilizando uma rede HTTP.
- IBM Business Process Manager e Process Server podem ser utilizados para criar uma ponte entre HTTP e outros protocolos, tais como serviços da Web SOAP/HTTP, adaptadores de recurso baseados em Java Connector Architecture (JCA), JMS, etc.

Pode encontrar informações detalhadas sobre como criar associações de importação e exportação HTTP no Information Center do Integration Designer. Consulte os tópicos **Developing integration applications (Desenvolver aplicações de integração) > Accessing external services with HTTP (Aceder a serviços externos com HTTP)**.

### **Descrição geral das associações HTTP:**

A associação HTTP fornece conectividade às aplicações alojadas em HTTP. Ela media a comunicação entre aplicações HTTP e permite aplicações existentes baseadas em HTTP a serem chamadas de um módulo.

#### **Associações de importação HTTP**

A associação de importação HTTP fornece conectividade externa de aplicações da área de comunicações partilhada (SCA, Service Component Architecture) para um servidor HTTP ou aplicações.

A importação chama um URL de ponto de terminação HTTP. O URL pode ser especificado de uma das três formas:

- O URL pode ser definido dinamicamente nos cabeçalhos HTTP através do URL de substituição dinâmico.
- O URL pode ser definido dinamicamente no elemento de endereço destino SMO.
- O URL pode ser especificado como uma propriedade de configuração na importação.

Esta chamada é sempre síncrona por natureza.

Apesar das chamadas HTTP serem sempre pedido-resposta, a importação HTTP suporta tanto operações unidireccionais como bidireccionais e ignora a resposta no caso de uma operação unidireccional.

#### **Associações de exportação HTTP**

A associação de exportação HTTP fornece conectividade interna de aplicações HTTP para uma aplicação SCA.



Um URL é definido na exportação HTTP. As aplicações HTTP que pretendem enviar mensagens de pedido à exportação utilizam este URL para chamar a exportação.

A exportação HTTP também suporta (comando) ping.

### Associações HTTP em tempo de execução

Uma importação com uma associação HTTP em tempo de execução envia um pedido com ou sem dados no corpo da mensagem da aplicação SCA para o serviço externo da web. O pedido é feito pela aplicação SCA para o serviço externo da web, como apresentado na Figura 26 na página 97.

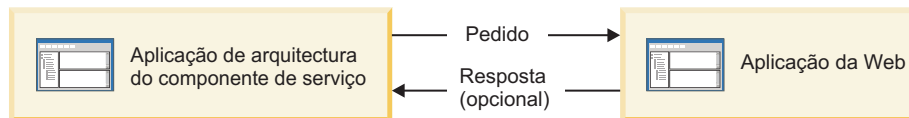


Figura 71. Fluxo de um pedido da aplicação SCA para a aplicação web

Opcionalmente, a importação com a associação HTTP pode receber dados de volta da aplicação da web numa resposta ao pedido.

Com uma exportação, o pedido é feito por uma aplicação de cliente para um serviço da web, como apresentado na Figura 27 na página 98.

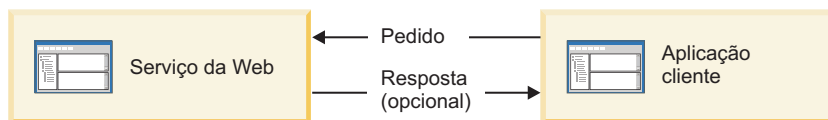


Figura 72. Fluxo de um pedido a partir da aplicação cliente para o serviço da Web.

O serviço da web é uma aplicação da web em execução no servidor. A exportação é implementada nessa aplicação da web como um servlet, para que o cliente envie o seu pedido para um endereço URL. O servlet passa o pedido à aplicação SCA em tempo de execução.

Opcionalmente, a exportação pode enviar dados à aplicação de cliente em resposta ao pedido.

### Cabeçalhos HTTP:

As associações de importação e exportação HTTP permitem a configuração de cabeçalhos HTTP e que os seus valores sejam utilizados para mensagens de saída. A importação HTTP utiliza estes cabeçalhos para pedidos e a exportação HTTP utiliza-os para respostas.

Cabeçalhos configurados estaticamente e informações de controlo têm precedência sobre valores definidos dinamicamente no tempo de execução. No entanto, os valores de controlo de URL, versão e método de substituição dinâmica substituem os valores estáticos, que, de outra forma, são considerados predefinições.

A associação suporta a natureza dinâmica do URL de importação HTTP ao determinar o valor do URL, versão e método destino de HTTP em tempo de execução. Estes valores são determinados ao extrair o valor da Referência do ponto final, URL de substituição dinâmica, versão e método.

- Para a Referência de ponto final utilize `com.ibm.websphere.sca.addressing.EndpointReference` APIs ou defina o campo `/headers/SMOHeader/Target/address` no cabeçalho SMO.
- Para URL de substituição dinâmica, versão e método utilize a secção de parâmetros de controlo HTTP da área de comunicações partilhada (SCA). Tenha em atenção que o URL de substituição dinâmica tem

precedência sobre a Referência do ponto final destino; no entanto, a referência do ponto final aplica-se em associações, sendo a abordagem preferida e devendo ser utilizada, se possível.

As informações sobre controlo e cabeçalho sobre mensagens de saída nas associações de exportação e importação HTTP são processadas na seguinte ordem:

1. As informações de cabeçalho e controlo, excluindo URL de substituição dinâmica HTTP, versão e método da mensagem SCA (prioridade mais baixa)
2. Alterações da consola administrativa ao nível de exportação/importação
3. Alterações da consola administrativa ao nível de método da exportação ou importação
4. Endereço destino especificado através da Referência do ponto final ou do cabeçalho SMO
5. Substituição dinâmica URL, versão e método da mensagem SCA
6. Cabeçalhos e informação de controlo da rotina de tratamento de dados ou associação de dados (prioridade alta)

A exportação e importação HTTP só preenche os cabeçalhos de direcção de entrada e os parâmetros de controlo com dados da mensagem recebida (HTTPExportRequest e HTTPImportResponse) se a propagação de cabeçalhos de protocolo estiver definida para **True**. De forma inversa, a exportação e importação HTTP só lê e processa cabeçalhos externos e parâmetros de controlo (HTTPExportResponse e HTTPImportRequest) se a propagação de cabeçalhos de protocolo estiver definida para **True**.

**Nota:** Alterações na rotina de tratamento de dados ou associação de dados para cabeçalhos ou parâmetros de controlo na resposta de importação ou no pedido de exportação não alteram as instruções de processamento da mensagem dentro da associação de importação ou exportação e só devem ser utilizadas para propagar valores modificados para componentes SCA na direcção do fluxo.

O serviço de contexto é responsável pela propagação do contexto (incluindo os cabeçalhos de protocolo, como o cabeçalho HTTP e o contexto de utilizado, como o ID de conta) ao longo de um caminho de chamada SCA. Durante o desenvolvimento no IBM Integration Designer, pode controlar a propagação do contexto através das propriedades de importação e exportação. Para obter mais detalhes, consulte as informações sobre associações de importação e exportação no Information Center do IBM Integration Designer.

### Estruturas de cabeçalho HTTP fornecidas e suporte

A Tabela 34 na página 99 lista os parâmetros de pedido/resposta para pedidos e respostas de importação HTTP e exportação HTTP.

*Tabela 63. Informações de cabeçalho HTTP fornecidas*

Nome de controlo	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
URL	Ignorado	Não definido	Ler a partir da mensagem de pedido. <b>Nota:</b> A cadeia de consulta também faz parte do parâmetro de controlo URL.	Ignorado
Versão (valores possíveis: 1.0, 1.1; predefinição é 1.1)	Ignorado	Não definido	Ler a partir da mensagem de pedido	Ignorado
Método	Ignorado	Não definido	Ler a partir da mensagem de pedido	Ignorado

Tabela 63. Informações de cabeçalho HTTP fornecidas (continuação)

Nome de controlo	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
URL de substituição dinâmica	Se definido na rotina de tratamento de dados ou associação de dados, substitui o URL de importação HTTP. Escrito para a mensagem na linha de pedido. <b>Nota:</b> A cadeia de consulta também faz parte do parâmetro de controlo URL.	Não definido	Não definido	Ignorado
Versão de substituição dinâmica	Se definido, substitui a versão de importação HTTP. Escrito para a mensagem na linha de pedido.	Não definido	Não definido	Ignorado
Método de substituição dinâmica	Se definido, substitui o Método de importação HTTP. Escrito para a mensagem na linha de pedido.	Não definido	Não definido	Ignorado
Tipo de meio (este parâmetro de controlo traz parte do valor do cabeçalho HTTP do tipo de conteúdo.)	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela rotina de tratamento de dados ou associação de dados.	Ler a partir da mensagem de pedido, cabeçalho do tipo de conteúdo	Ler a partir da mensagem de resposta, cabeçalho do tipo de conteúdo	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela rotina de tratamento de dados ou associação de dados.
Conjunto de caracteres (predefinição: UTF-8)	Se presente, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela associação de dados.	Ler a partir da mensagem de pedido, cabeçalho do tipo de conteúdo	Ler a partir da mensagem de resposta, cabeçalho do tipo de conteúdo	Suportado, escrito para a mensagem como parte do cabeçalho do tipo de conteúdo. <b>Nota:</b> Este valor de elemento de controlo deve ser fornecido pela associação de dados.
Codificação de transferência (valores possíveis: segmentado, identidade; predefinição é identidade)	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a transformação de mensagem.	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a transformação de mensagem.

Tabela 63. Informações de cabeçalho HTTP fornecidas (continuação)

Nome de controlo	Pedido de importação HTTP	Resposta de importação HTTP	Pedido de exportação HTTP	Resposta de exportação HTTP
Codificação de conteúdo (valores possíveis: gzip, x-gzip, deflacionar, identidade; predefinição é identidade)	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a carga útil.	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Se presente, escrito para a mensagem como um cabeçalho e controla de que forma é codificada a carga útil.
Comprimento do conteúdo	Ignorado	Ler a partir da mensagem de resposta	Ler a partir da mensagem de pedido	Ignorado
StatusCode (predefinição: 200)	Não suportado	Ler a partir da mensagem de resposta	Não suportado	Se presente, escrito para a mensagem na linha de resposta.
ReasonPhrase (predefinição: OK)	Não suportado	Ler a partir da mensagem de resposta	Não suportado	Valor de controlo ignorado. O valor da linha de resposta da mensagem é gerado a partir do StatusCode.
Autenticação (contém várias propriedades)	Se presente, utilizado para construir o cabeçalho de Autenticação básica. <b>Nota:</b> O valor para este cabeçalho só é codificado no protocolo HTTP. No SCA, é descodificado e passado como texto limpo.	Não aplicável	Ler a partir do cabeçalho de Autenticação básica da mensagem de pedido. A presença deste cabeçalho não indica que o utilizador tenha sido autenticado. A autenticação deve ser controlada na configuração servlet. <b>Nota:</b> O valor para este cabeçalho só é codificado no protocolo HTTP. No SCA, é descodificado e passado como texto limpo.	Não aplicável
Proxy (contém várias propriedades: sistema central, porta, autenticação)	Se presente, utilizado para estabelecer ligação através do proxy.	Não aplicável	Não aplicável	Não aplicável
SSL (contém várias propriedades: arquivo de chaves, palavra-passe do arquivo de chaves, Trustore, palavra-passe do Trustore, ClientAuth)	Se estiver preenchido e o URL de destino for HTTPS, é utilizado para estabelecer uma ligação através do SSL.	Não aplicável	Não aplicável	Não aplicável

## Associações de dados HTTP:

Para cada mapeamento diferente de dados entre uma mensagem de Arquitectura de Componentes de Serviço (SCA, Service Component Architecture) e uma mensagem de protocolo HTTP, tem de ser configurada uma rotina de tratamento de dados ou uma associação de dados HTTP. As rotinas de tratamento de dados fornecem uma interface neutra de associação que permite a reutilização em todas as associações de transporte e representa a abordagem recomendada; as associações de dados são específicas para uma determinada associação de transporte. As classes de associação de dados específicas de HTTP são fornecidas; também pode gravar rotinas de tratamento de dados personalizadas ou associações de dados.

**Nota:** As três classes de associação de dados HTTP descritas neste tópico (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão obsoletas desde o IBM Business Process Manager Versão 7.0. Em vez de utilizar as associações de dados descritas neste tópico, considere as seguintes rotinas de tratamento de dados:

- Utilize SOAPDataHandler em vez de HTTPStreamDataBindingSOAP.
- Utilize UTF8XMLDataHandler em vez de HTTPStreamDataBindingXML
- Utilize GatewayTextDataHandler em vez de HTTPServiceGatewayDataBinding

As associações de dados são fornecidas para a utilização com importações HTTP e exportações HTTP: associação de dados binários, associação de dados XML e associação de dados SOAP. Não há necessidade de uma associação de dados de resposta para operações unidireccionais. Uma associação de dados é representada pelo nome de uma classe Java cujas instâncias podem ser convertidas de HTTP para ServiceDataObject e vice-versa. Um selector de funções tem de ser utilizado numa exportação que, juntamente com associações de métodos, pode determinar qual é a associação de dados utilizada e qual a operação chamada. As associações de dados fornecidas são:

- Associações de dados binários, que tratam o corpo como dados binários não estruturados. A implementação do esquema XSD de associação de dados binários é como segue:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- Associações de dados XML que suportam o corpo como dados XML. A implementação da associação de dados XML é semelhante à associação de dados JMS XML e não tem restrições no esquema de interface.
- Associações de dados SOAP que suportam o corpo como dados SOAP. A implementação da associação de dados SOAP não tem restrições no esquema de interface.

## Implementar associações de dados HTTP personalizadas

Esta secção descreve como implementar uma associação de dados HTTP personalizada.

**Nota:** A abordagem recomendada é implementar uma rotina de tratamento de dados personalizada porque pode ser reutilizada em todas as associações de transporte.

HTTPStreamDataBinding é a interface principal para processar mensagens HTTP personalizadas. A interface está concebida para permitir o processamento de grandes cargas úteis. No entanto, para tal implementação funcionar, esta associação de dados tem de devolver as informações de controlo e cabeçalhos antes de gravar a mensagem na sequência.

Os métodos e a sua ordem de execução, listados abaixo, têm de ser implementados pela associação de dados personalizados.

Para personalizar uma associação de dados, grave uma classe que implemente o HTTPStreamDataBinding. A associação de dados deve ter quatro propriedades privadas:

- DataObject pDataObject privado
- HTTPControl pCtrl privado
- HTTPHeaders pHeaders privado
- yourNativeDataType nativeData privado

A associação HTTP chama a associação de dados personalizada na seguinte ordem:

- Processamento externo (DataObject para formato nativo):
  1. setDataObject(...)
  2. setHeaders(...)
  3. setControlParameters(...)
  4. setBusinessException(...)
  5. convertToNativeData()
  6. getControlParameters()
  7. getHeaders()
  8. write(...)
- Processamento interno (formato nativo para DataObject):
  1. setControlParameters(...)
  2. setHeaders(...)
  3. convertFromNativeData(...)
  4. isBusinessException()
  5. getDataObject()
  6. getControlParameters()
  7. getHeaders()

Necessita de chamar setDataObject(...) em convertFromNativeData(...) para definir o valor de dataObject, que é convertido de dados nativos para a propriedade privada "pDataObject".

```
public void setDataObject(DataObject dataObject)
    throws DataBindingException {
    pDataObject = dataObject;
}

public void setControlParameters(HTTPControl arg0) {
    this.pCtrl = arg0;
}

public void setHeaders(HTTPHeaders arg0) {
    this.pHeaders = arg0;
}
/*
* Adicione cabeçalho http "IsBusinessException" em pHeaders.
```

```

* Dois passos:
* 1.Remova primeiro todo o cabeçalho com o nome IsBusinessException
* (não sensível a maiúsculas e minúsculas)
* Isto serve para garantir que só está presente um cabeçalho.
* 2.Adicione o novo cabeçalho "IsBusinessException"
*/
public void setBusinessException(boolean isBusinessException) {
    //remova primeiro todos os cabeçalhos com o nome IsBusinessException
    (não sensível a maiúsculas e minúsculas).
    //Isto serve para garantir que só está presente um cabeçalho.
    //Adicione o novo cabeçalho "IsBusinessException", exemplo de código:
    HTTPHeader header=HeadersFactory.eINSTANCE.createHTTPHeader();
    header.setName("IsBusinessException");
    header.setValue(Boolean.toString(isBusinessException));
    this.pHeaders.getHeader().add(header);
}
public HTTPControl getControlParameters() {
    return pCtrl;
}
public HTTPHeaders getHeaders() {
    return pHeaders;
}
public DataObject getDataObject() throws DataBindingException {
    return pDataObject;
}
}
/*
* Obter o cabeçalho "IsBusinessException" de pHeaders, devolve o respectivo valor booleano
*/
public boolean isBusinessException() {
    String headerValue = getHeaderValue(pHeaders,"IsBusinessException");
    boolean result=Boolean.parseBoolean(headerValue);
    return result;
}
public void convertToNativeData() throws DataBindingException {
    DataObject dataObject = getDataObject();
    this.nativeData=realConvertWorkFromSD0ToNativeData(dataObject);
}
public void convertFromNativeData(HTTPInputStream arg0){
    //Customer-developed method to
    //Read data from HTTPInputStream
    //Convert it to DataObject
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}
public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}
}

```

## Associações EJB

As associações de importação Enterprise JavaBeans (EJB) permitem que componentes da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) chamem serviços fornecidos pela lógica empresarial Java EE a correr num servidor Java EE. As associações de exportação EJB permitem que componentes SCA sejam expostos como Enterprise JavaBeans, de forma que a lógica empresarial Java EE possa chamar componentes SCA de outra forma não disponíveis para.

### Associações de importação EJB:

As associações de importação EJB permitem que um módulo SCA invoque implementações EJB através da especificação da forma como o módulo de consumo está ligado ao EJB externo. Importar serviços de uma implementação EJB externa permite aos utilizadores ligarem as suas lógicas empresariais ao ambiente IBM Business Process Manager e participarem num processo empresarial.



Utiliza o Integration Designer para criar associações de importação EJB. Pode utilizar qualquer um dos seguintes procedimentos para gerar associações:

- Criar importação EJB utilizando o assistente do serviço externo  
Pode utilizar o assistente do serviço externo no Integration Designer para criar uma importação EJB baseada numa implementação existente. O assistente do serviço externo cria serviços com base em critérios fornecidos pelo utilizador. Em seguida, cria objectos de negócio, interfaces e ficheiros de importação baseados nos serviços descobertos.
- Criar importação EJB utilizando o editor de montagem  
Pode criar uma importação EJB dentro de um diagrama de montagem utilizando o editor de montagem do Integration Designer. A partir da paleta, pode utilizar uma Importação ou utilizar uma classe de Java para criar a associação EJB.

A importação gerada tem associações de dados que criam a ligação Java-WSDL em vez de necessitarem de um componente de ponte Java. Pode ligar directamente um componente com uma referência Web Services Description Language (WSDL) à importação EJB que comunica com um serviço baseado em EJB utilizando uma interface Java.

A importação EJB pode interagir com uma lógica empresarial Java EE utilizando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada para execução para a lógica empresarial Java EE pode ser local (apenas para EJB 3.0) ou remota.

- A chamada para execução local é utilizada quando pretende invocar a lógica empresarial Java EE que reside no mesmo servidor que a importação.

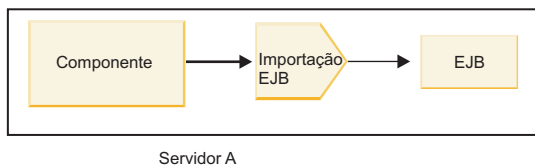


Figura 73. A chamada para execução local de um EJB (apenas EJB 3.0)

- A chamada para execução remota é utilizada quando pretende invocar a lógica empresarial Java EE que não reside no mesmo servidor que a importação.  
Por exemplo, na seguinte figura, uma importação EJB utiliza a Remote Method Invocation pelo Internet InterORB Protocol (RMI/IIOP) para chamar um método EJB noutra servidor.

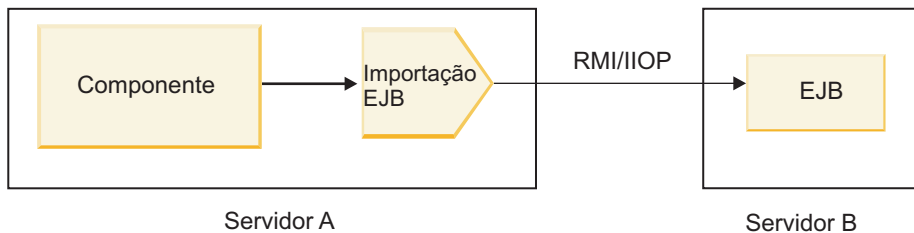


Figura 74. Chamada para execução remota de um EJB

Quando configura a associação EJB, o Integration Designer utiliza o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada para execução (local ou remota).

Associações de importação EJB contêm os seguintes componentes principais:

- rotina de tratamento de dados JAX-WS
- selector de falha EJB

- selector de função de importação EJB

Se o cenário de utilizador não for baseado no mapeamento JAX-WS, poderá necessitar de uma rotina de tratamento de dados, selector de função e selector de falha para executar as tarefas que são concluídas de outro modo pelos componentes que fazem parte das associações de importação EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento personalizado.

### Associações de exportação EJB:

Aplicações externas Java EE podem chamar um componente SCA através de uma associação de exportação de um EJB. Através da utilização de uma exportação EJB é possível expor componentes SCA para que as aplicações externas Java EE possam chamar esses componentes para execução através da utilização do modelo de programação EJB.

**Nota:** A exportação EJB é um bean sem registo.

Utiliza o Integration Designer para criar associações EJB. Pode utilizar qualquer um dos seguintes procedimentos para gerar associações:

- Criar associações de exportação EJB utilizando o assistente do serviço externo  
Pode utilizar o assistente do serviço externo no Integration Designer para criar um serviço de exportação EJB com base numa implementação existente. O assistente do serviço externo cria serviços com base em critérios fornecidos pelo utilizador. Em seguida, gera objectos de negócio, interfaces e ficheiros de exportação com base no serviços descobertos.
- Criar associações de exportação EJB utilizando o editor de montagem  
Pode criar uma exportação EJB utilizando o editor de montagem do Integration Designer.

**Importante:** Um cliente Java 2 Platform, Standard Edition (J2SE) não consegue chamar o cliente de exportação EJB que é gerado no Integration Designer.

Pode gerar a associação a partir de um componente SCA existente ou pode gerar uma exportação com uma associação EJB para uma interface Java.

- Ao gerar uma exportação para um componente SCA existente que tenha uma interface WSDL existente, à exportação é atribuída uma interface Java interface.
- Ao gerar uma exportação para uma interface Java, pode seleccionar uma interface WSDL ou uma interface Java para a exportação.

**Nota:** Uma interface Java utilizada para criar uma exportação EJB tem as seguintes limitações no que diz respeito a objectos (parâmetros de entrada e de saída e excepções) transferidas como parâmetros numa chamada remota:

- Têm de ser de um tipo concreto (em vez de uma interface ou tipo abstracto).
- Têm de estar em conformidade com a especificação Enterprise JavaBeans. Têm de poder ser colocados em série e terem o construtor sem argumento predefinido, para que todas as propriedades possam estar acessíveis através de métodos de obtenção e definição.

Consulte o sítio da Web da Sun Microsystems, Inc., em <http://java.sun.com> para mais informações sobre a especificação JavaBeans.

Para além disso, a excepção tem de ser uma excepção verificada, herdada de `java.lang.Exception`, e tem de ser singular (ou seja, não suporta emitir vários tipos de excepção verificada).

Note também que a interface empresarial de um Java EnterpriseBean e uma interface Java simples e não deve estender-se a `javax.ejb.EJBObject` ou `javax.ejb.EJBLocalObject`. Os métodos de interface empresarial não devem apresentar uma excepção `java.rmi.RemoteException`.

As associações de exportação EJB podem interagir com a lógica empresarial Java EE através da utilização do modelo de programação EJB 2.1 ou do modelo de programação EJB 3.0.

A chamada para execução pode ser local (para EJB 3.0 apenas) ou remota.

- A chamada para execução local é utilizada quando a lógica empresarial Java EE chama um componente SCA residente no mesmo servidor que a exportação.
- A chamada para execução remota é utilizada quando a lógica empresarial Java EE business não reside no mesmo servidor que a exportação.

Por exemplo, na seguinte figura, um EJB utiliza RMI/IIOP para chamar um componente SCA num servidor diferente.

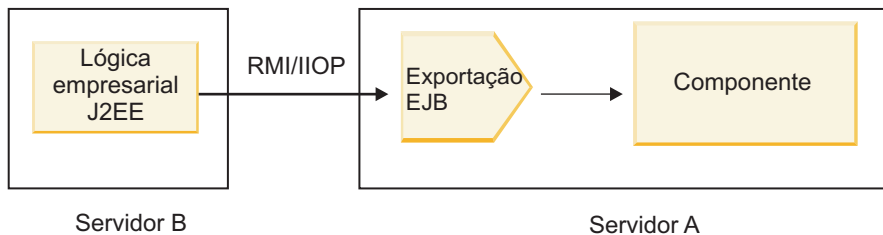


Figura 75. Chamada remota de um cliente para um componente SCA através de uma exportação EJB

Quando configura a associação EJB, o Integration Designer utiliza o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada para execução (local ou remota).

As associações de exportação EJB contêm os seguintes componentes principais:

- rotina de tratamento de dados JAX-WS
- selector de função de exportação EJB

Se o cenário de utilizador não for baseado no mapeamento JAX-WS, poderá necessitar de uma rotina de tratamento de dados e selector de função para executar as tarefas que são concluídas de outro modo pelos componentes que fazem parte das associações de exportação EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento personalizado.

### Propriedades de associação EJB:

As associações de importação EJB utilizam os seus nomes JNDI configurados para determinar o nível de modelo de programação EJB e tipo de chamada para execução (local ou remota). As associações de importação e exportação EJB utilizam a rotina de tratamento de dados JAX-WS para conversão de dados. A associação de importação EJB utiliza um selector de função EJB e um selector de falha EJB, e a associação de exportação EJB utiliza um selector de função de exportação EJB.

#### Nomes JNDI e associações de importação EJB:

Ao configurar a associação EJB numa importação, o Integration Designer utiliza o nome JNDI para determinar o nível de programação EJB e tipo de chamada para execução (local ou remota).

Se não for especificado nenhum nome JNDI, é utilizada a associação de interface EJB predefinida. Os nomes predefinidos que são criados dependem de ser chamado o EJB 2.1 JavaBeans ou EJB 3.0 JavaBeans.

**Nota:** Consulte o tópico "descrição geral das associações da aplicação EJB 3.0" no Information Center WebSphere Application Server para obter informações mais detalhadas sobre as convenções de nomeação.

- EJB 2.1 JavaBeans

O nome JNDI predefinido seleccionado previamente pelo Integration Designer é a associação EJB 2.1 predefinida, que inclui o formato **ejb/** e acrescenta a interface da página inicial, separados por barras.

Por exemplo, para a interface da página inicial do EJB 2.1 JavaBeans para `com.mycompany.myremotebusinesshome`, a associação predefinida é:

```
ejb/com/mycompany/myremotebusinesshome
```

Para EJB 2.1 apenas é suportada a chamada para execução EJB remota.

- EJB 3.0 JavaBeans

O nome JNDI predefinido seleccionado previamente pelo Integration Designer para o JNDI local é o nome de classe completo da interface local precedido por **ejblocal**: Por exemplo, para a interface completa da interface local `com.mycompany.mylocalbusiness`, o JNDI EJB 3.0 seleccionado previamente é:

```
ejblocal:com.mycompany.mylocalbusiness
```

Para a interface remota `com.mycompany.myremotebusiness`, o JNDI EJB 3.0 seleccionado previamente é a interface completa:

```
com.mycompany.myremotebusiness
```

As associações predefinidas EJB 3.0 são descritas na seguinte localização: Descrição geral das associações de aplicação EJB 3.0.

Integration Designer utilizará o nome "curto" como a localização JNDI predefinida para EJBs que utilizem o modelo de programação versão 3.0.

**Nota:** Se a referência JNDI implementada do EJB de destino for diferente da localização de associação JNDI predefinida, porque foi utilizado ou configurado um mapeamento personalizado, o nome JNDI de destino tem de ser especificado adequadamente. Pode especificar o nome no Integration Designer antes da implementação, ou , para a associação de importação, pode alterar o nome na consola administrativa (depois da implementação) para corresponder ao nome JNDI do EJB de destino.

Para obter mais informações sobre como criar associações EJB, consulte a secção dedicada a trabalhar com associações EJB no Information Center do Integration Designer.

#### *Rotina de tratamento de dados JAX-WS:*

A associação de importação do Enterprise JavaBeans (EJB) utiliza a rotina de tratamento de dados JAX-WS para transformar objectos de negócio de pedido em parâmetros de objecto Java e para transformar o valor de devolução do objecto Java no objecto de negócio de resposta. A associação de exportação EJB utiliza a rotina de tratamento de dados JAX-WS para transformar EJBs de pedido em objectos de negócio de pedido e para transformar o objecto empresarial de resposta num valor de devolução.

Esta rotina de tratamento de dados mapeia dados a partir da interface WSDL especificada por SCA para a interface Java EJB (e vice-versa) utilizando a especificação Java API for XML Web Services (JAX-WS) e a especificação Java Architecture for XML Binding (JAXB).

**Nota:** O suporte actual está restrito às especificações JAX-WS 2.1.1 e JAXB 2.1.3.

A rotina de tratamento de dados especificada no nível de associação EJB é utilizada para executar processamento de pedido, resposta, falha e excepção do tempo de execução.

**Nota:** Para falhas pode ser especificada uma rotina de tratamento de dados específica para cada falha através da especificação da propriedade de configuração `faultBindingType`. Isto substitui o valor especificado no nível de associação EJB.

A rotina de tratamento de dados JAX-WS é utilizada por predefinição quando a associação EJB tem uma interface WSDL. Esta rotina de tratamento de dados não pode ser utilizada para converter uma mensagem SOAP a representar uma chamada para execução JAX-WS num objecto de dados.

A associação de importação EJB utiliza uma rotina de tratamento de dados para converter um objecto de dados numa matriz Java Object (Object[]). Durante as comunicações externas, ocorre o seguinte processamento:

1. A associação EJB define o tipo esperado, elemento esperado e nome do método de destino no BindingContext (contexto da associação) para corresponder ao especificado no WSDL.
2. A associação EJB chama o método de conversão para o objecto de dados que requer a conversão de dados.
3. A rotina de tratamento de dados devolve um Object[] (objecto) que representa os parâmetros do método (na ordem da sua definição dentro do método).
4. A associação EJB utiliza o Object[] para chamar o método na interface EJB de destino.

A associação também prepara um Object[] para processar a resposta da chamada para execução EJB.

- O primeiro elemento no Object[] é o valor de devolução da chamada para execução de método Java.
- Os valores subsequentes representam os parâmetros de entrada para o método.

Isto é necessário para suportar os tipos de parâmetros In/Out e Out.

Para parâmetros do tipo Out, os valores têm de ser devolvidos no objecto de dados de resposta.

A rotina de tratamento de dados processa e converte valores encontrados no Object[] e depois devolve uma resposta ao objecto de dados.

A rotina de tratamento de dados suporta xs:AnyType, xs:AnySimpleType e xs:Any, juntamente com outros tipos de dados XSD. Para permitir suporte para xs:Any, utilize o **@XmlElement (lax=true)** para a propriedade JavaBeans no código Java, tal como mostrado no exemplo seguinte:

```
public class TestType {
    private Object[] object;

    @XmlElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Isto transforma o objecto da propriedade em TestType num campo xs:any. O valor da classe de Java utilizado no campo xs:any deve ter a anotação **@XmlElement**. Por exemplo, se Address (endereço) for a classe de Java a ser utilizada para preencher a matriz do objecto, a classe Address deve ter a anotação **@XmlElement**.

**Nota:** Para personalizar o mapeamento dos tipos de Java XSD definido pela especificação JAX-WS, altere as anotações JAXB para se adequarem às necessidades do negócio. A rotina de tratamento de dados JAX-WS suporta xs:any, xs:anyType e xs:anySimpleType.

As seguintes restrições são aplicáveis à rotina de tratamento de dados JAX-WS:

- A rotina de tratamento de dados não inclui suporte para a anotação de atributo de cabeçalho **@WebParam**.
- O espaço de nomes para ficheiros de esquema de objectos de negócio (ficheiros XSD) não inclui mapeamento por predefinição a partir do nome do conjunto de programas Java. A anotação **@XMLSchema** em package-info.java também não funciona. A única forma de criar um XSD com um espaço de nomes é utilizar as anotações **@XmlType** e **@XmlElement**. **@XmlElement** define o espaço de nomes de destino para o elemento global em tipos de JavaBeans.

- O assistente de importação EJB não cria ficheiros XSD para classes não relacionadas. A versão 2.0 não suporta a anotação `@XmlSeeAlso`, por isso, se a classe descendente não estiver referenciada directamente a partir da classe ascendente, não é criado um XSD. A solução para este problema é executar SchemaGen para essas classes descendentes.

SchemaGen é um utilitário de linha de comandos (localizado no directório `WPS_Install_Home/bin`) fornecido para criar ficheiros XSD para um determinado bean. Esses XSDs têm de ser copiados manualmente para o módulo para que a solução funcione.

#### *Selector de falha EJB:*

O selector de falha EJB determina se uma chamada para execução EJB resultou numa falha, excepção de tempo de execução ou numa resposta com êxito.

Se for detectada uma falha, o selector de falhas EJB devolve o nome da nativo ao tempo de execução da associação, para que a rotina de tratamento de dados JAX-WS possa converter o objecto de excepção num objecto de negócio.

Em de resposta com êxito (não-avaria), a associação de importação EJB monta uma matriz de objecto Java (`Object[]`) para devolver os valores.

- O primeiro elemento no `Object[]` é o valor de devolução da chamada para execução de método Java.
- Os valores subsequentes representam os parâmetros de entrada para o método.

Isto é necessário para suportar os tipos de parâmetros In/Out e Out.

Para cenários de excepção a associação monta um `Object[]` e o primeiro elemento representa a excepção detectada pelo método.

O selector de falha pode devolver qualquer um dos seguintes valores:

*Tabela 64. Valores de devolução*

Tipo	Valor de devolução	Descrição
Falha	<b>ResponseType.FAULT</b>	Devolvido quando o objecto <code>Object[]</code> transferido contém um objecto de excepção.
Excepção do tempo de execução	<b>ResponseType.RUNTIME</b>	Devolvido se o objecto de excepção não corresponder a qualquer um dos tipos de excepção declarados no método.
Resposta normal	<b>ResponseType.RESPONSE</b>	Devolvido em todos os outros casos.

Se o selector de falha devolver um valor de **ResponseType.FAULT**, o nome de falha nativo é devolvido. Este nome de falha é utilizado pela associação para determinar o nome de falha WSDL correspondente do modelo e para chamar a rotina de tratamento de dados correcta.

#### *Selector de função EJB:*

As associações EJB utilizam um selector de função de importação (para processamento externo) ou um selector de função de exportação (para processamento interno) para determinar o método EJB a invocar.

### **Selector de função de importação**

Para processamento externo, o selector de função de importação deriva o tipo de método EJB baseado no nome da operação chamada pelo componente SCA que está ligado à importação EJB. O selector de função procura a anotação `@WebMethod` na classe Java anotada no JAX-WS gerado no Integration Designer para determinar o nome da operação de destino associada.



- Se a anotação `@WebMethod` estiver presente, o selector de função utiliza a anotação `@WebMethod` para determinar o mapeamento de método de Java correcto para o método WSDL.
- Se a anotação `@WebMethod` estiver em falta, o selector de função assume que o nome do método de Java é o mesmo do nome da operação chamado.

**Nota:** Este selector de função é válido apenas para uma interface do tipo WSDL numa importação EJB, não é válido para uma interface do tipo de Java numa importação EJB.

O selector de função devolve um objecto `java.lang.reflect.Method` que representa o método da interface EJB.

O selector de função utiliza uma matriz de Java Object (`Object[]`) para conter a resposta do método de destino. O primeiro elemento no `Object[]` é um método de Java com o nome do WSDL, e o segundo elemento no `Object[]` é o objecto de negócio de entrada.

### Selector de função de exportação

Para processamento interno, o selector de função de exportação deriva o método de destino a ser chamado a partir do método de Java.

O selector de função de exportação mapeia o nome da operação de Java chamado pelo cliente EJB para o nome da operação na interface do componente de destino. O nome do método é devolvido como uma cadeia e resolvido pelo tempo de execução de SCA dependendo do tipo de interface do componente de destino.

## Associações EIS

As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é conseguida utilizando exportações EIS e importações EIS que suportem adaptadores de recurso JCA 1.5 e Websphere Adapters.

Os componentes SCA podem requerer que os dados sejam transferidos para ou de um EIS externo. Quando cria um módulo SCA que requer essa conectividade, incluirá (para além do componente SCA) uma importação ou exportação com uma associação EIS para comunicação com um EIS externo específico.

Os adaptadores de recurso no IBM Integration Designer são utilizados dentro do contexto de uma importação ou uma exportação. Desenvolve uma importação ou uma exportação com o assistente do serviço externo e, ao desenvolver, inclui o adaptador de recurso. Uma importação EIS que lhe permite chamar um serviço num sistema EIS, ou uma exportação que permite que uma aplicação num sistema EIS chame um serviço desenvolvido no IBM Integration Designer, são criadas com um adaptador de recurso. Por exemplo, criaria uma importação com o adaptador JD Edwards para chamar um serviço no sistema JD Edwards.

Quando utiliza o assistente do serviço externo, a informação da associação EIS é criada. Também pode utilizar outra ferramenta, o editor de montagem, para adicionar ou alterar a informação da associação. Consulte o tópico *Accessing external services with adapters* (Aceder a serviços externos com adaptadores) para mais informações.

Depois do módulo SCA que contém a associação EIS ser implementado no servidor, pode utilizar a consola administrativa para ver informações sobre a associação ou para configurar a associação.

### Descrição geral das associações EI:

A associação do Enterprise Information System (EIS), quando utilizada com um adaptador de recurso JCA, permite-lhe aceder a serviços num sistema de informação empresarial ou tornar os serviços disponíveis no EIS.



O exemplo seguinte mostra de que forma um módulo SCA chamado ContactSyncModule sincroniza informações de contacto entre um sistema Siebel e um sistema SAP.

1. O componente SCA chamado ContactSync escuta (através de uma exportação de aplicação EIS chamada Siebel Contact) por alterações para contactos Siebel.
2. O componente SCA ContactSync em si faz uso de uma aplicação SAP (através de uma importação de aplicação EIS) para actualizar respectivamente a informação de contacto SAP de forma.

Como as estruturas de dados utilizadas para armazenar contactos são diferentes nos sistemas Siebel e SAP, o componente SCA ContactSync tem de fornecer um mapeamento.

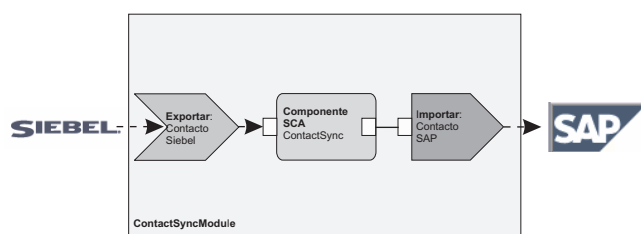


Figura 76. Fluxo de um sistema Siebel para um sistema SAP

A exportação de contactos Siebel e a importação de contactos SAP têm os adaptadores de recurso apropriados configurados.

### Funções-chave de associações ao EIS:

A importação EIS é uma importação do módulo Arquitectura do Componente de Serviço (SCA, Service Component Architecture) que permite que componentes desse mesmo módulo utilizem aplicações do EIS definidas fora do módulo SCA. É utilizada uma importação EIS para transferir dados do componente SCA para um EIS externo; é utilizada uma exportação EIS para transferir dados de um EIS externo para um módulo SCA.

### Importações

A função da importação EIS é estabelecer a comunicação entre componentes SCA e sistemas de EIS externo. As aplicações externas podem ser vistas como uma importação EIS. Neste caso, a importação EIS envia dados para o EIS externo e possui a opção de receber dados como resposta.

A importação EIS fornece componentes SCA com uma apresentação uniforme das aplicações exteriores ao módulo. Isto permite que os componentes comuniquem com um EIS externo, tal como o SAP, Siebel ou PeopleSoft, utilizando um modelo SCA consistente.

No lado da importação relativo ao cliente, existe uma interface, exposta pela aplicação da importação EIS, com um ou mais métodos, cada um deles considerando os objectos de dados como argumentos e valores devolvidos. No lado da implementação, existe uma Interface de Cliente Comum (CCI, Common Client Interface CCI) implementada por um adaptador de recurso.

A implementação do tempo de execução de uma importação EIS liga a interface do lado do cliente à CCI. A importação mapeia a chamada para execução do método pertencente à interface para a chamada para execução no CCI.

As associações são criadas em três níveis: a associação da interface, que depois utiliza as associações de métodos contidas, que, por sua vez, utilizam associações de dados.

A associação da interface relaciona a interface da importação com a ligação ao sistema EIS que fornece a aplicação. Isto reflecte o facto de o conjunto de aplicações, representado pela interface, ser fornecido pela ocorrência específica do EIS e de a ligação proporcionar o acesso a esta ocorrência. O elemento de

associação contém propriedades com informação suficiente para criar a ligação (estas propriedades fazem parte da ocorrência `javax.resource.spi.ManagedConnectionFactory`).

A associação de métodos associa o método que contém a interacção específica com o sistema EIS. Para a Arquitectura de Criptografia Java (JCA, Java Cryptography Architecture), a interacção é caracterizada pelo conjunto de propriedades da implementação da interface `javax.resource.cci.InteractionSpec`. O elemento de interacção da associação de métodos contém estas propriedades, juntamente com o nome da classe, proporcionando assim informação suficiente para efectuar a interacção. A associação de métodos utiliza associações de dados que descrevem à representação EIS o mapeamento do argumento e resultado do método da interface.

O modelo do tempo de execução para a importação EIS é o seguinte:

1. O método da interface de importação é chamado através da utilização do modelo de programação SCA.
2. O pedido, ao chegar à importação EIS, contém o nome do método e respectivos argumentos.
3. A importação cria primeiro uma implementação de associação de interface; depois, utilizando dados da associação de importação, cria uma `ConnectionFactory`, associando ambas. Ou seja, a importação pede o comando `setConnectionFactory` à associação da interface.
4. É então criada a implementação da associação de métodos correspondente ao método chamado.
5. A ocorrência `javax.resource.cci.InteractionSpec` é criada e preenchida; de seguida, são utilizadas associações de dados para associar os argumentos do método a um formato que seja entendido pelo adaptador de recurso.
6. A interface CCI é utilizada para executar a interacção.
7. Quando o pedido regressa, a associação de dados é utilizada para criar o resultado da chamada, sendo o resultado devolvido ao autor do pedido.

## Exportações

A função da exportação EIS é estabelecer a comunicação entre componentes SCA e sistemas de EIS externos. As aplicações externas podem ser vistas como uma exportação EIS. Neste caso, a aplicação externa envia os dados na forma de notificações periódicas. Pode-se pensar numa exportação EIS como se fosse uma aplicação de subscrição que recebe um pedido externo de um EIS. O componente SCA que utiliza a exportação EIS trata-a como uma aplicação local.

A exportação EIS proporciona aos componentes SCA uma vista uniforme das aplicações exteriores ao módulo. Isto permite que os componentes comuniquem com um EIS, como o SAP, Siebel ou PeopleSoft, utilizando um modelo SCA consistente.

A exportação apresenta uma implementação de serviço de escuta, que recebe pedidos do EIS. O serviço de escuta implementa um serviço de escuta específico para o adaptador de recurso. A exportação contém também uma interface de implementação de componentes, exposta ao EIS através da exportação.

A implementação do tempo de execução de uma exportação EIS liga o serviço de escuta à interface de implementação do componente. A exportação mapeia o pedido feito pelo EIS para a chamada da operação adequada do componente. As associações são criadas em três níveis: a associação do sistema de escuta, que depois utiliza um método de associação nativo, que, por sua vez, utiliza uma associação de dados.

A associação do sistema de escuta relaciona o sistema de escuta que recebe os pedidos com o componente exposto através da exportação. A definição da exportação contém o nome do componente, que o tempo de execução localiza e para o qual reencaminha os pedidos.

A associação do método nativo associa o método nativo ou o tipo de evento recebido pelo serviço de escuta, à operação implementada pelo componente exposto por via da exportação. Não existe qualquer

relação entre o método chamado no sistema de escuta e o tipo de evento; todos os eventos chegam através de um ou mais métodos do serviço de escuta. A associação do método nativo utiliza o selector de função definido na exportação para extrair o nome do método nativo dos dados internos e associações de dados para associar o formato de dados do EIS a um formato entendido pelo componente.

O modelo do tempo de execução para uma exportação EIS é o seguinte:

1. O pedido de EIS desencadeia a chamada do método na implementação do sistema de escuta.
2. O serviço de escuta localiza e chama a exportação, transferindo para esta todos os argumentos de chamada.
3. A exportação cria a implementação da associação do serviço de escuta.
4. A exportação instância o selector de função e define-o na associação do sistema de escuta.
5. A exportação inicializa as associações do método nativo e adiciona-as à associação do sistema de escuta. Para cada associação do método nativo, são também inicializadas as associações de dados.
6. A exportação chama a associação do sistema de escuta.
7. A associação do sistema de escuta localiza componentes exportados e utiliza o selector de função para obter o nome do método.
8. Este nome é utilizado para localizar a associação do método nativo, que depois chama o componente-destino.

O estilo de interação do adaptador permite que a associação da exportação EIS chame o componente-destino de forma assíncrona (a forma predefinida) ou síncrona.

### **Adaptadores de recurso**

Ao desenvolver uma importação ou exportação com o assistente do serviço externo, está a incluir um adaptador de recurso. Os adaptadores que vêm com IBM Integration Designer utilizados para aceder aos sistemas CICS, IMS, JD Edwards, PeopleSoft, SAP e Siebel destinam-se apenas a fins de desenvolvimento e teste. Isto significa que os irá utilizar para desenvolver e testar as suas aplicações.

Uma vez implementada a sua aplicação, irá precisar de adaptadores de tempo de execução licenciados para a executar. No entanto, ao criar o seu serviço, é possível incorporar o adaptador no mesmo. O licenciamento do adaptador pode permitir-lhe utilizar o adaptador incorporado como adaptador de tempo de execução licenciado. Estes adaptadores estão em conformidade com a Java EE Connector Architecture (JCA 1.5). A JCA, um padrão aberto, é o padrão Java EE para a conectividade EIS. A JCA fornece um contexto gerido, ou seja, a Qualidade de Serviço (QoS) é fornecida pelo servidor da aplicação que oferece a gestão do ciclo de vida e segurança às transacções. Estão também em conformidade com a especificação Identificação de metadados de empresa, com a excepção do Adaptador de recurso IBM CICS ECI e do IBM IMS Connector for Java.

O WebSphere Business Integration Adapters, um conjunto mais antigo de adaptadores, também é suportado pelo assistente.

### **Recursos Java EE**

O módulo EIS module, um módulo SCA que segue o padrão do módulo EIS, pode ser implementado na plataforma Java EE.

A implementação do módulo EIS na plataforma Java EE resulta numa aplicação pronta a ser iniciada, compactada como ficheiro EAR e implementada no servidor. São criados todos os artefactos e recursos Java EE, a aplicação é configurada e fica pronta a ser executada.

## Propriedades dinâmicas da Especificação de interação JCA e Especificação de ligação JCA:

A associação EIS pode aceitar a entrada para as InteractionSpec e ConnectionSpec especificadas, utilizando um objecto de dados descendente que acompanha a carga útil. Isto permite interações dinâmicas de pedido-resposta com um adaptador de recurso através da InteractionSpec e autenticação de componente através da ConnectionSpec.

A javax.cci.InteractionSpec traz informação sobre a forma como o pedido de interação com o adaptador de recurso deve ser processado. Também pode trazer informações sobre a forma como a interação foi alcançada depois do pedido. Estas comunicações bidireccionais através das interações são referidas, por vezes, como *conversações*.

A associação EIS espera que a carga útil, que será o argumento para o adaptador de recurso, contenha um objecto de dados descendente chamado **propriedades**. Este objecto de dados de propriedades vai conter pares nome-valor, com o nome das propriedades da especificação de interação num formato específico. As regras de formatação são:

- Os nomes têm de começar pelo prefixo **IS**, seguido pelo nome da propriedade. Por exemplo, uma especificação de interação com uma propriedade JavaBeans chamada **InteractionId** especifica o nome da propriedade como **ISInteractionId**.
- O par nome-valor representa o nome e o valor do tipo simples da propriedade de especificação de interação.

Neste exemplo, uma interface especifica que a entrada de uma operação é um **Account** objecto de dados. Esta interface chama uma aplicação de associação de importação EIS com a intenção de enviar e receber uma propriedade InteractionSpec dinâmica chamada **workingSet** com o valor **xyz**.

Os gráficos comerciais ou objectos de negócio no servidor contêm um objecto de negócio de propriedades **properties (propriedades)** subjacente que permite o envio de dados específicos de protocolo com a carga útil. Este objecto de negócio de **propriedades** está integrado e não necessita de ser especificado no esquema XML ao construir um objecto de negócio. Só precisa de ser criado e utilizado. Se tiver os seus próprios tipos de dados definidos com base num esquema XML, necessita de especificar um elemento de **properties** que contenha o seu par nome-valor esperado.

```
BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Conector para interfaces de estilo translineadas doc-lit,
//avançar para carga útil para não doc-lit
DataObject docLitWrapper = dataFactory.createByElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Criar a carga útil.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Execute a definição da carga útil
```

```
//Construir dados de propriedades para interação dinâmica
```

```
DataObject properties = account.createDataObject("properties");
```

Para o nome **workingSet**, defina o valor esperado(**xyz**).

```
properties.setString("ISworkingSet", "xyz");
```

```
//Chamar o serviço com argumento
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Obter propriedade devolvida
DataObject retProperties = result.getDataObject("properties");

String workingset = retProperties.getString("ISworkingSet");
```

Pode utilizar as propriedades ConnectionSpec para uma autenticação de componente dinâmica. Aplicam-se as mesmas regras, excepto que o prefixo do nome da propriedade precisa de ser **CS** (em vez de **IS**). As propriedades ConnectionSpec não são bidireccionais. O mesmo objecto de dados de **propriedades** pode conter propriedades IS e CS.

Para utilizar propriedades ConnectionSpec, defina o **resAuth** especificado na associação de importação para **Aplicação**. Certifique-se também de que o adaptador de recurso suporta a autorização de componente. Consulte o capítulo 8 da Especificação de arquitectura de Conector J2EE para mais detalhes.

### **Clientes externos com associações EIS:**

O servidor pode enviar mensagens a, receber mensagens de clientes externos utilizando associações EIS.

Um cliente externo, por exemplo um portal da web ou um EIS, necessita de enviar uma mensagem para um módulo SCA para o servidor ou necessita de ser chamado por um componente a partir do servidor.

O cliente chama a importação EIS com qualquer outra aplicação, utilizando a Interface de chamada dinâmica (DII, Dynamic Invocation Interface) ou a interface Java.

1. O cliente externo cria uma instância do ServiceManager e procura a importação EIS utilizando o seu nome de referência. O resultado da procura é a implementação da interface de serviço.
2. O cliente cria um argumento de entrada, um objecto de dados genérico, criado dinamicamente, utilizando o esquema de objecto de dados. Este passo é efectuado utilizando a implementação de interface DataFactory do objecto de dados de serviço.
3. O cliente externo chama o EIS e obtém os resultados pedidos.

Outra opção é o cliente poder chamar a importação EIS utilizando a interface Java.

1. O cliente cria uma instância do ServiceManager e procura a importação EIS utilizando o seu nome de referência. O resultado da procura é uma interface Java da importação EIS.
2. O cliente cria um argumento de entrada e o objecto de dados escrito.
3. O cliente chama o EIS e obtém os resultados pedidos.

A interface de exportação EIS define a interface do componente SCA exportado que está disponível para as aplicações EIS externas. Esta interface pode ser imaginada como a interface que uma aplicação externa (tal como SAP ou PeopleSoft) irá chamar através da implementação do tempo de execução da aplicação de exportação EIS.

A exportação utiliza EISExportBinding para associar serviços exportados à aplicação EIS externa. Permite que o utilizador subscreva uma aplicação contida no seu módulo SCA para escutar pedidos de serviço EIS. A associação de exportação EIS especifica o mapeamento entre a definição de eventos de entrada como é compreendida pelo adaptador de recurso (utilizando interfaces do Java EE Connector Architecture) e a chamada de operações SCA.

O EISExportBinding necessita de serviços EIS externos para ser basearem no nos contratos de entrada do Java EE Connector Architecture 1.5. O EISExportBinding necessita que uma rotina de tratamento de dados ou associação de dados seja especificada ao nível da associação ou ao nível do método.

## Associações JMS

Um fornecedor de Serviço de Mensagens Java (JMS - Message Service) permite mensagens com base na API de Java Messaging Service API e modelo de programação. Disponibiliza fornecedores de ligação JMS para criar ligações para destinos JMS e para enviar e receber mensagens.

As associações JMS podem ser utilizadas quando interage com a associação de fornecedor de barramento de integração de serviço (SIB, Service Integration Bus) e são compatíveis com JMS e JCA 1.5.

Pode utilizar as associações de importação e de exportação JMS e um módulo de arquitectura de componente de serviço (SCA) para chamar e receber mensagens de sistemas JMS externos.

As associações de importação e exportação JMS fornecem integração com aplicações JMS utilizando o fornecedor de JMS SIB baseado em JCA 1.5 que faz parte do WebSphere Application Server. Não são suportados outros adaptadores de recursos JMS baseados em JCA 1.5

### Descrição geral de associações JMS:

As associações JMS fornecem conectividade entre o ambiente de Arquitectura do Componente de Serviço (SCA, Service Component Architecture) e sistemas de JMS.

### Associações JMS

Os principais componentes de ambas as associações de importação JMS e de exportação JMS são:

- Adaptador de recurso: permite conectividade bidireccional gerida entre um módulo SCA e sistemas JMS externos
- Ligações: encapsulam uma ligação virtual entre uma aplicação cliente e uma aplicação fornecedora
- Destinos: utilizados por um cliente para especificar o destino de mensagens que produz ou a origem de mensagens que consome
- Dados de autenticação: utilizados para proteger o acesso à associação

### Funções-chave de associações JMS

#### Cabeçalhos especiais

As propriedades de cabeçalho especial são utilizadas em importações e exportações JMS para dizer ao destino como processar a mensagem.

Por exemplo, TargetFunctionName faz a correlação a partir do método nativo para o método de operação.

#### Recursos Java EE

É criado um número de recursos Java EE quando as importações e exportações JMS são implementadas no ambiente Java EE.

#### ConnectionFactory

Utilizado por clientes para criar uma ligação ao fornecedor JMS.

#### ActivationSpec

As importações utilizam este comando para receber a resposta a um pedido; as exportações usam-no ao configurarem os pontos finais da mensagem que representam serviços de escuta da mensagem nas suas interacções com o sistema de mensagens.

#### Destinos

- Destino de envio: numa importação, é para onde é enviado o pedido ou mensagem de saída; numa exportação, é o destino para onde será enviada a mensagem de resposta, caso o campo de cabeçalho JMSReplyTo não se lhe sobreponha na mensagem de entrada.



- Destino de recepção: onde deve ser colocada a mensagem de entrada; nas importações, trata-se de uma resposta; nas exportações, trata-se de um pedido.
- Destino da chamada de retorno: destino do sistema SCA JMS utilizado para armazenar informações de correlação. Não utilizar este destino para leitura nem escrita.

A tarefa de instalação cria a `ConnectionFactory` e três destinos. Cria também o `ActivationSpec` para permitir que o serviço de escuta da mensagem do tempo de execução capte respostas no destino de recepção. As propriedades destes recursos vêm especificadas no ficheiro de importação ou exportação.

### Integração JMS e adaptadores de recurso:

O Fornecedor de Serviços Java (JMS - Java Message Service) fornece integração através de um adaptador de recurso baseado em JCA 1.5 JMS disponível. O suporte completo para a integração JMS é fornecido para o adaptador de recurso JMS de barramento de integração de serviço (SIB - Service Integration Bus).

Utilize um fornecedor JMS para adaptador de recurso JCA 1.5 quando quiser integrar com um sistema JMS externo compatível com JCA 1.5. Os serviços externos compatíveis com JCA 1.5 podem receber mensagens e enviar mensagens para integrarem com componentes da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) utilizando o adaptador de recurso JMS SIB.

A utilização de outros adaptadores de recurso JCA 1.5 específicos de fornecedor não é suportada.

### Associações de importação e exportação JMS:

Pode fazer com que os módulos SCA interajam com serviços fornecidos por aplicações JMS externas através da utilização de associações de importação e exportação JMS.

### Associações de importação JMS

As ligações ao fornecedor de JMS associado de destinos JMS são criadas através da utilização de um fornecedor de ligação JMS. Utilize objectos administrativos de fornecedor de ligação para gerir fornecedores de ligação JMS para o fornecedor de serviço de mensagens predefinido.

A interacção com sistemas JMS externos inclui a utilização de destinos para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação de JMS, dependendo do tipo de operação chamada:

- Unidireccional: A importação JMS coloca uma mensagem no destino de envio configurado na associação de importação. Não está definido nada no campo `replyTo` do cabeçalho JMS.
- Bidireccionalidade (resposta ao pedido): A importação JMS coloca uma mensagem no destino de envio e depois mantém a resposta que recebe do componente SCA.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação de mensagem de pedido. A associação de importação também pode ser configurada para utilizar um destino de resposta dinâmico para correlacionar respostas com pedidos. É criado um destino temporário para cada pedido e a importação utiliza esse destino para receber a resposta.

O destino receive é definido na propriedade do cabeçalho `replyTo` da mensagem de saída. É implementada uma escuta de mensagem para escutar o destino de recepção, e quando uma mensagem é recebida, a escuta de mensagem passa a resposta de volta para o componente.



É possível especificar propriedades de cabeçalho dinâmicas e estáticas para ambos os cenários de utilização. As propriedades estáticas podem ser definidas a partir da associação de método de importação JMS. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 32 na página 119 ilustra a forma como a importação está ligada ao serviço externo.

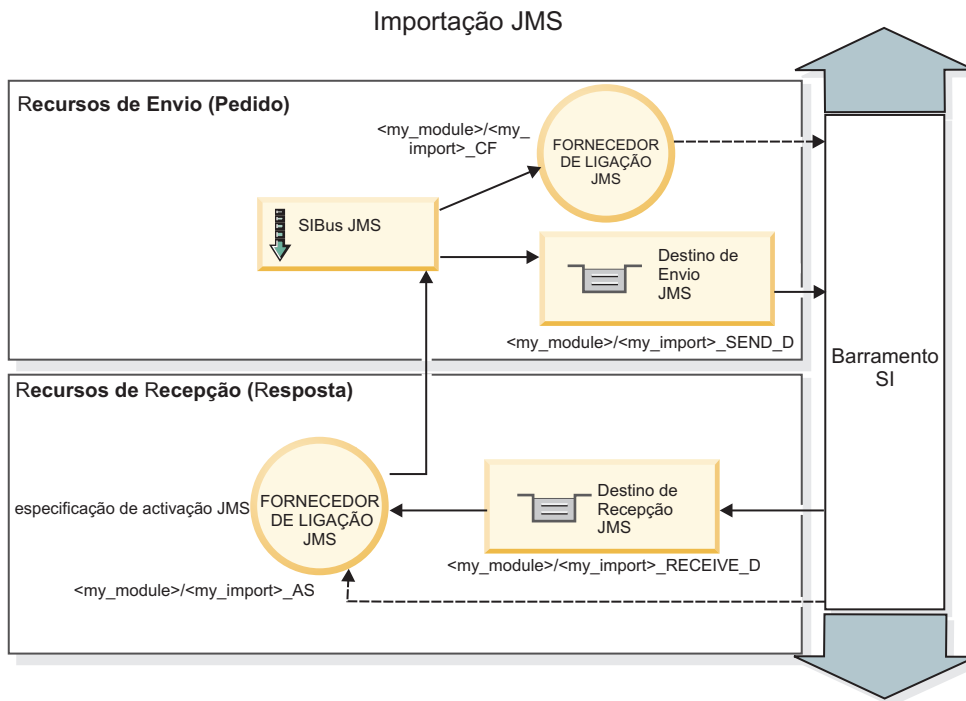


Figura 77. Recursos de associação de importação JMS

### Associações de exportação JMS

As associações de exportação JMS fornecem os meios para os módulos SCA fornecerem serviços para aplicações JMS externas.

A ligação que faz parte de uma exportação JMS é uma especificação de activação configurável.

Uma exportação JMS tem destinos de envio e de recebimento.

- O destino receive é onde a mensagem recebida do componente de destino deve ser colocada.
- O destino send é onde a resposta será enviada, excepto se a mensagem recebida o tiver substituído utilizando a propriedade de cabeçalho `replyTo`.

Uma escuta de mensagem é implementada para escutar pedidos de entrada no destino receive especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. O destino especificado no campo `replyTo` da mensagem recebida substitui o destino especificado em send.

A Figura 33 na página 120 ilustra como o solicitador externo está ligado à exportação.

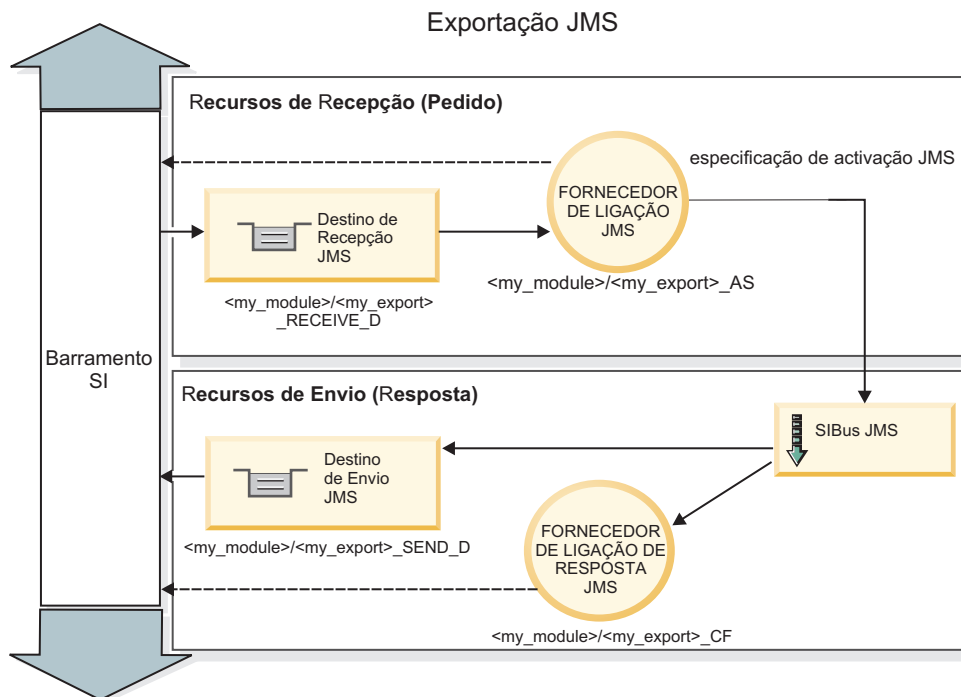


Figura 78. Recursos de associação de exportação JMS

### Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

### Cabeçalho de sistema JMS

O cabeçalho de sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos encontrados tipicamente num cabeçalho JMS. Apesar de estes poderem ser modificados na mediação (ou ContextService), alguns campos de cabeçalho de sistema JMS definidos no SMO não serão propagados na mensagem JMS de saída, uma vez que são substituídos por valores de sistema ou estáticos.

Os campos de chave no cabeçalho de sistema JMS que podem ser actualizados numa mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidos específicos
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não-persistente, a predefinição é persistente)
- **JMSPriority** - valor de prioridade (0 a 9, a predefinição é JMS\_Default\_Priority)

### Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista Propriedades. As propriedades podem ser adicionadas, actualizadas ou eliminadas numa mediação ou através da utilização da API ContextService.

As propriedades também podem ser definidas estaticamente na associação JMS. As propriedades que estiverem definidas estaticamente substituem definições (com o mesmo nome) que estejam definidas dinamicamente.

As propriedades de utilizador propagadas de outras associações (por exemplo, uma associação HTTP) serão produzidas na associação JMS como propriedades JMS.

### Definições de propagação de cabeçalho

A propagação do cabeçalho de sistema JMS e propriedades quer da mensagem JMS interna para componentes na direcção do fluxo ou de componentes na direcção oposta ao fluxo para a mensagem JMS de saída podem ser controlados pelo sinalizador Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) na associação.

Quando o Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) está definido, a informação de cabeçalho pode fluir para a mensagem ou para o componente de destino, tal como descrito na lista seguinte:

- Pedido de exportação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.
- Resposta de exportação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de exportação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de exportação JMS.
- Pedido de importação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de importação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de importação JMS.
- Resposta de importação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.

### Esquema de correlação de destino de resposta dinâmico temporário JMS:

O esquema de correlação de destino de resposta dinâmico temporário dá origem a uma fila dinâmica única ou tópico a ser criado para cada pedido enviado.

O destino de resposta estático especificado na importação é utilizado para derivar a natureza da fila de destino dinâmico temporário ou tópico. Isto é definido no campo **ReplyTo (Responder a)** do pedido e a importação JMS escuta por respostas nesse destino. Quando a resposta é recebida, ela é recolocada na fila do destino de resposta estática para processamento assíncrono. O campo **CorrelationID** da resposta não é utilizado e não precisa de ser definido.

### Questões transaccionais

Quando um destino dinâmico temporário está em utilização, a resposta tem de ser consumida no mesmo módulo que o da resposta enviada. O pedido tem de ser enviado fora da transacção global e tem de ser consolidado antes de ser recebido pelo serviço do programa emissor para que uma resposta tem seja devolvida.

## Persistência

As filas dinâmicas temporárias são entidades de duração breve e não garantem o mesmo nível de persistência associado a uma fila estática ou tópico. Uma fila dinâmica temporária ou tópico não sobrevivem a um reinício de servidor nem as mensagens. Depois de uma mensagem ter sido recolocada na fila do destino de resposta estático, ela retém a persistência definida na mensagem.

## Tempo limite

A importação espera para receber a resposta no destino de resposta dinâmico temporário durante um tempo fixo. Este intervalo de tempo é retirado do qualificador do momento de expiração de resposta SCA, se este estiver definido, de outro modo o tempo utilizado como predefinição é de 60 segundos. Se o tempo de espera for excedido, a importação apresenta uma `ServiceTimeoutRuntimeException`.

## Clientes externos:

O servidor pode enviar mensagens a ou receber mensagens de clientes externos utilizando associações JMS.

Um cliente externo (como um portal da web ou um sistema de informação empresarial) pode enviar uma mensagem a um módulo SCA no servidor ou pode ser chamado por um componente a partir do servidor.

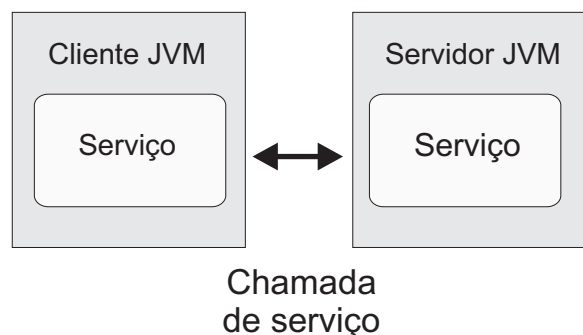
Os componentes de exportação JMS fornecem escutas de mensagem para escutar pedidos que entram no destino especificado na associação de exportação. O destino especificado no campo de envio é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações com a associação de exportação.

As importações JMS interagem com clientes externos ao enviar mensagens e ao receber mensagens de filas JMS.

### *Trabalhar com clientes externos:*

Um cliente externo (isto é, que está fora do servidor) poderá precisar de interagir com uma aplicação instalada no servidor.

Imagine um cenário muito simples, em que o cliente externo quer interagir com uma aplicação do servidor. O exemplo ilustra um cenário típico muito simples.



*Figura 79. Cenário de um caso de uso simples: o cliente externo interage com a aplicação de servidor.*

A aplicação SCA inclui uma exportação com uma associação JMS; o que faz com que a aplicação seja disponibilizada a clientes externos.

Quando tem um cliente externo numa máquina virtual Java (JVM virtual machine) separada do seu servidor, existem vários passos a tomar, de modo a estabelecer uma ligação e interagir com a exportação JMS. O cliente obtém um InitialContext (Contexto Inicial) com os valores correctos e, depois, pesquisa os recursos no JNDI. Nessa altura, o cliente utiliza a JMS 1.1 especificação do cliente para aceder a destinos, e às mensagens enviadas e recebidas nos destinos.

Os nomes JNDI predefinidos dos recursos criados automaticamente pelo tempo de execução estão catalogados no tópico de configuração desta secção. Contudo, se criou recursos previamente, utilize esses nomes de JNDI.

1. Configure os destinos JMS e o fornecedor de ligação para enviar mensagens.
2. Certifique-se de que o contexto JNDI, a porta do adaptador de recurso SIB, e a porta de envio de mensagens estão correctas.

O servidor utiliza algumas portas predefinidas mas, caso existam mais servidores instalados nesse sistema, são criadas portas alternadas na hora da instalação, para evitar conflitos com ocorrências de outros servidores. Pode utilizar a consola administrativa para determinar que portas o seu servidor está a utilizar. Aceda a **Servers (Servidores) > Application Servers (Servidores de Aplicação) > your\_server\_name > Configuration (Configuração)** e fazer clique **Ports (Portas)** sob **Communication (Comunicação)**. Em seguida, pode editar a porta que está a ser utilizada.

3. O cliente obtém um contexto inicial com os valores correctos e, depois, pesquisa os recursos no JNDI.
4. Ao utilizar as especificações JMS 1.1, o cliente acede a destinos, e às mensagens enviadas e recebidas nos destinos.

### Resolução de problemas de associação JMS:

Poderá diagnosticar e corrigir problemas através de associações JMS.

### Excepções de implementação

Em resposta a várias condições de erro, a implementação de importações e exportações JMS poderá devolver um ou dois tipos de excepções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter o original excepção (JMSException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

### Principais condições de erro

As principais condições de erro das associações JMS são determinadas pela semântica transaccional, pela configuração do fornecedor JMS, ou por referência a um comportamento existente noutros componentes. As principais condições de erro incluem:

- Erro ao ligar-se ao fornecedor JMS ou ao destino.  
Erro ao ligar-se ao fornecedor JMS para receber mensagens irá resultar num erro ao iniciar a escuta de mensagens. Esta condição será registada no registo do WebSphere Application Server. Mensagens que persistam irão permanecer no destino, até serem obtidas com sucesso (ou até expirem).  
Um erro ao ligar-se ao fornecedor JMS, para enviar mensagens de saída, irá causar a remoção de alterações da transacção que controla o envio.
- Um erro ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.

Um erro na associação de dados, ou na rotina de tratamento de dados, causa uma remoção de alterações do controlo de transacção do funcionamento.

- Um erro no envio da mensagem de saída.

Um erro no envio de uma mensagem causa uma remoção de alterações da transacção relevante.

- Mensagens de resposta tardias múltiplas ou inesperadas.

A importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Também o período de hora válida, em que se pode receber uma resposta, é determinado qualificador do Limite de Validade da Resposta SCA no pedido. Quando chega uma resposta, ou quando é ultrapassado o momento de expiração, apaga-se a gravação da correlação. Caso as mensagens de resposta cheguem de forma inesperada, ou tardia, é activada a Excepção de tempo de execução do serviço.

- A excepção do tempo de execução de espera do serviço causado por resposta tardia aquando da utilização do esquema de correlação do destino da resposta de dinâmica temporária.

O tempo de espera da importação JMS tem um período de tempo determinado pelo qualificador do limite de validade da resposta SCA ou, caso isto não esteja definido, passará a estar predefinido dentro de 60 segundos.

### **As mensagens JMS baseadas em SCA que não aparecem no gestor de eventos falhado.**

Se as mensagens SCA originadas através de uma interacção JMS falharem, poderá encontrá-las no gestor de eventos falhados. Se estas mensagens não forem apresentadas no gestor de eventos falhados, certifique-se de que o destino SIB subjacente ao destino JMS tem um valor máximo de entregas falhadas superior a 1. Definir este valor para 2, ou superior, activa a interacção com o gestor de eventos falhados, durante as chamadas para execução SCA para as associações JMS.

### **Excepções de processamento:**

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal excepção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

Quando qualquer excepção de tempo de execução, incluindo uma **DataBindingException**, é lançada:

- Se o fluxo de mediação está configurado para ser transaccional, a mensagem JMS, por predefinição, é guardada no Gestor de eventos falhados para repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no Gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a excepção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer excepção da rotina de tratamento de dados é reiniciada ciclicamente numa excepção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

## Associações genéricas JMS

A Associação JMS genérico fornece conectividade a fornecedores compatíveis com JMS 1.1 de terceiros. A operação das associações JMS Genérico é semelhante à de associações JMS.

O serviço fornecido através de uma associação JMS permite que um módulo da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) faça chamadas ou receba mensagens de sistemas externos. O sistema pode ser um sistema JMS externo.

A Associação JMS genérico fornece integração com fornecedores JMS compatíveis não JCA 1.5 que suportem JMS 1.1 e implementem o JMS Application Server Facility opcional. A associação JMS Generic suporta esses fornecedores JMS (incluindo Oracle AQ, TIBCO, SonicMQ, WebMethods, e BEA WebLogic) que não suportam JCA 1.5 mas suportam Application Server Facility da especificação JMS 1.1. O fornecedor JMS incorporado no WebSphere (SIBJMS), que é um fornecedor JMS JCA 1.5, não é suportado por esta associação; ao utilizar esse fornecedor, utilize "Associações JMS" na página 116.

Utilize esta Associação genérica ao integrar com um sistema baseado em JMS compatível com não JCA 1.5 dentro de um ambiente SCA. As aplicações de destino externas podem então receber mensagens e enviar mensagens para integrar com um componente SCA.

### Descrição geral das associações JMS genérico:

As associações JMS genérico são associações JMS não JCA que fornecem conectividade entre o ambiente de Arquitectura do Componente de Serviço (SCA, Service Component Architecture) e sistemas de JM compatíveis com JMS 1.1 e que implementem o JMS Application Server Facility opcional.

### Associações genéricas JMS

Os principais aspectos das associações de importação e exportação JMS genérico incluem o seguinte:

- Porta de recepção: permite que fornecedores JMS baseados em não JCA recebam mensagens e as enviem para um bean orientado por mensagens (MDB - Message Driven Bean)
- Ligações: encapsulam uma ligação virtual entre uma aplicação cliente e uma aplicação fornecedora
- Destinos: utilizados por um cliente para especificar o destino de mensagens que produz ou a origem de mensagens que consome
- Dados de autenticação: utilizados para proteger o acesso à associação

### Associações de importação JMS genérico

As associações de importação JMS genérico permitem que componentes dentro do módulo SCA comuniquem com serviços fornecidos por fornecedores JMS compatíveis com não JCA 1.5.

A parte de ligação de uma importação JMS é um fornecedor de ligação. Um fornecedor de ligação, o objecto que um cliente utiliza para criar uma ligação a um fornecedor, encapsula um conjunto de parâmetros de configuração da ligação definido por um administrador. Cada fornecedor de ligação é uma instância da interface ConnectionFactory, QueueConnectionFactory ou TopicConnectionFactory.

A interacção com sistemas JMS externos inclui a utilização de destinos para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação JMS Genérico, dependendo do tipo de operação chamada:

- Unidireccional: A importação JMS genérico coloca uma mensagem no destino de envio configurado na associação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- Bidireccional (pedido-resposta): a importação JMS genérico coloca uma mensagem no destino de envio e depois persiste a resposta que recebe do componente SCA.



O destino receive é definido na propriedade do cabeçalho replyTo da mensagem de saída. Um bean orientado por mensagens (MDB) é implementado para escutar no destino de recepção, e quando uma mensagem é recebida, o MDB transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação da mensagem de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação da mensagem de pedido.

É possível especificar propriedades de cabeçalho dinâmicas e estáticas para ambos os cenários de utilização. As propriedades estáticas podem ser definidas a partir da associação de método de importação JMS genérico. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS genérico é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS genérico de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 35 na página 126 ilustra a forma como a importação está ligada ao serviço externo.

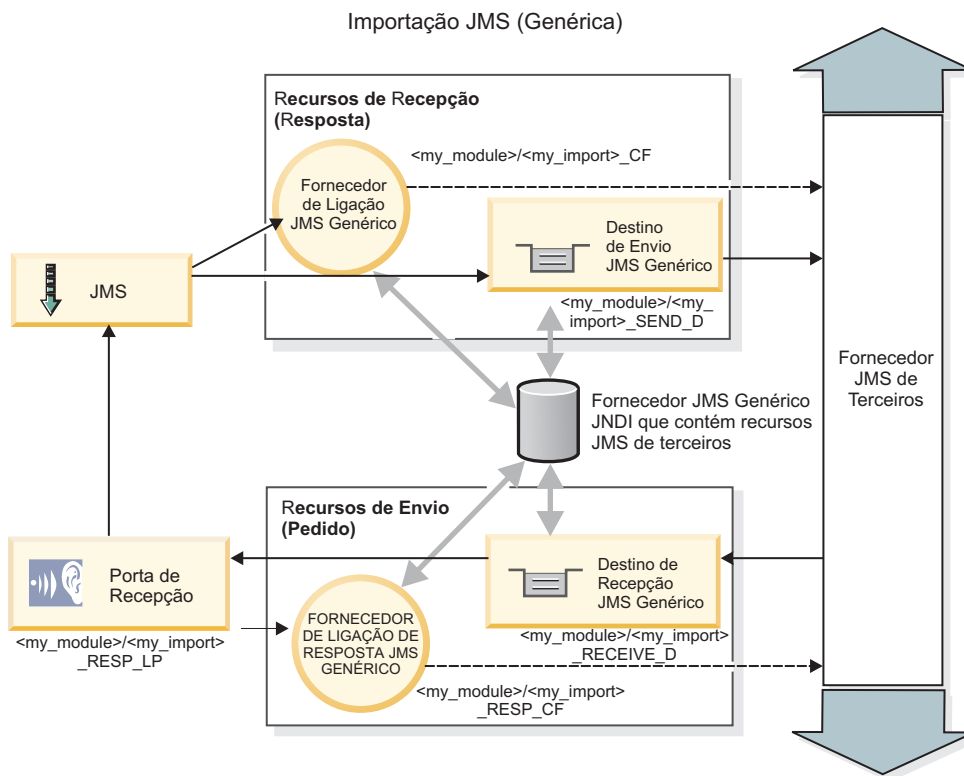


Figura 80. Recursos de associação de importação JMS genérico

### Associações de exportação JMS genérico

As associações de exportação JMS genérico fornecem os meios para módulos SCA fornecerem serviços a aplicações JMS externas.

A parte de ligação de uma exportação JMS é composta por um fornecedor de ligação e uma porta de recepção.

Uma exportação JMS Genérico tem destinos de envio e de recepção.

- O destino receive é onde a mensagem recebida do componente de destino deve ser colocada.
- O destino send é onde a resposta será enviada, excepto se a mensagem recebida o tiver substituído utilizando a propriedade de cabeçalho replyTo.

Um bean orientado por mensagens é implementado para escutar pedidos de entrada no destino receive especificado na associação de exportação.

- O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta.
- O destino especificado no campo replyTo da mensagem recebida substitui o destino especificado no campo send.
- Para cenários de pedido/resposta, a associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que a resposta copie o message ID (identificador da mensagem) do pedido para o campo correlation ID (identificador de correlação) da mensagem de resposta (predefinição) ou a mensagem pode copiar o correlation ID do pedido para o campo correlation ID da mensagem de resposta.

A Figura 36 na página 127 ilustra como o solicitador externo está ligado à exportação.

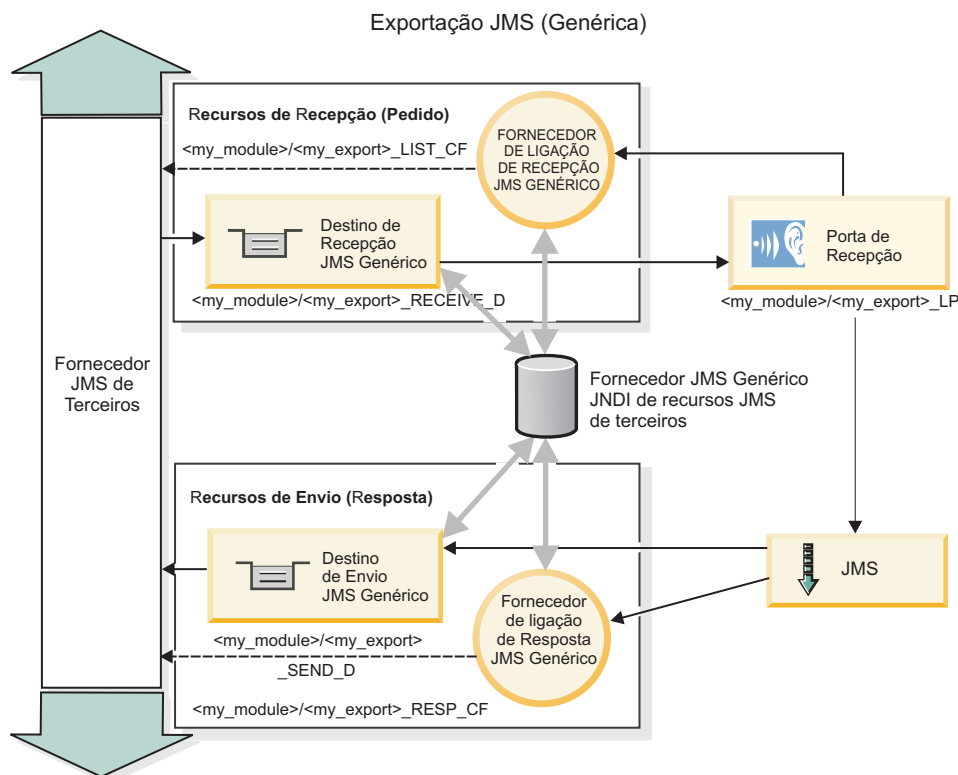


Figura 81. Recursos de associação de exportação JMS genérico

### Funções-chave de associações JMS Genérico:

Os componentes da associação de importação e exportação JMS Genérico são consistentes com os do JMS incorporado e associações de importação MQ JMS do WebSphere. As funções-chave incluem definições de cabeçalho e acesso a recursos Java EE existentes. No entanto, dada a sua natureza genérica, não existem opções de conectividade específicas para o fornecedor JMS e esta associação tem uma capacidade limitada de gerar recursos no momento da implementação e instalação.

### Generic imports (Importações genéricas)

Tal como a aplicação de importação MQ JMS, a implementação do JMS Genérico é assíncrona e suporta três invocações: unidireccional, bidireccional (também chamada pedido-resposta) e chamada de retorno.

Quando a importação JMS é implementada, é implementado um bean orientado por mensagens (MDB) fornecido pelo ambiente do tempo de execução. O MDB procura obter respostas à mensagem de pedido. O MDB está associado ao (procura obter respostas no) destino enviado com o pedido no campo de cabeçalho `replyTo` da mensagem JMS.

### **Generic exports (Exportações genéricas)**

As associações de exportação JMS genéricas diferem das associações de exportação EIS no modo de processamento da devolução do resultado. Uma exportação JMS Genérico envia explicitamente a resposta ao destino `replyTo` especificado na mensagem de entrada. Se não estiver nenhum destino especificado, é utilizado o destino de envio.

Quando é implementada a exportação de JMS Genérico, é implementado um bean orientado por mensagens (um MDB diferente do utilizado para as importações de JMS Genérico. Escuta os pedidos de chegada no destino de recepção e envia-os para o tempo de execução SCA para serem processados.

### **Cabeçalhos especiais**

As propriedades dos cabeçalhos especiais são utilizadas nas importações e exportações de JMS Genérico para informar a associação-destino sobre como processar a mensagem.

Por exemplo, a propriedade `TargetFunctionName` é utilizada pelo selector de função predefinido para identificar o nome da operação na interface de exportação que está a ser chamada.

**Nota:** A associação de importação pode ser configurada para definir o cabeçalho `TargetFunctionName` para o nome de operação de cada operação.

### **Recursos Java EE**

São criados vários recursos Java EE quando é implementada uma associação JMS num ambiente Java EE.

- Porta de recepção para serviço de escuta no destino de recepção (resposta) – apenas bidireccional – para importações e no destino de recepção (pedido) para exportações
- Fornecedor de ligação a JMS Genérico para `outboundConnection` (importação) e `inboundConnection` (exportação)
- Destino de JMS Genérico para os destinos de envio (importação) e recepção (exportação; apenas bidireccional)
- Fornecedor de ligação ao JMS Genérico para `responseConnection` (apenas bidireccional e opcional; em alternativa, é utilizada a `outboundConnection` para importações e a `inboundConnection` para exportações)
- Destino de JMS Genérico para os destinos de envio (importação) e recepção (exportação) – apenas bidireccional.
- Destino JMS da chamada de retorno do fornecedor predefinido do serviço de mensagens utilizado para aceder ao destino da fila da chamada de retorno SIB (apenas bidireccional)
- Fornecedor de ligação JMS da chamada de retorno do fornecedor predefinido do serviço de mensagens utilizado para aceder ao destino (apenas bidireccional) JMS da chamada de retorno
- Destino da fila de chamadas de retorno SIB utilizado para armazenar informações sobre a mensagem de pedido a utilizar durante o processamento da resposta (apenas bidireccional)

A tarefa de instalação cria a `ConnectionFactory`, os três destinos e a `ActivationSpec` a partir das informações nos ficheiros de importação e exportação.

## Cabeçalhos JMS genéricos:

Cabeçalhos JMS genéricos são Objectos de dados de serviço (SDO, Service Data Objects) que contêm todas as propriedades das propriedades de mensagem JMS genéricas. Estas propriedades podem ser da mensagem de entrada ou podem ser as propriedades que serão aplicadas à mensagem de saída.

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

As seguintes propriedades são definidas estaticamente no methodBinding:

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

A associação JMS genérica também suporta a modificação dinâmica de cabeçalhos e propriedades JMS na mesma forma que as associações JMS e MQ JMS.

Alguns fornecedores de JMS Genérico colocam restrições sobre quais as propriedades que podem ser definidas pela aplicação e em que combinações. Tem de consultar a documentação de produto de terceiros para mais informações. No entanto, uma propriedade adicional pode ser acrescentada ao methodBinding, ignoreInvalidOutboundJMSProperties, que permite propagar quaisquer excepções.

Os cabeçalhos e propriedades de mensagem JMS genéricos são utilizados apenas quando está ligado o comutador de associação SCDL de arquitectura de componente de serviço base. Quando o comutador está ligado, são propagadas informações de contexto. Por predefinição, este comutador está ligado. Para prevenir a propagação de informação de contexto, altere o valor para **false**.

Quando a propagação de contexto é activada, é permitido que a informação de cabeçalho flua para a mensagem ou para o componente destino. Para ligar e desligar a propagação de contexto, especifique **true** ou **false** para o atributo contextPropagationEnabled das associações de importação e exportação. Por exemplo:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextPropagationEnabled="true">
```

A predefinição é **true**.

## Resolução de problemas Generic Associações JMS:

Poderá diagnosticar e corrigir problemas através de Generic associações JMS.

### Excepções de implementação

Em resposta a várias condições de erro, a implementação de importações e exportações JMS Genéricas poderá devolver um ou dois tipos de excepções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter a excepção (JMSException) original.

## Resolução de problemas Genéricos Expiração de mensagens JMS

Uma mensagem de pedido efectuada pelo fornecedor JMS está sujeita a um limite de validade.

*Expiração de pedido* refere-se ao limite da validade de uma mensagem de pedido efectuada pelo fornecedor JMS quando a JMSExpiration quando é alcançada a hora da mensagem de pedido. Conforme acontece com outras associações JMS, a associação JMS Genérica processa a expiração do pedido através da definição do limite de validade, na mensagem da chamada de retorno colocada pela importação, de modo a que seja a mesma do pedido de saída. A notificação do limite de validade da mensagem da chamada de retorno irá indicar que a mensagem de pedido atingiu o limite de validade, e o cliente deverá ser notificado por meios de uma excepção de negócio.

Contudo, caso o destino da chamada de retorno seja movido para o fornecedor da terceira parte, este tipo de expiração de pedido não é suportada.

*Expiração de pedido* refere-se ao limite da validade de uma mensagem de pedido efectuada pelo fornecedor JMS quando a hora do JMSExpiration da mensagem de pedido é alcançada.

A expiração da resposta para a associação JMS genérica não é suportada, devido ao facto de o comportamento de expiração exacto de um terceiro fornecedor JMS não estar definido. Pode, no entanto, verificar se a resposta excedeu o limite da validade caso, ou quando, a receba.

Para mensagens de pedidos externos, o valor Limite de validade JMS será calculado a partir da hora esperada e a partir dos valores Limite da validade do pedido levados a cabo no asyncHeader, IF SET.

### **Resolução de problemas Genéricosno fornecedor de ligação JMS.**

Quando definir determinados tipos de fornecedores de ligação no seu fornecedor JMS Genérico, pode receber uma mensagem de erro quando tentar iniciar a aplicação. É possível modificar o fornecedor de ligação externo para evitar este problema.

Quando iniciar uma aplicação, poderá receber as seguintes mensagens de erro:

Porta de Recepção MDB JMSConnectionFactory não corresponde  
Tipo JMSDestination

Este problema pode surgir quando definir fornecedores de ligação externos. A excepção pode ser fornecida, especificamente, quando criar um Fornecedor de Ligação Tópico JMS 1.0.2, em vez de um Fornecedor de Ligação JMS 1.1 (unificado), (isto é, um que seja capaz de suportar tanto a comunicação ponto a ponto, como a comunicação publicação/subscrição).

Para resolver esta questão, executar os seguintes passos:

1. Aceda ao fornecedor JMS Genéricoque está a utilizar.
2. Substituir o Fornecedor de Ligação Tópico JMS 1.0.2 que definiu por um Fornecedor de Ligação JMS 1.1. (unificado).

Quando iniciar a aplicação com o Fornecedor de Ligação JMS 1.1. que acabou de definir, não deverá voltar a receber uma mensagem de erro.

### **As mensagens JMS baseadas em SCA não aparecem no gestor de eventos falhados.**

Se as mensagens SCA originadas através de interacção JMS genérica falharem, poderá esperar encontrá-las no gestor de eventos falhados. Caso essas mensagens não constem do gestor de eventos falhados, certifique-se de que o valor da propriedade de tentativa máxima na porta de recepção sublinhada é igual ou superior a 1. Definir este valor para 1, ou para um número superior, permite a interacção com o gestor de eventos que falharam, durante chamadas para execução de SCA para as associações JMS genéricas.

## Excepções de processamento:

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal excepção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer excepção da rotina de tratamento de dados é reiniciada ciclicamente numa excepção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer excepção de tempo de execução, incluindo uma excepção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a excepção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma excepção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

## Associações JMS do WebSphere MQ

As associações JMS do WebSphere MQ fornecem integração com aplicações externas que utilizem um fornecedor baseado em JMS do WebSphere MQ.

Utilize as associações de exportação e importação JMS do WebSphere MQ para integrar directamente com sistemas JMS ou MQ JMS externos do ambiente de servidor. Isto elimina a necessidade de utilizar as funcionalidades MQ Link ou Client Link do Barramento de integração de serviço.

Quando um componente interage com um serviço baseado em JMS do WebSphere MQ através de uma importação, a associação de importação JMS do WebSphere MQ utiliza um destino para o qual os dados serão enviados e um destino onde a resposta possa ser recebida. A conversão dos dados para e de uma mensagem JMS é efectuada através da rotina de tratamento de dados JMS ou do componente de associação de dados.

Quando um módulo SCA fornece um serviço a clientes JMS do WebSphere MQ, a associação de exportação JMS do WebSphere MQ utiliza o destino onde o pedido pode ser recebido e a resposta pode ser enviada. A conversão dos dados para e de uma mensagem JMS é efectuada através da rotina de tratamento de dados JMS ou associação de dados.

O selector de função fornece um mapeamento para a operação no componente de destino a ser chamado.

## Descrição geral das associações JMS do WebSphere MQ:

A associação JMS do WebSphere MQ fornece integração com aplicações externas que utilizem o fornecedor JMS do WebSphere MQ.

## Tarefas administrativas do WebSphere MQ

O administrador de sistema do WebSphere MQ deve criar o gestor de filas (Queue Manager) subjacente do WebSphere MQ, o qual as associações JMS do WebSphere MQ utilizarão antes de executar uma aplicação que contenha essas associações.

## Associações de importação JMS do WebSphere MQ

A importação JMS do WebSphere MQ permite que componentes dentro do módulo SCA comuniquem com serviços fornecidos por fornecedores baseados em JMS do WebSphere MQ. É necessário estar a utilizar uma versão suportada do WebSphere MQ. Os requisitos de hardware e software detalhados podem ser encontrados nas páginas de suporte da IBM.

São suportados dois tipos de cenários de utilização para associações de importação JMS do WebSphere MQ, dependendo do tipo de operação chamada:

- Unidireccional: a importação JMS do WebSphere MQ coloca uma mensagem no destino de envio configurado na associação de importação. Nada é enviado para o campo `replyTo` do cabeçalho JMS.
- Bidireccional (pedido-resposta): a importação JMS do WebSphere MQ coloca uma mensagem no destino de envio.

O destino recebe é definido no campo do cabeçalho `replyTo`. Um bean orientado por mensagens (MDB) é implementado para escutar no destino de recepção, e quando uma mensagem é recebida, o MDB transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)** no Integration Designer) para esperar que o identificador de correlação da mensagem de resposta tenha sido copiado do identificador da mensagem de pedido (a predefinição) ou do identificador de correlação da mensagem de pedido.

Para ambos os cenários de utilização unidireccional e bidireccional, as propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser definidas a partir da associação do método de importação JMS. Algumas dessas propriedades têm significados especiais para o tempo de execução JMS da SCA.

É importante notar que o JMS do WebSphere MQ é uma associação assíncrona. Se um componente de chamada invocar uma importação JMS do WebSphere MQ de forma sincronizada (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço JMS.

A Figura 37 na página 133 ilustra a forma como a importação está ligada ao serviço externo.



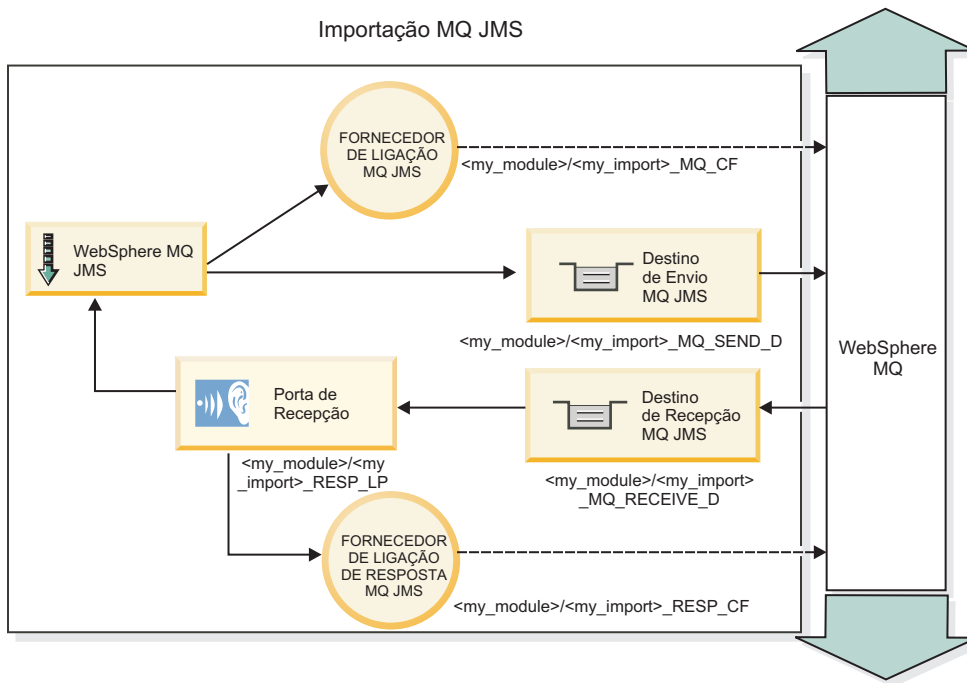


Figura 82. Recursos de associação de importação JMS do WebSphere MQ

### Associações de exportação JMS do WebSphere MQ

A associação de exportação JMS do WebSphere MQ fornece os meios para os módulos SCA fornecerem serviços para aplicações JMS externas no fornecedor JMS baseado no WebSphere MQ.

Um bean orientado por mensagens é implementado para escutar pedidos de entrada no destino recebe especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. O destino especificado no campo replyTo da mensagem de resposta substitui o destino especificado no campo send.

A Figura 38 na página 134 ilustra como o solicitador externo está ligado à exportação.

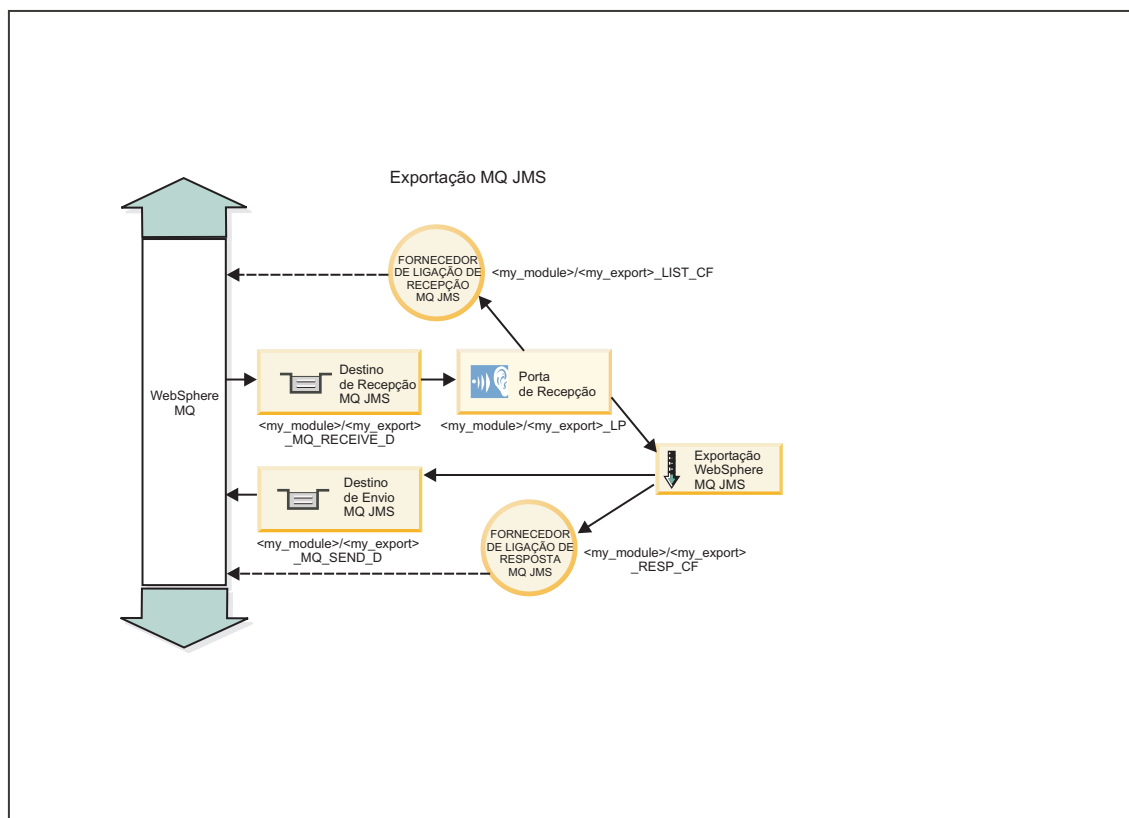


Figura 83. Recursos de associação de exportação JMS do WebSphere MQ

**Nota:** Figura 37 na página 133 e Figura 38 na página 134 ilustram a forma como uma aplicação de uma versão anterior de IBM Business Process Manager está ligado a um serviço externo. Para aplicações desenvolvidas para a versão 7.0 de IBM Business Process Manager, a especificação de activação é utilizada em vez da porta de recepção e do fornecedor de ligação.

### Funções-chave de associações ao WebSphere MQ JMS:

As funções-chave de associações ao WebSphere MQ JMS incluem cabeçalhos, artefactos Java EE e recursos criados Java EE.

#### Cabeçalhos

Um cabeçalho com mensagem JMS contém uma série de campos predefinidos com valores utilizados tanto por clientes como fornecedores para identificar e encaminhar mensagens. É possível propriedades de associação para configurar estes cabeçalhos com valores fixos ou especificar dinamicamente os cabeçalhos no tempo de execução.

#### JMSCorrelationID

Liga-se a uma mensagem relacionada. Normalmente, este campo fica definido para a cadeia do identificador de mensagem da mensagem à qual se está a responder.

#### TargetFunctionName

Este cabeçalho é utilizado por um dos selectores de função fornecidos para identificar a operação que está a ser chamada. Definir adequadamente o cabeçalho TargetFunctionName JMS em mensagens enviadas para uma exportação JMS permite que este selector de função seja utilizado. A propriedade pode ser definida directamente nas aplicações de cliente JMS ou ao ligar uma importação a uma

associação JMS para a referida exportação. Neste caso, a associação de importação JMS deve ser configurada para definir o cabeçalho TargetFunctionName em cada operação da interface para o nome da operação.

## Esquemas de correlação

As associações WebSphere MQ JMS oferecem vários esquemas de correlação que são utilizados para determinar a forma de correlação entre mensagens de pedido e mensagens de resposta.

### RequestMsgIDToCorrelID

O JMSMessageID é copiado para o campo JMSCorrelationID. Esta é a definição predefinida.

### RequestCorrelIDToCorrelID

O JMSCorrelationID é copiado para o campo JMSCorrelationID.

## Recursos Java EE

São criados vários recursos Java EE quando é implementada uma importação MQ JMS num ambiente Java EE.

## Parameters (Parâmetros)

### Fornecedor de ligação MQ

Utilizado por clientes para criar uma ligação ao fornecedor MQ JMS.

### Fornecedor de ligação de respostas

Utilizado pelo tempo de execução SCA MQ JMS quando o destino de envio está num Gestor de fila diferente do destino de recepção.

### Especificação de activação

É associada uma especificação de activação MQ JMS a um ou mais beans orientados por mensagens, que proporciona a configuração necessária para que estes recebam mensagens.

## Destinos

- Destino de envio:
  - Importações: para onde será enviado o pedido ou mensagem de saída.
  - Exportações: para onde será enviada a mensagem de resposta, caso o campo de cabeçalho JMSReplyTo da mensagem de entrada não se lhe sobreponha.
- Destino de recepção:
  - Importações: onde deverá ficar colocada a resposta ou mensagem de entrada.
  - Exportações: onde deverá ficar colocada a mensagem de entrada ou de pedido.

## Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos-o cabeçalho do sistema JMS e várias propriedades JMS. Ambos os tipos de cabeçalhos podem ser acedidos num módulo de mediação no Objecto de mensagens de serviço (SMO, Service Message Object) ou através da API ContextService.

## Cabeçalho de sistema JMS

O cabeçalho de sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos encontrados tipicamente num cabeçalho JMS. Apesar de estes poderem ser modificados na mediação (ou ContextService), alguns campos de cabeçalho de sistema JMS definidos no SMO não serão propagados na mensagem JMS de saída, uma vez que são substituídos por valores de sistema ou estáticos.

Os campos de chave no cabeçalho de sistema JMS que podem ser actualizados numa mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidos específicos
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não-persistente, a predefinição é persistente)
- **JMSPriority** - valor de prioridade (0 a 9, a predefinição é JMS\_Default\_Priority)

### Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista Propriedades. As propriedades podem ser adicionadas, actualizadas ou eliminadas numa mediação ou através da utilização da API ContextService.

As propriedades também podem ser definidas estaticamente na associação JMS. As propriedades que estiverem definidas estaticamente substituem definições (com o mesmo nome) que estejam definidas dinamicamente.

As propriedades de utilizador propagadas de outras associações (por exemplo, uma associação HTTP) serão produzidas na associação JMS como propriedades JMS.

### Definições de propagação de cabeçalho

A propagação do cabeçalho de sistema JMS e propriedades quer da mensagem JMS interna para componentes na direcção do fluxo ou de componentes na direcção oposta ao fluxo para a mensagem JMS de saída podem ser controlados pelo sinalizador Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) na associação.

Quando o Cabeçalho de Protocolo de Propagação (Propagate Protocol Header) está definido, a informação de cabeçalho pode fluir para a mensagem ou para o componente de destino, tal como descrito na lista seguinte:

- Pedido de exportação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.
- Resposta de exportação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de exportação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de exportação JMS.
- Pedido de importação JMS  
Quaisquer campos de cabeçalho JMS definidos no serviço de contexto serão utilizados na mensagem de saída, se não forem substituídos pelas propriedade estáticas definidas na associação de importação JMS. Quaisquer propriedades definidas no serviço de contexto serão utilizadas na mensagem de saída, se não forem substituídas por propriedades estáticas definidas na associação de importação JMS.
- Resposta de importação JMS  
O cabeçalho JMS recebido na mensagem será propagado para componentes de destino através do serviço de contexto. As propriedade JMS recebidas na mensagem serão propagadas para componentes de destino através do serviço de contexto.

### Clientes externos:

O servidor pode enviar mensagens para ou receber mensagens de clientes externos utilizando associações JMS WebSphere MQ.

Um cliente externo (como um portal Web ou um sistema de informação empresarial) pode enviar uma mensagem para um componente SCA na aplicação através de uma exportação ou pode ser chamado por um componente SCA na aplicação através de uma importação.

A associação de exportação JMS WebSphere MQ fornece beans orientados por mensagens (MDBs) para escutar por pedidos de entrada no destino recebe especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações através da associação de exportação.

As importações JMS WebSphere MQ são associadas a, e podem entregar mensagens, a clientes externos. Esta mensagem pode ou não exigir uma resposta do cliente externo.

Pode encontrar mais informações sobre como interagir com clientes externos utilizando o WebSphere MQ no Information Center do WebSphere MQ.

### **Resolução de problemas WebSphere MQ Associações JMS:**

Poderá diagnosticar e corrigir problemas através de WebSphere MQ associações JMS.

### **Exceções de implementação**

Em resposta a várias condições de erro, a implementação de importações e exportações MQ JMS poderá devolver um ou dois tipos de exceções:

- Excepção empresarial do serviço: esta excepção é devolvida caso ocorra a falha especificada na interface do negócio de serviço (tipo de porta WSDL).
- Excepção do tempo de execução do serviço: criado em todos os outros casos. Na maioria dos casos, a excepção do caso irá conter a excepção original (JMSEException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

### **As mensagens MQ WebSphere baseadas em SCA não estão a surgir no gestor de eventos falhados.**

Se as mensagens SCA que originadas através de uma WebSphere interacção JMS falharem, poderá encontrá-las no gestor de eventos falhados. Se essas mensagens não aparecerem no gestor de eventos falhados, certifique-se de que o valor da propriedade de tentativa máxima na porta de recepção sublinhada é igual, ou superior ao **1**. Definir esta valor para **1** ou mais permite a interacção com o gestor de eventos que falhou, durante chamadas para execução da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) para as associações MQ JMS.

### **Cenários de uso indevido: comparação com WebSphere associações MQ**

A WebSphere associação MQ JMS é concebida para cooperar com as aplicações JMS implementadas contra WebSphere MQ, o que expõe mensagens de acordo com o modelo de mensagens JMS. Contudo, as WebSphere importações e exportações MQ são concebidas, sobretudo, para cooperar com WebSphere aplicações MQ nativas, e para expor o conteúdo completo do WebSphere corpo de mensagens MQ TO MEDIATIONS.

Os cenários seguintes devem ser construídos utilizando a WebSphere associação MQ JMS, e não a associação WebSphere MQ:

- Chamar um bean orientado por mensagens (MDB) JMS de um módulo SCA, onde o bean orientado por mensagens é implementado contra o WebSphere fornecedor MQ JMS. Utilizar uma WebSphere importação MQ JMS.
- Permitir que o módulo da SCA seja chamado por um componente servlet Java EE ou EJB por meio de JMS. Utilizar uma WebSphere exportação MQ JMS.
- Mediar os conteúdos de um MapMessage JMS, em trânsito por WebSphere MQ. Utilizar WebSphere exportações e importações MQ JMS conjuntamente com a associação de dados apropriada tratamento ou dados.

Em algumas situações é provável que a associação WebSphere MQ e a WebSphere associação MQ JMS cooperem. Em específico, quando se abre caminho entre EE Java e EE não-JAVA WebSphere as aplicações MQ utilizam WebSphere exportações MQ e WebSphere importações MQ JMS (ou vice-versa) conjuntamente com associações de dados ou módulos de mediação apropriados (ou ambos).

### Exceções de processamento:

A forma em que a associação é configurada determina a forma como as exceções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal exceção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal exceção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer exceção da rotina de tratamento de dados é reiniciada ciclicamente numa exceção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a exceção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma exceção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

### Associações do WebSphere MQ

A associação do WebSphere MQ fornece uma conectividade Service Component Architecture (SCA) com aplicações do WebSphere MQ.

Utilize as associações de exportação e importação do WebSphere MQ para serem integradas directamente um sistema baseado no WebSphere MQ a partir do ambiente de servidor. Isto elimina a necessidade de utilizar as funcionalidades MQ Link ou Client Link do Barramento de integração de serviço.

Quando um componente interage com um serviço do WebSphere MQ através de uma importação, a associação de importação do WebSphere MQ utiliza uma fila para a qual os dados são enviados e uma fila onde a resposta pode ser recebida.

Quando um módulo SCA fornece um serviço a clientes do WebSphere MQ, a associação de exportação do WebSphere MQ utiliza uma fila onde o pedido pode ser recebido e a resposta pode ser enviada. O selector de função fornece um mapeamento para a operação no componente de destino a ser chamado.

A conversão de dados de carga útil para e de uma mensagem MQ é efectuada através da rotina de tratamento de dados ou associação de dados do corpo de MQ. A conversão dos dados de cabeçalho para e de uma mensagem MQ é efectuada através da associação de dados de cabeçalho MQ.

Para obter mais informações sobre as versões do WebSphere MQ suportadas, consulte a página da web requisitos detalhados do sistema.

### Descrição geral das associações do WebSphere MQ:

A associação do WebSphere MQ fornece integração com aplicações baseadas em MQ nativas.

### Tarefas administrativas do WebSphere MQ

O administrador de sistema do WebSphere MQ deve criar o gestor de filas subjacente do WebSphere MQ, que as associações do WebSphere MQ utilizarão antes de executarem um aplicação que contém essas associações.

### Tarefas administrativas do WebSphere

Tem de definir a propriedade do **Native library path (Caminho da biblioteca nativa)** do adaptador de recurso MQ no Websphere para a versão do WebSphere MQ suportada pelo servidor e reiniciar o servidor. Isto garante que as bibliotecas de uma versão suportada do WebSphere MQ estão a ser utilizadas. Os requisitos detalhados do hardware e do software podem ser encontrados nas páginas de suporte da IBM.

### Associações de importação do WebSphere MQ

A associação de importação do WebSphere MQ permite aos componentes dentro do módulo SCA para comunicar com serviços fornecidos por aplicações baseadas no WebSphere MQ. É necessário estar a utilizar uma versão suportada do WebSphere MQ. Os requisitos detalhados do hardware e do software podem ser encontrados nas páginas de suporte da IBM.

A interacção com sistemas do WebSphere MQ inclui a utilização de filas para enviar pedidos e receber respostas.

São suportados dois tipos de cenários de utilização para a associação de importação do WebSphere MQ, dependendo do tipo de operação a ser chamada:

- Unidireccional: a importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Send destination queue (Fila de Destino de Envio)** da associação de importação. Nada é enviado para o campo `replyTo` do cabeçalho MQMD.
- Bidireccional (pedido-resposta): a importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Send destination queue (Fila de Destino de Envio)**

A fila receive é definida no campo do cabeçalho `replyTo` MQMD. Um bean orientado por mensagens (MDB) é implementado para escutar na fila de recepção, e quando uma mensagem é recebida, o bean orientado por mensagens transmite a resposta de volta para o componente.

A associação de importação pode ser configurada (utilizando o campo **Response correlation scheme (Esquema de Correlação de Resposta)**) para esperar o identificador de correlação de mensagem de



resposta tenha sido copiado pelo identificador da mensagem de pedido (a predefinição) ou do identificador de correlação de mensagem de pedido.

É importante notar que o WebSphere MQ é uma associação assíncrona. Se um componente de chamada invocar uma importação de forma sincronizada do WebSphere MQ (para uma operação bidireccional), o componente de chamada é bloqueado até que a resposta seja devolvida pelo serviço WebSphere MQ.

A Figura 39 na página 140 ilustra como a importação está ligada ao serviço externo.

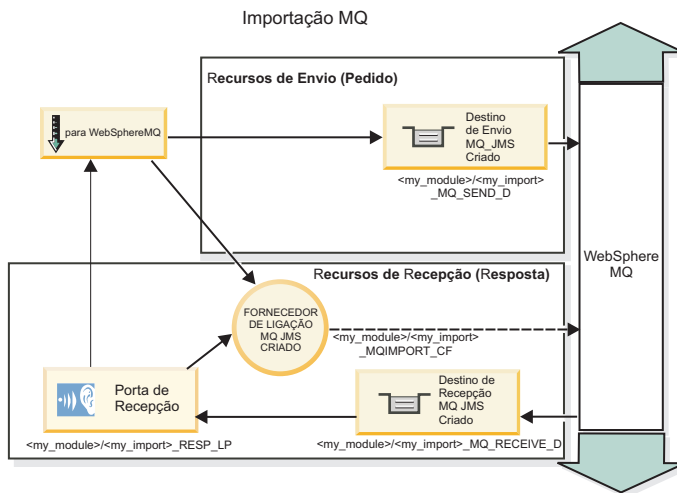


Figura 84. Recursos de associação de importação do WebSphere MQ

### Associações de exportação do WebSphere MQ

A associação de exportação do WebSphere MQ fornece os meios para os módulos SCA fornecerem serviços a aplicações baseadas no WebSphere MQ externas.

Um bean orientado por mensagens é implementado para escutar por pedidos de entrada no campo **Receive destination queue (Fila de Recepção de Destino)** especificado na associação de exportação. A fila especificada no campo **Send destination queue (Fila de Destino de Envio)** é utilizada para enviar a resposta para o pedido interno se o componente chamado fornecer uma resposta. A fila especificada no campo replyTo da mensagem de resposta substitui a fila especificada no campo **Send destination queue (Fila de Destino de Envio)**.

A Figura 40 na página 141 ilustra como o solicitador externo está ligado à exportação.

## Exportação MQ

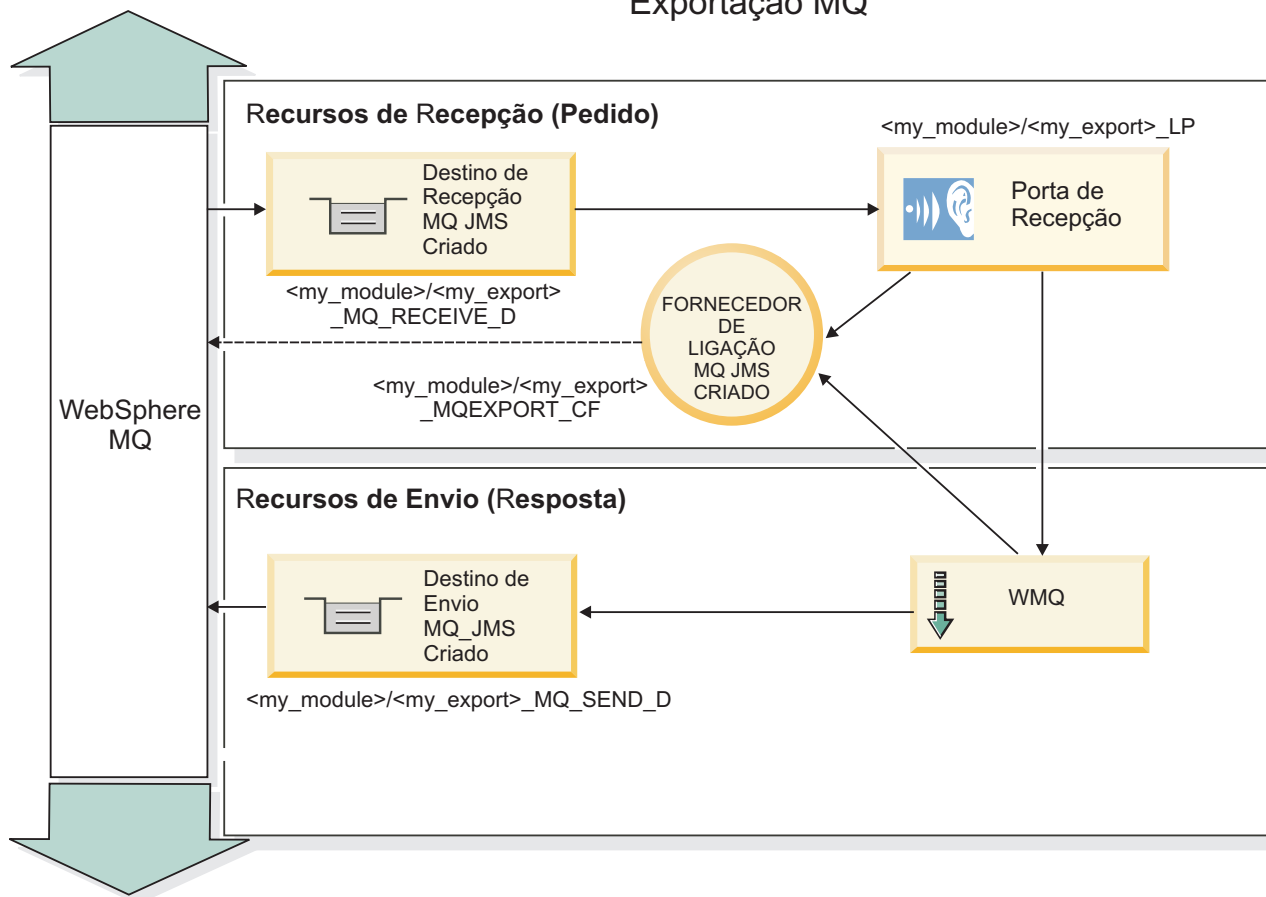


Figura 85. Recursos de associação de exportação do WebSphere MQ

**Nota:** A Figura 39 na página 140 e a Figura 40 na página 141 ilustram a forma como uma aplicação de uma versão anterior do IBM Business Process Manager está ligada a um serviço externo. Para aplicações desenvolvidas para a versão 7.x, ou superior, do IBM Business Process Manager, a especificação de activação é utilizada em vez da porta de recepção e do fornecedor de ligação.

### Funções-chave de uma associação WebSphere MQ:

As funções-chave de uma associação WebSphere MQ incluem cabeçalhos, artefactos Java EE e recursos criados Java EE.

### Esquemas de correlação

A aplicação de pedido/resposta WebSphere MQ pode utilizar uma de várias técnicas para correlacionar mensagens de resposta com pedidos, criadas com base nos campos MQMD MessageID e CorrelID. Na grande maioria dos casos, o solicitador permite que o gestor de filas selecione um MessageID e espera que a aplicação de resposta o copie para o CorrelID da resposta. Na maioria dos casos, o solicitador e aplicação de resposta sabem implicitamente qual a técnica de correlação que está a ser utilizada. Por vezes, a aplicação de resposta irá respeitar vários sinalizadores no campo Relatório do pedido que descrevem como processar estes campos.

As associações de exportação para mensagens WebSphere MQ podem ser configuradas com as seguintes opções:

### Response MsgId options (Opções de resposta MsgId):

**Novo MsgID**

Permite que o gestor de filas selecione um MsgId único para a resposta (por predefinição).

**Copiar a partir do MsgID do Pedido**

Copia o campo MsgId a partir do campo MsgId do pedido.

**Copiar a partir da mensagem SCA**

Define o MsgId para que seja incluído nos cabeçalhos WebSphere MQ na mensagem de resposta da SCA, ou permite que o gestor de filas defina um novo Id, caso o valor não exista.

**Como opções de relatório**

Inspecciona o campo Relatório do MQMD no pedido de sugestão sobre o procedimento a adoptar no processamento do MsgId. As opções MQRO\_NEW\_MSG\_ID e MQRO\_PASS\_MSG\_ID são suportadas e comportam-se como Novo MsgId e Copiar a partir do MsgID do pedido.

**Response CorrelId options (Opções de resposta CorrelId):****Copiar a partir do MsgID do Pedido**

Copia o campo CorrelId a partir do campo MsgId do pedido (por predefinição).

**Copiar a partir do CorrelID do Pedido**

Copia o campo CorrelId a partir do campo CorrelId do pedido.

**Copiar a partir da mensagem SCA**

Define o CorrelId para ser incluído nos cabeçalhos WebSphere MQ na mensagem de resposta da SCA, ou deixa-o em branco, caso o valor não exista.

**Como opções de relatório**

Inspecciona o campo Relatório do MQMD no pedido de sugestão sobre o procedimento a adoptar no processamento do CorrelId. As opções MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID e MQRO\_PASS\_CORREL\_ID são suportadas e comportam-se como Copiar a partir do MsgID do pedido e Copiar a partir do CorrelID do pedido.

As associações de importação para mensagens WebSphere MQ podem ser configuradas com as seguintes opções:

**Request MsgId options (Opções do MsgId do pedido):****Novo MsgID**

Permite que o gestor de filas selecione um MsgId único para o pedido (por predefinição).

**Copiar a partir da mensagem SCA**

Define o MsgId para ser incluído nos cabeçalhos WebSphere MQ na mensagem de pedido da SCA, ou permite que o gestor de filas defina um novo Id, caso o valor não exista.

**Response correlation options (Opções de correlação de resposta):****A resposta copiou o CorrelID a partir do MsgId**

Espera-se que a mensagem de resposta tenha um campo CorrelId definido, de acordo com o MsgId do pedido (por predefinição).

**A resposta copiou o MsgID a partir do MsgId**

Espera-se que a mensagem de resposta tenha um campo MsgId definido, de acordo com o MsgId do pedido.

**A resposta copiou o CorrelID a partir do CorrelId**

Espera-se que a mensagem de resposta tenha um campo CorrelId definido, de acordo com o CorrelId do pedido.

## Recursos Java EE

São criados vários recursos Java EE quando é implementada uma associação WebSphere MQ num ambiente Java EE.

### Parameters (Parâmetros)

#### Fornecedor de ligação MQ

Utilizado por clientes para criar uma ligação ao fornecedor WebSphere MQ.

#### Fornecedor de ligação de respostas

Utilizado pelo tempo de execução SCA MQ quando o destino de envio está num Gestor de fila diferente do destino de recepção.

#### Especificação de activação

É associada uma especificação de activação MQ JMS a um ou mais beans orientados por mensagens, que proporciona a configuração necessária para que estes recebam mensagens.

#### Destinos

- Destino de envio: é para onde é enviado o pedido ou mensagem de saída (importação); é para onde será enviada a mensagem de resposta (exportação), caso o campo de cabeçalho MQMD ReplyTo não se lhe sobreponha na mensagem de entrada.
- Destino de recepção: é onde deverá ficar colocada a resposta/pedido ou a mensagem de entrada.

#### Cabeçalhos WebSphere MQ:

Os cabeçalhos WebSphere MQ incorporam certas convenções para conversão de mensagens de Arquitectura do Componente de Serviço SCA).

As mensagens WebSphere MQ consistem num cabeçalho do sistema (o MQMD), zero ou outros mais cabeçalhos MQ (sistema ou personalizados) e num corpo de mensagem. Se existirem múltiplos cabeçalhos de mensagem na mensagem, a ordem dos cabeçalhos é significativa.

Cada cabeçalho contém informação que descreve a estrutura do seguinte cabeçalho. MQMD descreve o primeiro cabeçalho.

#### Como são analisados os cabeçalhos MQ

É utilizada uma associação de dados do cabeçalho MQ para analisar os cabeçalhos MQ. Os seguintes cabeçalhos são automaticamente suportados:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Os cabeçalhos que começam com **MQH** são processados de modo diferente. Campos específicos do cabeçalho não são analisados; permanecem como bytes não analisados.

Para outros cabeçalhos MQ, pode inserir associações de dados de cabeçalho MQ personalizadas para analisar esses cabeçalhos.

#### Como os cabeçalhos MQ são acedidos

Os cabeçalhos MQ podem ser acedidos no produto de uma de duas maneiras:

- Através do objecto de mensagem de serviço (SMO) numa mediação
- Através de ContextService API

Os cabeçalhos MQ são representados internamente com o elemento SMO MQHeader. MQHeader é um contentor de dados do cabeçalho que expande MQControl mas contém um elemento de valor de anyType. Contém MQMD, MQControl (informação de controlo do corpo da mensagem MQ) e uma lista de outros cabeçalhos MQ.

- MQMD representa os conteúdos da descrição da mensagem WebSphere MQ, excepto para informação que determine a estrutura e codificação do corpo.
- MQControl contém informação que determina a estrutura e codificação de um corpo de mensagem.
- MQHeaders contém uma lista de objectos de MQHeader.

A cadeia do cabeçalho MQ está desenrolada para que, dentro do SMO, cada cabeçalho MQ transporte a sua própria informação de controlo (CCSID, Codificação e Formato). Os cabeçalhos podem ser facilmente adicionados ou eliminados, sem alterar outros dados do cabeçalho.

### Campos de definição no MQMD

É possível actualizar o MQMD utilizando o Contexto API ou através do objecto de mensagem de serviço (SMO) numa mediação. Os seguintes campos são automaticamente propagados para a mensagem MQ de saída:

- Codificação
- CodedCharacterSet
- Formato
- Relatório
- Expiração
- Comentários
- Prioridade
- Persistência
- CorrelId
- MsgFlags

Configure a associação MQ numa Importação ou Exportação para propagar as seguintes propriedades para a mensagem MQ de saída:

#### MsgID

Defina **Request Message ID (Pedir ID de Mensagem)** para copy from SCA message (copiar de mensagem SCA).

#### MsgType

Limpe a caixa de verificação de **Set message type to MQMT\_DATAGRAM or MQMT\_REQUEST for request-response operation (Definir tipo de mensagem para MQMT\_DATAGRAM ou MQMT\_REQUEST para operação de pedido-resposta)**.

#### ReplyToQ

Limpe a caixa de verificação **Override reply to queue of request message (Substituir resposta para fila de mensagem de pedido)**.

#### ReplyToQMgr

Limpe a caixa de verificação **Override reply to queue of request message (Substituir resposta para fila de mensagem de pedido)**.

A partir da versão 7.0, os campos de contexto podem ser substituídos utilizando uma propriedade personalizada na definição de destino de JNDI. Defina a propriedade personalizada MDCTX com o valor SET\_IDENTITY\_CONTEXT no destino de envio para propagar os seguintes campo para a mensagem MQ de saída:

- UserIdentifier

- AppIdentityData

Defina a propriedade personalizada MDCTX com o valor SET\_ALL\_CONTEXT no destino de envio para propagar as seguintes propriedades para a mensagem MQ de saída:

- UserIdentifier
- AppIdentityData
- PutApplType
- PutApplName
- ApplOriginData

Alguns campos não são propagados para a mensagem MQ de saída. Os seguintes campos são substituídos durante o envio da mensagem:

- BackoutCount
- AccountingToken
- PutDate
- PutTime
- Deslocamento
- OriginalLength

#### **Acrescentar estatisticamente MQCIH numa associação WebSphere MQ:**

O IBM Business Process Manager suporta a adição estatística de informação de cabeçalho MQCIH sem utilizar um módulo de mediação.

Existem várias formas de adicionar informação de cabeçalho MQCIH a uma mensagem (por exemplo, ao utilizar a primitiva de mediação Definidor de Cabeçalho). Pode ser útil adicionar estatisticamente esta informação de cabeçalho, sem a utilização de um módulo de mediação adicional. A informação estatística de cabeçalho, incluindo o nome de programa, identificador de transacção e outros detalhes de cabeçalho do CICS podem ser definidos e adicionados como parte da associação WebSphere MQ.

O WebSphere MQ, o MQ CICS Bridge e o CICS devem ser configurados para informação de cabeçalho MQCIH, para serem adicionados estatisticamente.

Pode utilizar o Integration Designer para configurar a importação WebSphere MQ com os valores estáticos que são necessários para a informação de cabeçalho MQCIH.

Quando chega uma mensagem e é processada pela importação WebSphere MQ, é realizada uma verificação para ver se a informação de cabeçalho MQCIH já está presente na mensagem. Se a MQCIH estiver presente, os valores estáticos definidos na importação WebSphere MQ são utilizados para substituir os correspondentes valores dinâmicos na mensagem. Se o MQCIH não estiver presente, é criado um na mensagem e os valores estáticos definidos na importação WebSphere MQ são acrescentados.

Os valores estáticos definidos na importação WebSphere MQ são específicos a um método. Pode especificar diferentes valores estáticos MQCIH para diferentes métodos dentro da mesma importação WebSphere MQ.

Este serviço não é utilizado para fornecer valores padrão se o MQCIH não contiver informação de cabeçalho específica porque um valor estático definido na importação WebSphere MQ vai substituir um correspondente valor fornecido na mensagem recebida.

## **Cientes externos:**

O IBM Business Process Manager pode enviar mensagens para ou receber mensagens de clientes externos utilizando associações WebSphere MQ.

Um cliente externo (por exemplo, um portal Web ou um sistema de informação empresarial) pode enviar uma mensagem para um componente SCA na aplicação através de uma exportação ou pode ser chamado por um componente SCA na aplicação através de uma importação.

A associação de exportação do WebSphere MQ fornece beans orientados por mensagens (MDBs) para escutar por pedidos de entrada no destino receive especificado na associação de exportação. O destino especificado no campo send é utilizado para enviar a resposta ao pedido interno se a aplicação chamada fornecer uma resposta. Assim, um cliente externo consegue chamar aplicações através da associação de exportação.

As importações do WebSphere MQ são associadas a, e podem entregar mensagens, a clientes externos. Esta mensagem pode ou não exigir uma resposta do cliente externo.

Pode encontrar mais informações sobre como interagir com clientes externos utilizando o WebSphere MQ no Information Center do WebSphere MQ.

## **Resolução de problemas das associações WebSphere MQ:**

Poderá diagnosticar e corrigir falhas e condições de erro que ocorram com as WebSphere associações MQ.

### **Principais condições de erro**

As principais condições de erro WebSphere das associações MQ são determinadas pela semântica transaccional, pela WebSphere configuração MQ, ou por referência a um comportamento existente noutros componentes. As principais condições de erro incluem:

- Erro ao ligar-se ao WebSphere gestor de filas MQ ou à fila MQ.  
Erro ao ligar-se ao WebSphere MQ para receber mensagens irá resultar num erro inicial, ao nível da porta de recepção do bean orientado por mensagens. Esta condição será registada no registo do WebSphere Application Server. Mensagens que persistam irão permanecer na WebSphere fila MQ até serem obtidas com sucesso (ou até expirarem por WebSphere MQ).  
Um erro ao ligar-se ao WebSphere MQ para enviar mensagens de saída, irá causar a remoção de alterações da transacção que controla o envio.
- Um erro ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.  
Um erro na associação de dados causa uma remoção de alterações do controlo de transacção do funcionamento.
- Um erro no envio da mensagem de saída.  
Um erro no envio de uma mensagem causa uma remoção de alterações da transacção relevante.
- Mensagens de resposta múltiplas ou inesperadas.  
A importação espera apenas uma mensagem de resposta para cada mensagem de pedido. Se receber mais do que uma resposta, ou se receber uma resposta tardiamente (uma resposta em que o limite da validade da SCA tenha expirado), é lançada a Excepção do tempo de execução do serviço. São removidas as alterações da transacção, e a mensagem de resposta é novamente retirada da fila, e guiada pelo gestor de eventos que falhou.

## **Cenários de uso indevido: comparação com WebSphere associações MQ JMS**

As WebSphere importações e exportações MQ são concebidas, sobretudo, para cooperar com WebSphere aplicações MQ e expor o conteúdo completo do WebSphere corpo de mensagens MQ para as mediações.



A WebSphere associação MQ JMS é concebida para cooperar com as aplicações JMS implementadas contra WebSphere MQ, o que expõe mensagens de acordo com o modelo de mensagens JMS.

Os cenários seguintes devem ser construídos utilizando a WebSphere associação MQ JMS, e não a associação WebSphere MQ:

- Chamar um bean orientado por mensagens (MDB) JMS de um módulo SCA, onde o bean orientado por mensagens é implementado contra o WebSphere fornecedor MQ JMS. Utilizar uma WebSphere importação MQ JMS.
- Permitir que o módulo da SCA seja chamado por um componente servlet Java EE ou EJB por meio de JMS. Utilizar uma WebSphere exportação MQ JMS.
- Mediar os conteúdos de um MapMessage JMS, em trânsito por WebSphere MQ. Utiliza WebSphere exportações e importações MQ JMS conjuntamente com a associação de dados apropriada.

Em algumas situações é provável que a associação WebSphere MQ e a WebSphere associação MQ JMS cooperem. Em específico, quando se abre caminho entre EE Java e EE não-JAVA WebSphere as aplicações MQ utilizam WebSphere exportações MQ e WebSphere importações MQ JMS (ou vice-versa) conjuntamente com associações de dados ou módulos de mediação apropriados (ou ambos).

### **Mensagens não entregues**

Caso WebSphere o MQ não consiga entregar uma mensagem a um destino pretendido (devido a erros de configuração, por exemplo), em vez disso, este envia a mensagem para uma fila de mensagens retidas.

Ao fazê-lo, adiciona um cabeçalho de mensagens retidas para o início do corpo da mensagem. Este cabeçalho contém os motivos do erro, o destino original e outras informações.

### **As mensagens MQ baseadas em SCA não aparecem no gestor de eventos falhados.**

Se as mensagens SCA originadas através de uma WebSphere falharem, poderá encontrá-las no gestor de eventos falhados. Se essas mensagens não aparecerem no gestor de eventos falhados, certifique-se de que o destino WebSphere MQ que está por baixo apresenta um valor máximo de entregas falhadas superior a 1. Definir este valor para 2, ou superior, permite a interação com o gestor de eventos falhados durante as chamadas para execução da Arquitectura do Componente de Serviço (SCA, Service Component Architecture) para as associações WebSphere MQ.

### **Os eventos MQ que falharam são reproduzidos para o gestor de filas errado.**

Quando um fornecedor de ligação predefinido está prestes a ser utilizado para ligações de envio, as propriedades de ligação devem corresponder às propriedades definidas, na especificação de activação utilizada para ligações de recepção.

O fornecedor de ligação predefinido é utilizado para criar uma ligação aquando da reprodução de um evento falhado e deve, por isso, ser configurado para o mesmo gestor de filas em que a mensagem foi inicialmente recebida.

### **Excepções de processamento:**

A forma em que a associação é configurada determina a forma como as excepções que são criadas por rotinas de tratamento de dados ou associações de dados são processadas. Além disso, a natureza do fluxo de mediação dita o comportamento do sistema quando tal excepção é lançada.

Pode ocorrer uma variedade de problemas quando uma rotina de tratamento de dados ou associação de dados é chamada pela sua associação. Por exemplo, uma rotina de tratamento de dados recebe uma mensagem que tem uma carga útil corrupta ou pode tentar ler uma mensagem que tem um formato incorrecto.

A forma como a sua associação processa tal exceção é determinada pela forma como implementa a rotina de tratamento de dados ou associação de dados. O comportamento recomendado é que conceba a sua associação de dados para lançar uma **DataBindingException**.

A situação é semelhante para uma rotina de tratamento de dados. Uma vez que a rotina de tratamento de dados é chamada pela associação de dados, qualquer exceção da rotina de tratamento de dados é reiniciada ciclicamente numa exceção de associação de dados. Por isso, uma **DataHandlerException** é reportada como uma **DataBindingException**.

Quando é lançada qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**:

- Se o fluxo de mediação for configurado para ser transaccional, a mensagem JMS é guardada no Gestor de eventos falhados por predefinição por repetição manual ou eliminação.

**Nota:** Pode alterar o modo de recuperação na associação para que a mensagem remova alterações em vez de ser guardada no gestor de eventos falhados.

- Se o fluxo de mediação não for transaccional, a exceção é registada e a mensagem perde-se.

A situação é semelhante para uma rotina de tratamento de dados. Como a rotina de tratamento de dados é chamada pela associação de dados, uma rotina de tratamento de dados é produzida dentro de uma exceção de associação de dados. Por isso, uma **DataHandlerException** é registada como uma **DataBindingException**.

## Limitações de associações

As associações têm algumas limitações na sua utilização que estão listadas aqui.

### Limitações da associação MQ:

A associação MQ tem algumas limitações na sua utilização, que são listadas aqui.

### Nenhuma distribuição publicação-subscrição

O método publicação-subscrição de distribuir mensagens não é suportado actualmente pela associação MQ, apesar do próprio WMQ suportar publicação-subscrição. No entanto, a associação JMS MQ suporta este método de distribuição.

### Filas de recebimento partilhadas

Várias associações de exportação e importação do WebSphere MQ esperam que quaisquer mensagens presentes nas suas filas de recebimento configuradas são para essa exportação ou importação. As associações de importação e exportação devem ser configuradas com as seguintes considerações:

- Cada importação MQ tem de ter uma fila de recebimento diferente, porque a associação de importação MQ assume que todas as mensagens na fila de recebimento são respostas a pedidos que enviou. Se a fila de recebimento for partilhada por mais de uma importação, as respostas podem ser recebidas pela importação incorrecta e não será correlacionada com a mensagem de pedido original.
- Cada exportação MQ deve ter uma fila de recebimento diferente, porque de outro modo não é possível prever qual a exportação que receberá qualquer mensagem de pedido em particular.
- As importações e exportações MQ podem indicar a mesma fila de envio.

### Limitações das associações JMS, MQ JMS e JMS genérico:

As associações JMS e MQ JMS têm algumas limitações.

## Implicações de gerar associações predefinidas

As limitações de utilização das associações JMS, MQ JMS e JMS genérico são debatidas nas secções seguintes:

- Implicações de gerar associações predefinidas
- Esquema de correlação de resposta
- Suporte bidireccional

Quando gera uma associação, vários campos serão preenchidos como predefinições, se não escolher introduzir os valores. Por exemplo, um nome de fornecedor de ligação será criado. Se souber que irá colocar a aplicação num servidor e aceder-lhe remotamente com um cliente, deverá introduzir na criação da associação os nomes JNDI, em vez de aceitar a predefinição, uma vez que provavelmente quererá controlar esses valores através da consola administrativa no tempo de execução.

Porém, se tiver aceite as predefinições e descobrir mais tarde que não consegue aceder a partir de um cliente remoto, pode utilizar a consola administrativa para definir explicitamente o valor do fornecedor de ligação. Localize os pontos de terminação nas definições do fornecedor de ligação e adicione um valor como <server\_hostname>:7276 (se estiver a utilizar o número da porta predefinido).

## Esquema de correlação de resposta

Se utilizar o esquema de correlação de resposta CorrelationId To CorrelationId, utilizado para correlacionar mensagens numa operação de pedido-resposta, tem de ter um identificador de correlação dinâmico na mensagem.

Para criar um ID de correlação dinâmico num módulo de mediação ao utilizar o editor de fluxo de mediação, adicione uma primitiva de mediação de mapeamento antes da importação com a associação JMS. Abra o editor de mapeamento. Os cabeçalhos de arquitectura de componente de serviço estarão disponíveis na mensagem de destino. Arraste um campo que contenha um ID único na mensagem de origem para o ID de correlação no cabeçalho JMS na mensagem de destino.

## Suporte bidireccional

Apenas são suportados caracteres ASCII para nomes Java Naming and Directory Interface (JNDI) no tempo de execução.

## Filas de recebimento partilhadas

Várias associações de exportação e importação esperam que quaisquer mensagens presentes nas suas filas de recebimento configuradas são para essa exportação ou importação. As associações de importação e exportação devem ser configuradas com as seguintes considerações:

- Cada associação de importação tem de ter uma fila de recebimento diferente, porque a associação de importação assume que todas as mensagens na fila de recebimento são respostas a pedidos que enviou. Se a fila de recebimento for partilhada por mais de uma importação, as respostas podem ser recebidas pela importação incorrecta e não será correlacionada com a mensagem de pedido original.
- Cada exportação deve ter uma fila de recebimento diferente, porque de outro modo não é possível prever qual a exportação que receberá qualquer mensagem de pedido em particular.
- As importações e exportações podem indicar a mesma fila de envio.

---

## Objectos de negócio

A indústria do software de computadores desenvolveu vários modelos e contextos de programação em que os *objectos de negócio* fornecem uma representação natural dos dados de negócio para a Process Application.

Em geral, estes objectos de negócio:

- São definidos utilizando padrões da indústria
- Mapeiam de forma transparente dados para as tabelas de bases de dados ou sistemas de informações empresariais
- Suportam protocolos de chamada para execução remota
- Fornecem a fundação do modelo de programação de dados para a programação de aplicações

Process Designer e Integration Designer fornece aos programadores um modelo de objecto de negócio comum para representar diferentes tipos de entidades de negócio de diferentes domínios. Em termos de desenvolvimento, este modelo permite que os programadores definam objectos de negócio como definições de esquema de XML.

Em tempo de execução, os dados de negócio definidos pelas definições de esquema de XML são representados como Objectos de Negócio Java. Neste modelo, os objectos de negócio baseiam-se em rascunhos antecipados da especificação SDO (Service Data Object) e fornecem o conjunto completo de interfaces de aplicação de modelo de programação requeridas para manipular os dados de negócio.

## Definir objectos de negócio

É possível definir objectos de negócio utilizando o editor de objectos de negócio no Integration Designer. O editor de objectos de negócio armazena os objectos de negócio como definições de esquema de XML.

Utilizar o esquema de XML para definir objectos de negócio tem várias vantagens:

- O esquema de XML fornece um modelo de definição de dados baseado em padrões e uma fundação para a interoperabilidade entre sistemas e aplicações heterogéneos diferentes. O esquema de XML é utilizado em conjunto com a Web Services Description Language (WSDL) para fornecer contratos de interface baseados em padrões entre componentes, aplicações e sistemas.
- O esquema de XML define um modelo de definição de dados ricos para representar dados de negócio. Este modelo inclui tipos complexos, tipos simples, tipos definidos pelo utilizador, herança de tipos e cardinalidade, entre outras características.
- Os objectos de negócio podem ser definidos por interfaces e dados empresariais definidos na Web Services Description Language, bem como pelo esquema de XML das organizações de padrões da indústria ou de outros sistemas e aplicações. O Integration Designer pode importar estes objectos de negócio directamente.

O Integration Designer também fornece suporte para detectar dados de negócio em bases de dados e sistemas de informações empresariais e para gerar em seguida a definição de objecto de negócio do esquema de XML baseado em padrões desses dados de negócio. Os objectos de negócio gerados desta forma são muitas vezes referidos como *objectos de negócio específicos de aplicação* porque imitam a estrutura dos dados de negócio definidos no sistema de informações empresariais.

Quando um processo está a manipular dados de muitos sistemas de informações diferentes, pode ser importante transformar a representação distinta de dados de negócio (por exemplo, CustomerEIS1 e CustomerEIS2 ou OrderEIS1 e OrderEIS2) numa única representação canónica (por exemplo, Customer ou Order). A representação canónica é muitas vezes referida como o *objecto de negócio genérico*.

As definições de objectos de negócio, em particular objectos de negócio genéricos, são frequentemente utilizadas por mais de uma aplicação. Para suportar esta reutilização, o Integration Designer permite que os objectos de negócio sejam criados em bibliotecas que possam em seguida ser associados a múltiplos módulos de aplicação.

A Linguagem de Descrição de Serviços da Web (WSDL, Web Services Description Language) define os contratos para os serviços fornecidos e consumidos por um módulo de aplicação Arquitectura do Componente de Serviço (SCA, Service Component Architecture), bem como os contratos utilizados para criar os componentes no módulo de aplicação. Num contrato, uma WSDL pode representar tanto operações como objectos de negócio (que são definidos pelo esquema XML para representar os dados de negócio).

## Trabalhar com objectos de negócio

A SCA (Service Component Architecture) fornece o contexto para definir um módulo de aplicação, os serviços que fornece, os serviços que consome e a composição de componentes que fornecem a lógica empresarial do módulo de aplicação. Os objectos de negócio desempenham um papel importante na aplicação, definindo os dados de negócio que são utilizados para descrever os contratos de serviço e componente e os dados de negócio que os componentes manipulam.

O seguinte diagrama representa um módulo de aplicação de SCA e ilustra muitos dos locais onde o programador trabalha com objectos de negócio.

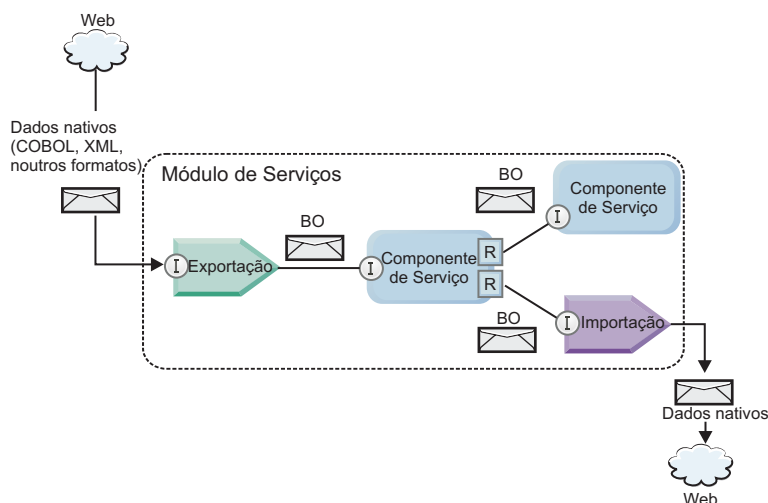


Figura 86. Os objectos de negócio representam os dados que fluem entre os serviços numa aplicação

**Nota:** Este tópico descreve como os objectos de negócio são utilizados pelos módulos de aplicação de SCA. Se estiver a utilizar interfaces de Java, os módulos de aplicação de SCA também podem processar objectos de Java.

## Modelo de programação de objecto de negócio

O modelo de programação de objecto de negócio consiste num conjunto de interfaces de Java que representam:

- A definição de objecto de negócio e os dados da instância
- Um conjunto de serviços que suportam as operações nos objectos de negócio

As definições de tipo de objecto de negócio são representadas pelas interfaces `commonj.sdo.Type` e `commonj.sdo.Property`. O modelo de programação de objecto de negócio fornece um conjunto de regras

para o mapeamento de informações de tipo complexo do esquema XML para a interface Type e cada um dos elementos na definição de tipo complexo para a interface Property.

As instâncias de objectos de negócio são representadas pela interface `commonj.sdo.DataObject`. O modelo de programação de objecto de negócio não tem tipo, o que significa que a mesma interface `commonj.sdo.DataObject` pode ser utilizada para representar diferentes definições de objecto de negócio, como Customer e Order. A definição das propriedades que podem ser definidas e obtidas de cada objecto de negócio é determinada pelas informações de tipo definidas no esquema de XML associado a cada objecto de negócio.

O comportamento do modelo de programação de objecto de negócio é baseado na especificação Service Data Object 2.1. Para mais informações, consulte a especificação SDO 2.1 para Java, os guias de iniciação e os javadocs na web: <http://osoa.org/display/Main/Service+Data+Objects+Specifications>.

Os serviços de objectos de negócio suportam várias operações de ciclo de vida (tais como criação, igualdade, análise e serialização) em objectos de negócio.

Para especificidades no modelo de programação de objectos de negócio, consulte Programação utilizando serviços de objectos de negócio e a Documentação de API e SPI gerada sobre objectos de negócio.

## **Associações, associações de dados e rotinas de tratamento de dados**

Tal como mostrado na Figura 41 na página 151, os dados de negócio que são utilizados para chamar serviços fornecidos por módulos de aplicação de SCA são transformados em objectos de negócio para que os componentes de SCA possam manipular os dados de negócio. Do mesmo modo, os objectos de negócio manipulados por componentes de SCA são convertidos no formato de dados requerido pelos serviços externos.

Em alguns casos, tal como a associação do serviço da Web, a associação utilizada para exportar e importar serviços automaticamente transforma os dados no formato apropriado. Noutros casos, tal como a associação de JMS, os programadores podem fornecer uma associação de dados ou uma rotina de tratamento de dados que converte formatos não nativos em objectos de negócio representados pela interface `DataObject`.

Para mais informações sobre como desenvolver associações de dados e rotinas de tratamento de dados, consulte “Rotinas de tratamento de dados” na página 57 e “Associações de dados” na página 59.

## **Componentes**

Os componentes de SCA definem os respectivos contratos de serviço de provisão e consumo utilizando uma combinação da Linguagem de Descrição dos Serviços da Web e do esquema de XML. Os dados de negócio que o SCA transmite entre componentes são representados como objectos de negócio utilizando a interface `DataObject`. O SCA verifica que estes tipos de objecto de negócio são compatíveis com o contrato de interface definido pelo componente a ser chamado.

As abstrações do modelo de programação para manipular objectos de negócio variam de componente para componente. O componente POJO e a primitiva Custom do componente de fluxo de mediação fornecem uma manipulação directa dos objectos de negócio activando a programação de Java directamente utilizando as interfaces e serviços de programação de objectos de negócio. A maior parte dos componentes fornecem abstrações de nível superior para a manipulação de objectos de negócio, mas também fornecem fragmentos de código Java para definir um comportamento personalizado nas interfaces e serviços de objectos de negócio.

Os objectos de negócio podem ser transformados utilizando a combinação do componente Mediação de Fluxo da Interface e Mapa de Objecto de Negócio ou o componente de fluxo de mediação e a respectiva primitiva XML Map. Estas capacidades de transformação de objectos de negócio são úteis para converter



objectos de negócio específicos da aplicação em e de objectos de negócio genéricos.

## Objectos de negócio especiais

Os objectos de mensagem de serviço e gráficos comerciais são dois tipos especializados de objectos de negócio que são utilizados para finalidades de aplicação específicas.

### Objecto de mensagem de serviço

Um objecto de mensagem de serviço (SMO) é um objecto de negócio especializado que é utilizado por componentes de fluxo de mediação para representar a colecção de dados associados a uma chamada para execução de serviço.

Um SMO tem uma estrutura de nível superior fixa que consiste em cabeçalhos, contexto, corpo e anexos (se estiverem presentes).

- Os cabeçalhos transportam informações relacionadas com a chamada para execução de serviço através de um determinado protocolo ou associação. Os exemplos são cabeçalhos de SOAP e cabeçalhos de JMS.
- Os dados de contexto transportam informações lógicas adicionais associadas à chamada para execução enquanto estão a ser processadas pelo componente de fluxo de mediação. Estas informações não fazem normalmente parte dos dados de aplicação enviados ou recebidos pelos clientes.
- O corpo do SMO transporta os dados de negócio da carga que representa a mensagem de aplicação principal ou dados de chamada para execução sob a forma de um objecto de negócio padrão.

O SMO também pode transportar dados de anexos para a chamada para execução de serviço da web utilizando SOAP com anexos.

Os fluxos de mediação executam essas tarefas como encaminhamento de pedidos e transformação de dados e o SMO fornece a vista combinada de conteúdo de cabeçalho e carga numa única estrutura unificada.

### Gráfico empresarial

Os gráficos comerciais são um objecto de negócio especial utilizado para fornecer suporte para a sincronização de dados nos cenários de integração.

Considere um exemplo em que dois sistemas de informação empresarial têm uma representação de uma encomenda específica. Quando a encomenda é alterada num sistema, uma mensagem pode ser enviada para o outro sistema para sincronizar os dados da encomenda. Os gráficos comerciais suportam a noção de envio apenas da parte da encomenda que mudou para o outro sistema e da sua anotação com informações de resumo de alterações para definir o tipo de alteração.

Neste exemplo, os gráficos comerciais da encomenda informarão o outro sistema que um dos itens de linha na encomenda foi eliminado e que a propriedade da data de expedição prevista da encomenda foi actualizada.

Os gráficos comerciais podem ser facilmente adicionados a objectos de negócio no Integration Designer. São muitas vezes encontrados em cenários em que os adaptadores WebSphere estão a ser utilizados e para suportar a migração das aplicações WebSphere InterChange Server.

### Modo de análise do objecto de negócio

O Integration Designer fornece uma propriedade de módulos e bibliotecas que pode utilizar para configurar o modo de análise XML para objectos de negócio para activo ou passivo.

- Se a opção for definida para *activo*, as sequências de bytes de XML são transmitidas de forma activa para criar o objecto de negócio.



- Se a opção for definida para *passivo*, o objecto de negócio é criado normalmente, mas a análise real da sequência de fluxo de XML é adiada e só parcialmente analisada quando as propriedades de objecto de negócio forem acedidas.

No modo de análise XML, os dados não XML são sempre analisado de forma activa para criar o objecto de negócio.

### **Considerações ao escolher o modo de análise do objecto de negócio**

O modo de análise do objecto de negócio determina a forma como os dados XML são analisados no tempo de execução. Um modo de análise do objecto de negócio é definido num módulo ou biblioteca quando é criado. É possível alterar o modo de análise do módulo ou biblioteca, mas deve ter atenção às implicações.

O modo de análise de objectos de negócio é definido ao nível do módulo e da biblioteca. Os módulos que foram criados numa versão do IBM Integration Designer anterior à versão 7 são executados no modo de análise activo sem necessidade de quaisquer alterações. Por predefinição, os módulos e bibliotecas que são criados no IBM Integration Designer versão 7 e posteriores irão receber o modo de análise mais adequado consoante vários factores, tais como o modo de análise dos projectos existentes no respectivo espaço de trabalho ou o modo de análise dos projectos dependentes ou outros projectos na mesma solução, etc. É possível alterar o modo de análise do objecto de negócio de um módulo ou biblioteca para se adequar à sua implementação, mas tem de ter em consideração o seguinte.

#### **Considerações**

- O modo de análise do objecto de negócio passivo processa os dados XML mais rapidamente. Contudo, existem diferenças de compatibilidade entre o modo activo e o passivo que é necessário ter em atenção antes de alterar a configuração de um módulo ou biblioteca. Estas diferenças irão afectar o comportamento de tempo de execução dos módulos. Para obter mais informações sobre que tipo de análise é otimizada para a aplicação, consulte "Benefits of using lazy versus eager parsing mode (Benefícios da utilização do modo de análise passivo versus activo)" nas ligações relacionadas.
- Um módulo só pode ser configurado para ser executado num modo de análise. As bibliotecas podem ser configuradas para suportar qualquer um dos modos de analisar ou ambos os modos de analisar. Uma biblioteca que esteja configurada para suportar ambos os modos de analisar pode ser referenciada por um módulo que utiliza o modo de análise activo e por um módulo que utiliza o modo de análise passivo. O modo de análise de uma biblioteca em tempo de execução é determinado pelos módulos que consultam a biblioteca. No tempo de execução, um módulo declara o seu modo de análise e este é utilizado pelo módulo e por quaisquer bibliotecas que esse módulo utilizar.
- Os módulos e as bibliotecas que estão configurados para modos de analisar diferentes são compatíveis nos seguintes casos:
  - Os módulos e bibliotecas configurados com o modo de análise passivo são compatíveis com bibliotecas que utilizam tanto o modo de análise passivo como ambos os modos de análise activo e passivo.
  - Os módulos e bibliotecas configurados com o modo de análise activo são compatíveis com as bibliotecas que utilizam tanto o modo de análise activo como ambos os modos de análise activo e passivo.
  - As bibliotecas configuradas com os modos de análise passivo e activo são compatíveis apenas com bibliotecas que utilizam ambos os modos de análise passivo e activo.
- Utilize o mesmo modo de análise para os módulos de interacção que comunicam utilizando a associação SCA. Se os módulos comunicarem utilizando modos de analisar diferentes, podem surgir problemas de desempenho.

### **Conceitos relacionados:**

“Benefícios da utilização do modo de análise passivo versus activo” na página 155

Algumas aplicações beneficiam de um modo de análise XML passivo, enquanto o desempenho de outros é melhorado através de um modo de análise activo. É recomendável que teste a sua aplicação em ambos os modos de análise para determinar qual o modo que melhor se adequa às características específicas da sua aplicação.

### **Benefícios da utilização do modo de análise passivo versus activo**

Algumas aplicações beneficiam de um modo de análise XML passivo, enquanto o desempenho de outros é melhorado através de um modo de análise activo. É recomendável que teste a sua aplicação em ambos os modos de análise para determinar qual o modo que melhor se adequa às características específicas da sua aplicação.

As aplicações que analisam grandes sequências de dados XML provavelmente verão melhoramentos no desempenho quando é utilizado o modo de análise de XML passivo. As vantagens de desempenho aumentam na medida que aumenta a sequência de bites XML e decresce a quantidade de dados na sequência de bites que é acedida pela aplicação.

**Nota:** O modo de análise passivo de objectos de negócios é suportado no WebSphere Process Server versão 7.0.0.3 e posteriores. Também é suportado no IBM Process Server. Os módulos e os módulos de mediação que incluem componentes de fluxo de mediação não são suportados.

As aplicações seguintes têm mais probabilidade de terem um melhor desempenho utilizando um modo de análise activo:

- Aplicações que analisam sequências de dados não XML
- Aplicações que utilizam mensagens criadas através do serviço BOFactory
- Aplicações que analisam mensagens de XML muito pequenas

### **Referências relacionadas:**

“Considerações ao escolher o modo de análise do objecto de negócio” na página 154

O modo de análise do objecto de negócio determina a forma como os dados XML são analisados no tempo de execução. Um modo de análise do objecto de negócio é definido num módulo ou biblioteca quando é criado. É possível alterar o modo de análise do módulo ou biblioteca, mas deve ter atenção às implicações.

### **Considerações sobre desenvolvimento e migração de aplicações**

Se estiver a configurar uma aplicação que tenha sido originalmente desenvolvida utilizando um modo de análise activo para utilizar agora um modo de análise passivo ou se estiver a planear alternar uma aplicação entre o modo de análise passivo e activo, tenha em atenção as diferenças entre os modos e as considerações quando estiver a alternar de modo.

### **Parâmetro identificador de erros**

Se a sequência de bytes de XML a ser analisada estiver incorrectamente formada, ocorrem excepções na análise.

- No modo de análise XML activo, estas excepções ocorrem assim que o objecto de negócio for analisado na sequência de XML interna.
- Se o modo de análise XML passivo for configurado, as excepções de análise ocorrem de forma dissimulada quando as propriedades de objecto de negócio forem acedidas e a parte do XML incorrectamente formado for analisada.

Para processar XML incorrectamente formado, seleccione uma das seguintes opções:

- Implementar um barramento de serviço de empresa nas extremidades para validar o XML interno
- Criar uma lógica de detecção de erros passiva no ponto onde as propriedades de objecto de negócio são acedidas

## **Pilhas de exceções e mensagens**

Visto que os modos de análise XML activo e passivo têm implementações subjacentes diferentes, os rastreios de pilha emitidos pelos serviços e interfaces de programação de objecto de negócio têm o mesmo nome de classe de excepção, mas poderão não conter a mesma mensagem de excepção ou conjunto translineado de classes de excepção específicas de implementação.

## **Formato de serialização de XML**

O modo de análise XML passivo fornece uma optimização do desempenho que tenta copiar XML não modificado da sequência de bytes interna para a sequência de bytes externa segundo a serialização. O resultado é um desempenho aumentado, mas o formato de serialização da sequência de bytes de XML externa poderá ser diferente se o objecto de negócio inteiro tiver sido actualizado no modo de análise XML passivo ou se tiver sido executado no modo de análise XML activo.

Embora o formato de serialização de XML possa não ser precisamente equivalente na forma sintáctica, o valor semântico fornecido pelo objecto de negócio é equivalente e independente dos modos de análise e o XML pode ser transmitido com segurança entre as aplicações em execução nos diferentes modos de análise com equivalência semântica.

## **Controlo de validação de instâncias de objecto de negócio**

O controlo de validação de ocorrências do modo de objectos de negócio de análise XML passivo fornece uma validação com a mais alta fidelidade dos objectos de negócio, em particular a validação dos valores de propriedade. Devido a estes melhoramentos, o controlo de validação de ocorrências de modo de análise passivo detecta problemas adicionais que não são detectados no modo de análise activo e fornece mensagens de erro mais detalhadas.

## **XML Maps, versão 602**

Os fluxos de mediação que se desenvolveram originalmente antes de WebSphere Integration Developer Versão 6.1 poderão conter as primitivas de mapeamento que utilizam um mapa ou folha de estilo que não podem ser executadas directamente no modo de análise XML passivo. Quando a aplicação migra para utilização no modo de análise XML passivo, os ficheiros de mapa associados com as primitivas de mapeamento podem ser actualizadas automaticamente pelo assistente de migração para ser executado no novo modo. No entanto, de uma primitiva de mapeamento se referir directamente a uma folha de estilos que foi editada manualmente, a folha de estilos não migra e não pode ser executada no modo de análise XML passivo.

## **APIs não publicadas privadas**

Se uma aplicação tirar partido de interfaces de programação de objecto de negócio não publicadas, privadas e específicas da implementação, é provável que a aplicação falhe a compilação quando o modo de análise é trocado. No modo de análise intenso, estas interfaces privadas são normalmente classes de implementação de objecto de negócio definidas pelo EMF (Eclipse Modeling Framework).

Em todos os casos, é recomendável que as APIs privadas sejam removidas da aplicação.

## **APIs Service Message Object EMF**

Um componente de mediação no IBM Integration Designer fornece a capacidade de manipular o conteúdo da mensagem utilizando as classes e interfaces de Java fornecidas no pacote `com.ibm.websphere.sibx.smobo`. No modo de análise XML passivo, as interfaces de Java no pacote `com.ibm.websphere.sibx.smobo` podem ainda ser utilizadas, mas provavelmente os métodos que se referem directamente às classes e interfaces de EMF (Eclipse Modeling Framework) ou que são herdados das interfaces de EMF irão falhar.

O ServiceMessageObject e o respectivo conteúdo não podem ser convertidos em objectos EMF no modo de análise XML passivo.

## Serviço BOMode

O serviço BOMode é utilizado para determinar se o modo de análise XML actualmente em execução é activo ou passivo.

## Migração

Todas as aplicações anteriores à versão 7.0.0.0 são executadas no modo de análise XML activo. Quando são migradas em tempo de execução utilizando as ferramentas de migração em tempo de execução BPM, continuam a ser executadas no modo de análise XML activo.

Para activar uma aplicação anterior à versão 7.0.0.0 para ser configurada para utilizar o modo de análise XML passivo, primeiro tem de utilizar o Integration Designer para migrar os artefactos da aplicação. Após a migração, configura a aplicação para utilizar a análise de XML passiva.

Consulte Migrar artefactos de origem para obter informações sobre como migrar artefactos no Integration Designer e consulte Configurar o modo de análise de objecto de negócio de módulos e bibliotecas para obter informações sobre como definir o modo de análise.

---

## Relações

Um relação é uma associação entre duas ou mais entidades de dados, tipicamente objectos de negócios. Em IBM Business Process Manager Advanced, as relações podem ser utilizadas para transformar dados equivalentes em objectos de negócios e outros dados, mas que são representados de forma diferente ou podem ser utilizados para criar associações em objectos diferentes que se encontram em aplicações diferentes. Podem ser partilhados entre aplicações, entre soluções e até entre produtos.

O serviço de relações do IBM Business Process Manager Advanced fornece a infra-estrutura e operações para gerir relações. Uma vez que permite lidar com objectos de negócio independentemente da respectiva localização, pode fornecer uma vista holística unificada em todas as aplicações de uma empresa e funciona como bloco de construção para soluções BPM. Dado que as relações podem ser extensíveis e geridas, podem ser utilizadas em soluções de integração complexas.

### O que são relações?

Uma relação é uma associação entre objectos de negócio. Cada um dos objectos de negócio de uma relação é denominado um *participante* na relação. Cada um dos participantes da relação distingue-se de outros participantes com base na *função*, na relação em questão. Uma relação possui uma lista de funções.

A *definição* da relação descreve cada uma das funções e especifica a forma como as funções estão relacionadas. Descreve também a "forma" global da relação. Por exemplo, esta função apenas pode conter um participante, mas a outra função pode conter todos os participantes que forem necessários. Pode, por exemplo, ser definida uma relação *veículo-proprietário*, em que o proprietário pode possuir diversos veículos. Por exemplo, uma ocorrência pode conter os participantes seguintes para cada uma das funções:

- Veículo (Ferrari)
- Proprietário (Joaquim)

A definição da relação é um modelo para a *ocorrência* da relação. A ocorrência da relação é a instanciação em tempo de execução. No exemplo do *veículo-proprietário*, uma ocorrência poderá descrever qualquer das associações seguintes:

- Joaquim proprietário Ferrari
- Sara proprietária Mazda

- Bruno Proprietário Ferrari

A utilização de relações liberta-o da necessidade de personalizar a criação do controlo da relação persistente na lógica empresarial. Em determinados cenários, o serviço de relações faz um trabalho completo. Consulte o exemplo descrito na secção em Relações de identidade.

## Cenários

Este é um exemplo típico de uma situação em que uma solução de integração poderá utilizar relações. Uma grande empresa adquire diversas empresas ou unidades de negócio. Cada uma das unidades de negócio utiliza um software diferente para supervisionar o pessoal e os blocos de notas. A empresa necessita de uma forma para supervisionar os empregados e os respectivos blocos de notas. Pretende-se uma solução que permita:

- Ver todos os empregados das diversas unidades de negócio como se estivessem numa base de dados
- Obter uma vista única de todos os blocos de notas
- Permitir aos empregados ligarem-se ao sistema e comprarem um bloco de notas
- Acomodar os diferentes sistemas de aplicações empresariais das várias unidades de negócio

Para conseguir isso, a empresa necessita de uma forma para assegurar, por exemplo, que o John Smith e o John A. Smith em aplicações diferentes são vistos como o mesmo empregado. Por exemplo, é necessário obter uma forma de consolidar uma entidade única em múltiplos espaços de aplicação.

Os cenários mais complexos de relações envolvem criar processos BPEL que chamam relações em diferentes objectos encontrados em diversas aplicações. Nos cenários complexos de relações, os objectos de negócio residem na solução da integração, e não na aplicação. O serviço de relações fornece uma plataforma para gerir relações de forma persistente. Antes da utilização do serviço de relação, era necessário criar um objecto de serviço persistente. Dois exemplos de cenários de relações complexas, são:

- Existe um objecto de negócio de **car** com um número VIN na aplicação SAP e, pretende-se controlar o facto do veículo ser propriedade de terceiros. No entanto, a relação de propriedade está com alguém da aplicação PeopleSoft. Neste padrão de relações, existem duas soluções, pelo que é necessário criar uma ponte entre ambas.
- Uma grande empresa de retalho pretende ser capaz de supervisionar mercadoria devolvida para crédito ou devolução do dinheiro. Existem duas aplicações diferentes envolvidas: um sistema de gestão de compras (OMS, order management system) para compras e um sistema de gestão de devoluções (RMS, returns management system) para devoluções. Os objectos de negócio encontram-se em mais do que uma aplicação e é necessário mostrar as relações existentes entre elas.

## Padrões de utilização comum

Os padrões de relações mais comuns são padrões de *equivalência*. São baseados em referências cruzadas ou correlação. Estes são dois tipos de relações que se enquadram neste padrão: *não-identidade* e *identidade*.

- **relações de não-identidade** estabelecem associações entre objectos de negócio ou outros dados numa base um-para-muitos ou muitos-para-muitos. Para cada uma das ocorrências de relação, podem existir uma ou mais ocorrências de cada um dos participantes. Um tipo de relação não-identidade é uma relação de procura estática. Um exemplo disto é uma relação na qual **CA** numa aplicação SAP está relacionada com **California** numa aplicação Siebel.

•

**Relações de identidade** estabelecem associações entre objectos de negócio ou outros dados numa base um-para-um. Para cada ocorrência de relação, apenas poderá existir uma ocorrência de cada um dos participantes. As relações de identidade capturam referências cruzadas entre objectos de negócio equivalentes semanticamente, mas isso é identificado de forma diferente em aplicações diferentes. Cada um dos participantes da relação está associado a um objecto de negócio que possui um valor (ou

combinação de valores) que identifica esse objecto de forma única. As relações de identidade transformam tipicamente os atributos chave de objectos de negócio, tais como números de ID e códigos do produto.

Por exemplo, se existir um objecto de negócio **car** nas aplicações SAP, PeopleSoft, e Siebel, e se pretender criar uma solução que faça a respectiva sincronização, normalmente necessitaria de introduzir uma lógica de sincronização de relação manual em seis mapas:

SAP -> genérico

genérico -> SAP

PeopleSoft-> genérico

genérico-> PeopleSoft

Siebel-> genérico

genérico-> Siebel

No entanto, se foram utilizadas relações na solução, o serviço de relações fornece implementações de padrões pré-construídos fazem a gestão de todas as ocorrências destas relações.

## Ferramentas para funcionar com relações

O *editor de relações* no Integration Designer é a ferramenta utilizada para modelar e conceber relações e funções de integração de negócio. Para obter informações de segundo plano e tarefas detalhadas sobre criar relações e utilizar o editor de relação, consulte Criar relações.

O *serviço de relações* é um serviço de infra-estrutura no IBM Business Process Manager que mantém relações e funções no sistema e, fornece operações para gerir relações e funções.

O *gestor de relações* é a interface administrativa para gerir relações. É acessado através das páginas do Gestor de Relações (Relationship Manager) na consola administrativa.

As relações podem ser chamadas programaticamente através das APIs do serviço de relações.

## Serviço de relações

O serviço de relações guarda dados de relações em tabelas de relações, onde mantém o controlo de valores específicos da aplicação entre aplicações e entre soluções. O serviço de relações fornece operações para gestão de relações e funções.

## Como funcionam as relações

As relações e as funções são definidas utilizando a interface gráfica da ferramenta de edição de relações no Integration Designer. O serviço de relações guarda os dados de correlação em tabelas na base de dados da relação na origem de dados predefinida especificada durante a configuração do serviço de relação. Uma tabela separada (por vezes denominada tabela participante) guarda informações para cada um dos participantes na relação. O serviço de relações utiliza estas tabelas de relações para controlar os valores específicos da aplicação relacionados e as informações actualizadas propagadas em todas as soluções.

As relações que são artefactos de negócio, são implementadas num projecto ou numa biblioteca partilhada. Durante a primeira implementação, o serviço de relações preenche os dados.

Em tempo de execução, quando os mapas ou outros componentes do IBM Business Process Manager necessitam de uma ocorrência de relação, as ocorrências da relação são actualizadas ou obtidas, dependendo do cenário.

Dados da relação e da ocorrência da função podem ser manipulados através de três formas:



- Chamada para execução de fragmentos do componente IBM Business Process Manager Java do serviço de relações das APIs
- Transformações da relação do servidor de mapeamento de objectos de negócio do IBM Business Process Manager
- A ferramenta de gestão de relações

Para obter informações detalhadas sobre a tarefa de criação de relações, identificar tipos de relações e utilizar o editor de relações, consulte o tópico Criar relações.

## Gestor de relações

O gestor de relações é a interface administrativa para gestão de relações. É acessado através das páginas do Gestor de Relações (Relationship Manager) na consola administrativa.

O gestor de relações fornece uma interface gráfica do utilizador para criar e manipular dados de relações e de funções em tempo de execução. É possível gerir entidades de relações em todos os níveis: ocorrência da relação, ocorrência da função, dados do atributo e níveis de dados da propriedade. A utilização do gestor de relações, permite:

- Ver uma lista de relações no sistema e informações detalhadas para relações individuais
- Gerir ocorrências de relações:
  - Consultar dados de relações para ver subconjuntos de dados de ocorrências
  - Consultar dados de relações para ver subconjuntos de dados de ocorrências utilizando vistas de base de dados
  - Ver uma lista de ocorrências de relações que correspondem a uma consulta de relação e informações detalhadas sobre uma ocorrência
  - Editar os valores da propriedade para uma ocorrência de relação
  - Criar e eliminar ocorrências de relações
- Gerir funções e ocorrências de funções:
  - Ver detalhes sobre uma função ou uma ocorrência de função
  - Editar propriedades de ocorrência de uma função
  - Criar e eliminar ocorrências de função para uma relação
  - Remover alterações de dados de uma ocorrência de relação para um ponto no tempo em que tenha conhecimento que os dados são fidedignos
- Importar dados de uma relação estática existente para o sistema ou exportar dados de uma relação estática existente para um ficheiro RI ou CSV
- Remover esquema da relação e os dados do repositório quando a aplicação a utilizar a relação é desinstalada

## Relações em ambientes de Implementação de Rede

As relações podem ser utilizadas em ambientes de Implementação de Rede (ND, Network Deployment), sem qualquer configuração adicional.

Em ambientes de Implementação de Rede (ND, Network Deployment), as relações são instaladas num conjunto de aplicação. As relações são então visíveis no conjunto de unidades e todos os servidores do conjunto de unidades têm acesso aos dados da ocorrência guardados na base de dados da relação. A capacidade de executar o serviço de relação num ambiente ND torna-o escalável e altamente disponível.

O gestor de relações permite a gestão de relações ao nível de diferentes conjuntos de unidades através de uma interface administrativa centralizada. É possível ligar o gestor de relações a um servidor num conjunto de unidades seleccionando a respectiva relação MBean.



## APIs do serviço de relações

As relações podem ser chamadas programaticamente através das APIs do serviço de relações, dentro ou fora dos mapas de objectos de negócio.

Estão disponíveis três tipos de API:

- APIs para manipulação de ocorrências de relações (incluindo a criar, actualizar, eliminar directamente dados da ocorrência)
- APIs para suporte de padrões de relações (incluindo `correlate()`, `correlateforeignKeyLookup`)
- Padrões de procura de relações (APIs de procura)

---

## O barramento de serviços empresariais no IBM Business Process Manager

O IBM Business Process Manager suporta a integração de servidos de aplicação, incluindo as mesmas funções do WebSphere Enterprise Service Bus.

### Ligar serviços através de um barramento de serviço de empresas

Com um barramento de serviço de empresas (ESB, enterprise service bus), pode maximizar a flexibilidade de um SOA. Os participantes numa interacção de serviço são ligados ao ESB, em vez de um ao outro, directamente.

Quando o solicitador de serviços se liga ao ESB, este fica responsável por entregar os seus pedidos, por meio de mensagens, a um fornecedor de serviços que oferece a função e qualidade de serviço necessárias. O ESB facilita as interacções solicitador-fornecedor e endereça protocolos díspares, padrões de interacção ou funções de interacção. Um ESB pode também activar ou melhorar a monitorização e a gestão. O ESB fornece as funções de virtualização e gestão que implementam e prolongam as funções centrais da SOA.

O ESB resume as seguintes funções:

#### Localização e identidade

Os participantes não necessitam de saber a localização ou identidade dos outros participantes. Por exemplo, os solicitadores não necessitam de saber que um pedido pode ser reparado por qualquer um dos vários fornecedores; podem ser adicionados ou removidos fornecedores de serviços sem ruptura.

#### Protocolo de interacção

Os participantes não necessitam de partilhar o mesmo protocolo de comunicação ou estilo de interacção. Por exemplo, um pedido expresso como SOAP sobre HTTP pode ser prestado por um servidor que apenas compreenda SOAP sobre Serviço de Mensagens Java (JMS).

#### Interface

Os solicitadores e os fornecedores não necessitam de concordar numa interface comum. Um ESB reconcilia diferenças ao converter mensagens de pedido e resposta numa forma esperada pelo fornecedor.

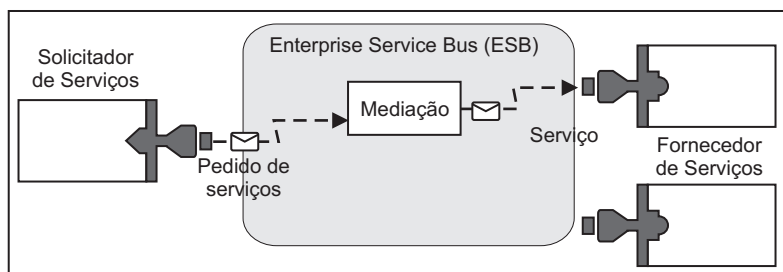
#### Qualidades do serviço (de interacção)

Os participantes, ou administradores de sistema, declaram os seus requisitos de qualidade de serviço, incluindo a autorização de pedidos, encriptação e descriptação de conteúdos de mensagens, auditoria automática de interacções de serviços e como os seus pedidos devem ser encaminhados (por exemplo, optimizar para velocidade ou custo).

Interpor o ESB entre participantes permite-lhe modular a sua interacção através de uma construção lógica chamada *mediação*. As mediações operam em mensagens em curso entre solicitadores e fornecedores. Por exemplo, as mediações podem ser utilizadas para encontrar serviços com características específicas que um solicitador está a pedir, ou para resolver as diferenças de interface entre solicitadores e fornecedores. As mediações podem ser encadeadas sequencialmente para interacções complexas.

Ao utilizar mediações, um barramento de serviço de empresa efectua as seguintes acções entre o solicitador e o serviço:

- *Encaminhar* mensagens entre serviços. Um barramento de serviço de empresas oferece uma infra-estrutura de comunicação comum que pode ser utilizada para ligar serviços e, assim, as funções de negócios que representam, sem a necessidade que os programadores escrevam e mantenham uma lógica de conectividade complexa.
- *Converter* os protocolos de transporte entre o solicitador e o serviço. Um barramento de serviço de empresas fornece uma forma consistente e baseada em padrões de integrar funções de negócios que utilizam padrões IT diferentes. Isto permite a integração de funções de negócios que normalmente não podiam comunicar, como ligar aplicações em silos departamentais ou permitir que aplicações em diferentes companhias participem em interacções de serviços.
- *Converter* formatos de mensagens entre solicitador e serviço. Um barramento de serviço de empresas permite que funções de negócios troquem informação em diferentes formatos, com o barramento a assegurar que a informação entregue a uma função de negócios está no formato requerido pela aplicação.
- *Processar* eventos de negócios de origens desiguais. Um barramento de serviço de empresas suporta interacções baseadas em eventos, para além das trocas de mensagens para processarem pedidos de serviço.



*Figura 87. Um barramento de serviço de empresas. O barramento de serviço de empresas está a encaminhar mensagens entre aplicações, que são solicitadores ou fornecedores de serviços. O barramento está a converter os protocolos de transporte e a converter os formatos de mensagens entre solicitadores e fornecedores. Nesta figura, cada aplicação utiliza um protocolo diferente (representado pelas formas geométricas dos seus conectores) e formatos de mensagens diferentes.*

Ao utilizar o barramento de serviço de empresa pode focar-se no seu negócio principal, em vez de se focar nos seus sistemas de computador. Pode alterar ou adicionar aos serviços, se necessário; por exemplo, para responder às alterações nos requisitos de negócios, para adicionar uma capacidade de serviço extra ou para adicionar nova capacidades. Pode fazer as alterações necessárias ao reconfigurar o barramento, com pouco ou nenhum impacto em serviços e aplicações existentes que utilizem o barramento.

## **Infra-estrutura de mensagens de barramento de serviços da empresa**

O IBM Business Process Manager inclui funções de barramento de serviços da empresa. O IBM Business Process Manager suporta a integração de tecnologias orientadas para serviços, orientadas para mensagens e orientadas para eventos, para fornecer uma infra-estrutura de mensagens baseada em standards num barramento de serviços da empresa integrado.

As funções do serviço empresarial que podem ser utilizadas nas aplicações da empresa, fornecem não só uma camada de transporte como também, suporte de mediação para facilitar as interacções do serviço. O barramento de serviços da empresa é criada a partir de padrões abertos e da Arquitectura Orientada para Serviços (SOA, service-oriented architecture). É baseado na infra-estrutura robusta Java EE e serviços de plataforma associados, fornecidos por IBM WebSphere Application Server Network Deployment.

IBM Business Process Manager is powered by the same technology available with IBM WebSphere Enterprise Service Bus. Esta função faz parte da funcionalidade subjacente do IBM Business Process Manager e não é necessária qualquer licença adicional do WebSphere Enterprise Service Bus para tirar partido destas funções.

No entanto, é possível implementar licenças autónomas adicionais do WebSphere Enterprise Service Bus na empresa para expandir o alcance da conectividade das soluções de integração do processo suportadas pelo IBM Business Process Manager. Por exemplo, o WebSphere Enterprise Service Bus pode ser instalado juntamente com uma aplicação SAP para alojar um IBM WebSphere Adapter para SAP e para transformar mensagens SAP antes de enviar essa informação pela rede para um processo empresarial coreografado por IBM Business Process Manager.

### **Sistema central de destino para o serviço de mensagens ou fila**

Um sistema central de destino para o serviço de mensagens ou fila fornece a função de serviço de mensagens num servidor. Um servidor torna-se o sistema central de destino para o serviço de mensagens quando é configurado como destino do serviço de mensagens.

Um motor de serviço de mensagens é executado num servidor. O motor do serviço de mensagens fornece funções de serviço de mensagens e um ponto de ligação entre as aplicações e o barramento. A comunicação assíncrona da Arquitectura de Componente de Serviço (SCA, Service Component Architecture), a importação e exportação JMS, o processamento interno assíncrono utilizam filas de mensagens no motor do serviço de mensagens.

O ambiente de implementação liga a origem da mensagem ao destino da mensagem através do barramento durante a implementação dos módulos da aplicação. Conhecer a origem da mensagem e o destino da mensagem ajuda a determinar o tipo de ambiente de implementação necessário.

As aplicações podem armazenar dados sistemáticos num armazenamento de dados, que é um conjunto de tabelas numa base de dados ou esquema ou num armazenamento de ficheiros. O motor do serviço de mensagens utiliza uma ocorrência de uma origem de dados JDBC para interagir com essa base de dados.

A configuração do sistema central de destino para o serviço de mensagens deve ser efectuada durante a definição do ambiente de implementação utilizando **Server (Servidor)** na consola administrativa ou designando o servidor como sistema central de destino durante a instalação do software.

#### **Armazenamentos de dados:**

Todos os motores de serviço de mensagens podem utilizar um armazenamento de dados, que é um conjunto de tabelas numa base de dados ou esquema que armazena dados sistemáticos.

Todas as tabelas do armazenamento de dados são retidas no mesmo esquema de base de dados. É possível criar cada um dos armazenamentos de dados numa base de dados individual. Em alternativa, é possível criar múltiplos armazenamentos de dados na mesma base de dados, utilizando cada um dos armazenamentos de dados um esquema diferente.

O motor do serviço de mensagens utiliza uma ocorrência da origem de dados JDBC para interagir com a base de dados que contém o armazenamento de dados para esse motor de serviço de mensagens.

#### **Fornecedores JDBC**

É possível utilizar fornecedores JDBC para interagir aplicações com bases de dados relacionais.

As aplicações utilizam fornecedores JDBC para interagir com bases de dados relacionais. O fornecedor JDBC fornece a classe de implementação do controlador JDBC específico para acesso a um tipo específico de base de dados. Para criar um conjunto de ligações para determinada base de dados, deverá associar uma origem de dados com o fornecedor JDBC. Em conjunto, o fornecedor JDBC e os objectos de origem

de dados são funcionalmente equivalentes ao fornecedor de ligação Java EE Connector Architecture (JCA), que proporciona a conectividade com bases de dados não-relacionais.

Consulte os exemplos para Configuração típica de ambiente autónomo e Configuração típica de ambiente de implementação no tópico anterior.

Para obter mais informações acerca de fornecedores JDBC, consulte “JDBC providers (fornecedores JDBC)” no Information Center do WebSphere Application Server.

## **Barramento de integração de serviço para IBM Business Process Manager**

O barramento de integração de serviço é um mecanismo de comunicações gerido que suporta integração de serviços através do serviço de mensagens assíncrono e síncrono. Um barramento consiste em interligar motores de serviço de mensagens que gerem recursos de barramento. É uma das tecnologias WebSphere Application Server em que o IBM Business Process Manager é baseado.

Um único barramento de serviço de integração é único motor de serviço de mensagens utiliza o mesmo esquema de base de dados que a base de dados do produto por predefinição. Cada ambiente de implementação tem o próprio barramento. O único barramento é denominado **BPM.deployment\_environment\_name.Bus**.

Um barramento destino é um endereço lógico ao qual as aplicações podem ligar-se como produtores, consumidores ou ambos. Uma fila destino é um barramento destino utilizado para serviço de mensagens ponto a ponto.

Cada um dos barramentos possui um ou mais membros de barramento, em que cada um é um servidor ou um conjunto.

A *topologia de barramento* é a disposição física do servidor de aplicações, motores do serviço de mensagens, gestores de filas WebSphere MQ e, o padrão de ligações de barramento e ligações entre eles, que representam o barramento do serviço da empresa.

## **Aplicações e módulos de serviços**

Um módulo de serviço é um módulo de Arquitectura de Componentes de Serviços (SCA) que fornece serviços no ambiente em tempo de execução. Quando é implementado um módulo de serviço para o IBM Business Process Manager, é construída uma aplicação de serviço associada que é empacotada como um ficheiro de Arquivo de Empresas (EAR).

Os módulos de serviço são unidades básicas de implementação e podem conter componentes, bibliotecas e módulos de simulação utilizados pela aplicação de serviços associada. Os módulos de serviço têm exportações e, opcionalmente, importações para definir as relações entre módulos e solicitadores e fornecedores de serviços. O WebSphere Process Server suporta módulos para serviços de negócios e módulos de mediação. Ambos os módulos e módulos de mediação são tipos de módulos SCA. Um módulo de mediação permite a comunicação entre aplicações ao converter a chamada para execução de serviços para um formato compreendido pelo destino, passando o pedido e devolvendo o resultado ao autor. Um módulo para um serviço de negócios implementa a lógica para um processo empresarial. Contudo, um módulo pode também incluir a mesma lógica de mediação que pode ser empacotada num módulo de mediação.

## **Implementar uma aplicação de serviços**

O processo de implementar um ficheiro EAR que contém uma aplicação de serviços é o mesmo que o de implementar um ficheiro EAR. Os valores de parâmetros de mediação podem ser alterados na altura de implementação. Depois de ser implementado um ficheiro EAR que contém um módulo SCA, podem ser revistos detalhes acerca da aplicação de serviços e o seu módulo associado. Pode ser visto como um módulo de serviço está ligado a solicitadores de serviços (através de exportações) e solicitadores de serviços (através de importações).

## Ver detalhes de módulos SCA

Os detalhes de módulos de serviços que podem ser vistos dependem do módulo SCA. Incluem os seguintes atributos.

- Nome do módulo SCA
- Descrição de módulo SCA
- Nome de aplicação associada
- Informação de versão de módulo SCA, se o módulo tiver uma versão
- Importações de módulos SCA:
  - As interfaces de importações são definições abstractas que descrevem como um módulo SCA acede a um serviço.
  - As associações de importação são definições concretas que especificam o mecanismo físico pelo qual um módulo SCA acede a um serviço. Por exemplo, utilizando SOAP/HTTP.
- Exportações de módulos SCA:
  - As interfaces de exportação são definições abstractas que descrevem como os solicitadores de serviços acedem a um módulo SCA.
  - As associações de exportação são definições concretas que especificam o mecanismo físico pelo qual um solicitador de serviços acede a um módulo SCA e, indirectamente, a um serviço.
- Propriedades de módulos SCA

## Importações e associações de importação

Importações definem interações entre módulos SCA e fornecedores de serviços. Os módulos SCA utilizam importações para permitir que os componentes acedam aos serviços externos (serviços que estão fora do módulo SCA) utilizando uma representação local. As associações de importações definem a forma específica em que um serviço externo é acedido.

Se os módulos SCA não necessitarem de aceder serviços externos, não necessitam de ter importações. Os módulos de mediação têm, geralmente, uma ou mais importações que são utilizadas para passar mensagens ou pedidos aos seus destinos.

## Interfaces e associações

Uma importação de módulo SCA necessita pelo menos de uma interface e tem uma única associação.

- As interfaces de importação são definições abstractas que definem um conjunto de operações utilizando a Linguagem de Descrição de Serviços da Web (Web Services Description Language) (WSDL), uma língua XML para descrever serviços da web. Um módulo SCA pode ter várias interfaces de importação.
- As associações de importação são definições concretas que especificam o mecanismo físico que os módulos SCA utilizam para aceder a um serviço.

## Associações de importação suportadas

O IBM Business Process Manager suporta as seguintes associações de importações:

- As associações SCA ligam módulos SCA a outros módulos SCA. As associações SCA também são referidas como associações padrão.
- As associações de Serviço da Web permitem que componentes chamem serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.

Pode ser utilizada uma associação SOAP1.1/HTTP ou SOAP1.2/HTTP baseada no API Java de Serviços da Web XML (JAX-WS), que permite a interacção com serviços utilizando associações literais de documentos ou RPC e que utiliza processadores JAX-WS para personalizar chamadas para execução. É fornecida outra associação SOAP1.1/HTTP para permitir a interacção com serviços que

utilizam uma associação de codificação RPC ou onde houver uma necessidade de utilizar processadores JAX-RPC para personalizar as chamadas para execução.

- As associações HTTP permitem-lhe aceder a aplicações utilizando o protocolo HTTP.
- As associações de importação de JavaBeans de Empresa (Enterprise JavaBeans) (EJB) permitem aos componentes SCA chamar serviços fornecidos pela lógica de negócios Java EE executados num servidor Java EE.
- As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é alcançada através da utilização de adaptadores de recursos.
- As associações 1.1 do Serviço de Mensagens Java (JMS) permitem interoperacionalidade com o fornecedor de mensagens padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente.
- As associações genéricas JMS permitem interoperacionalidade com fornecedores JMS de terceiros que se integram com o WebSphere Application Server utilizando o JMS Application Server Facility (ASF).
- As associações JMS do WebSphere MQ permitem interoperacionalidade com os fornecedores JMS baseados no WebSphere MQ. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente. Se quiser utilizar o WebSphere MQ como fornecedor JMS, utilize associações JMS do WebSphere MQ.
- As associações do WebSphere MQ permitem interoperacionalidade com o WebSphere MQ. As associações do WebSphere MQ apenas podem ser utilizadas com gestores de fila remotos por via de uma ligação de cliente WebSphere MQ; não podem ser utilizadas com gestores de fila locais. Deverão ser utilizadas associações do WebSphere MQ se for desejado comunicar com aplicações nativas do WebSphere MQ.

### **Chamada para execução dinâmica de serviços**

Podem ser chamados serviços através de qualquer associação de importação suportada. Um serviço é normalmente encontrado num ponto de terminação especificado na importação. Este ponto de terminação chama-se ponto de terminação estático. É possível chamar um serviço diferente ao substituir o ponto de terminação estático. A substituição dinâmica de pontos de terminação estáticos permite chamar um serviço em qualquer outro ponto de terminação através de qualquer associação de importação suportada. A chamada dinâmica de serviços também permite chamar para execução um serviço em que a associação de importação suportada não tem um ponto de terminação estático.

Uma importação com uma associação associada é utilizada para especificar o protocolo e a sua configuração para uma chamada para execução dinâmica. A importação utilizada para a invocação dinâmica pode ser ligada ao componente de chamada ou pode ser seleccionada de forma dinâmica no tempo de execução.

Para chamadas para execução de Serviços da web e SCA, é possível fazer uma chamada para execução dinâmica sem uma importação, com o protocolo e a configuração deduzidas do URL do ponto de terminação. O tipo de destino de chamada para execução é identificado do URL do ponto de terminação. Se for utilizada uma importação, o URL deve ser compatível com o protocolo da associação de importação.

- Um URL de SCA indica a chamada para execução de outro módulo SCA.
- Um URL HTTP ou JMS indica, por defeito, a chamada para execução de um serviço da web; para estes URLs é possível fornecer um valor de tipo de associação adicional que indique que o URL representa uma invocação por via de uma associação HTTP ou JMS.
- Para um URL HTTP de serviço da web, o normal é utilizar SOAP 1.1, e pode ser especificado um valor de associação que indique a utilização de SOAP 1.2.



## Exportações e associações de exportações

As exportações definem interações entre módulos SCA e solicitadores de serviços. Os módulos SCA utilizam exportações que oferecem serviços a outros. As associações de exportações definem a forma específica que um módulo SCA é acessado por solicitadores de serviços.

## Interfaces e associações

Uma exportação de módulo SCA necessita de pelo menos uma interface.

- As interfaces de exportações são definições abstractas que definem um conjunto de operações utilizando a Linguagem de Descrição de Serviços da Web (WSDL, Web Services Description Language), uma língua XML para descrever serviços da web. Um módulo SCA pode ter várias interfaces de exportação.
- As associações de exportação são definições concretas que especificam o mecanismo físico que os solicitadores de serviços utilizam para aceder a um serviço. Normalmente, uma exportação de módulo SCA tem uma associação especificada. Uma exportação sem nenhuma associação especificada é interpretada pelo ambiente em tempo de execução como uma exportação com uma associação SCA.

## Associações de exportação suportadas

O IBM Business Process Manager suporta as seguintes associações de exportação:

- As associações SCA ligam módulos SCA a outros módulos SCA. As associações SCA também são referidas como associações padrão.
- As associações de Serviços da Web permitem que as exportações sejam chamadas como serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.

Pode utilizar uma associação SOAP1.1/HTTP ou SOAP1.2/HTTP baseada no API Java para Serviços da Web XML (JAX-WS), o que permite a interação com serviços utilizando associações literais de documentos ou RPC e que utiliza processadores JAX-WS para personalizar as chamadas para execução. É fornecida outra associação SOAP1.1/HTTP para permitir a interação com serviços que utilizam uma associação de codificação RPC ou onde houver uma necessidade de utilizar processadores JAX-RPC para personalizar as chamadas para execução.

- As associações HTTP permitem que as exportações sejam acedidas utilizando o protocolo HTTP.
- As associações de exportação Enterprise JavaBeans (EJB) permitem que os componentes SCA sejam expostos como EJBs, para que a lógica de negócios Java EE possa chamar componentes SCA que de outra forma lhes estariam indisponíveis.
- As associações do Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Esta comunicação é alcançada através da utilização de adaptadores de recursos.
- As associações 1.1 do Serviço de Mensagens Java (JMS) permitem interoperacionalidade com o fornecedor de mensagens padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente.
- As associações genéricas JMS permitem interoperacionalidade com fornecedores JMS de terceiros que se integram com o WebSphere Application Server utilizando o JMS Application Server Facility (ASF).
- As associações JMS do WebSphere MQ permitem interoperacionalidade com os fornecedores JMS baseados no WebSphere MQ. A classe de Mensagens JMS e os seus cinco subtipos (Texto, Bytes, Objecto, Fluxo e Mapa) são suportados automaticamente. Se quiser utilizar o WebSphere MQ como fornecedor JMS, utilize associações JMS do WebSphere MQ.
- As associações do WebSphere MQ permitem interoperacionalidade com o WebSphere MQ. Utilize uma ligação remota (ou de cliente) para se ligar a um gestor de fila MQ numa máquina remota. Uma ligação local (ou associações) é uma ligação directa ao WebSphere MQ. Isto pode apenas ser utilizado para uma ligação a um gestor de fila MQ na mesma máquina. O WebSphere MQ vai permitir ambos os tipos de ligação, mas as associações MQ apenas suportam a ligação "remota" (ou "de cliente").



## Módulos de mediação

Os módulos de mediação são módulos de Arquitetura de Componentes de Serviços (Service Component Architecture) (SCA) que podem alterar o formato, conteúdo ou destino dos pedidos de serviços.

Os módulos de mediação operam em mensagens que estão em curso entre solicitadores de serviços e fornecedores de serviços. Pode encaminhar mensagens para vários fornecedores de serviços e para corrigir o conteúdo ou a forma da mensagem. Os módulos de mediação podem disponibilizar funções como o registo de mensagens e processamento de erros personalizados de acordo com os seus requisitos.

Certos aspectos dos módulos de mediação podem ser alterados, a partir da consola administrativa, sem ser necessário implementá-los novamente.

## Componentes de módulos de mediação

Os módulos de mediação contêm os seguintes itens:

- Importações, que definem interações entre os módulos SCA e os fornecedores de serviços. Permitem que os módulos SCA chamem serviços externos como se fossem locais. As importações de módulos de mediação podem ser vistas e a sua associação alterada.
- Exportações, que definem interações entre módulos SCA e solicitadores de serviços. Permitem que um módulo SCA ofereça um serviço e defina as interfaces externas (pontos de acesso) de um módulo SCA. As exportações dos módulos de mediação podem ser vistas.
- Os componentes SCA, que são blocos de construção para módulos SCA modules, como os módulos de mediação. É possível criar e personalizar graficamente módulos e componentes SCA, utilizando o Integration Designer. Depois da implementação de um módulo de mediação, é possível personalizar certos aspectos do mesmo a partir da consola administrativa, sem ser necessário implementá-lo novamente.

Normalmente, os módulos de mediação contêm um tipo específico de componente SCA chamado *componente de fluxo de mediação*. Os componentes de fluxo de mediação definem fluxos.

Um componente de fluxo de mediação pode algumas ou nenhuma primitivas de mediação. O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação que fornecem funcionalidade ao encaminhamento e conversão de mensagens. Para uma flexibilidade de primitiva de mediação adicional, utilize a primitiva de mediação personalizada para chamar a lógica de personalização.

O objectivo de um módulo de mediação que não contém um componente de fluxo é o de converter pedidos de serviços de um protocolo para outro. Por exemplo, pode ser feito um pedido de serviço utilizando SOAP/JMS mas pode necessitar de conversão para SOAP/HTTP antes de ser enviado.

**Nota:** Certas alterações podem ser vistas e executadas a partir do IBM Business Process Manager. Contudo, não podem ser vistas nem executadas alterações aos componentes SCA dentro de um módulo a partir do IBM Business Process Manager. O Integration Designer é utilizado para personalizar componentes SCA.

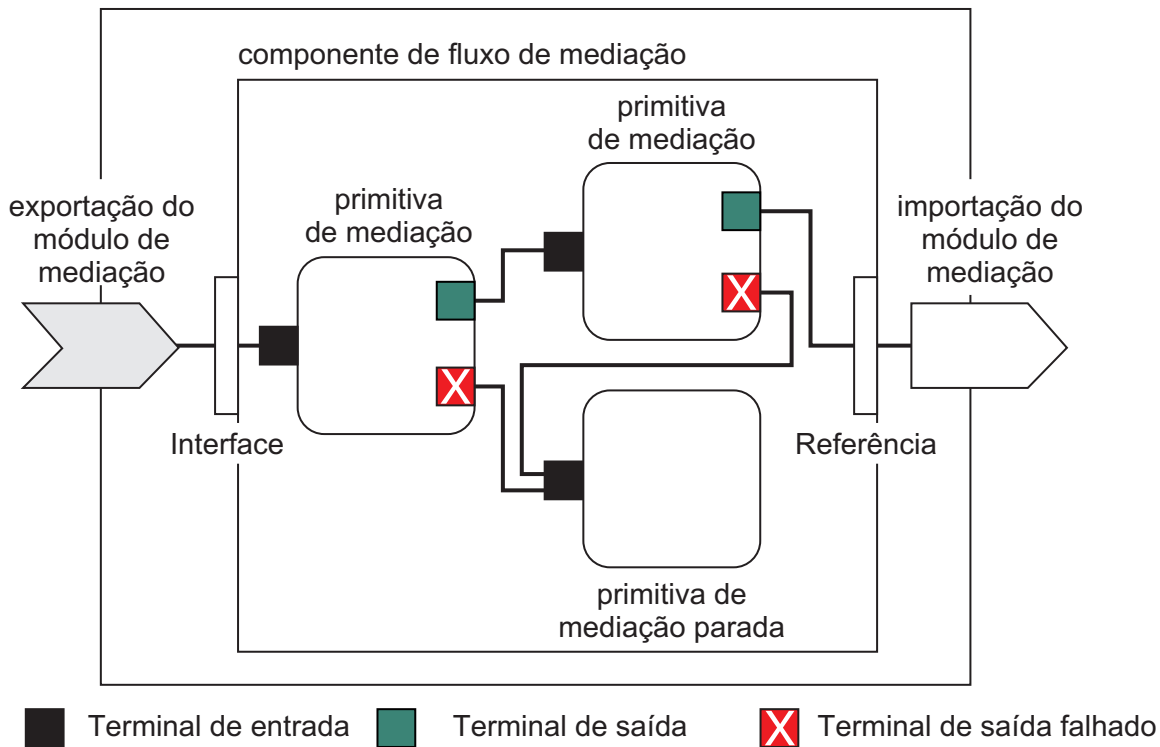


Figura 88. Exemplo simplificado de um módulo de mediação. O módulo de mediação contém um componente de fluxo de mediação, que contém primitivas de mediação.

- Propriedades

As primitivas de mediação têm propriedades, algumas das quais podem ser apresentadas na consola administrativa como propriedades adicionais de um módulo SCA.

Para que as propriedades das primitivas sejam visíveis a partir da consola administrativa do IBM Business Process Manager, o programador de integração deve promover as propriedades. Certas propriedades deixam-se ser configuradas administrativamente e o Integration Designer descreve-as como propriedades passíveis de promoção, porque podem ser promovidas do ciclo de integração para o ciclo administrativo. Outras propriedades não são adequadas para a configuração administrativa, porque modificá-las pode afectar o fluxo de mediação de uma forma que o módulo de mediação necessite de ser novamente implementado. O Integration Designer lista as propriedades seleccionáveis para promover de entre as propriedades promovidas de uma primitiva de mediação.

A consola administrativa do IBM Business Process Manager pode ser utilizada para alterar o valor de propriedades promovidas sem ser necessário implementar novamente um módulo de mediação ou reiniciar o servidor ou módulo.

Na generalidade, os fluxos de mediação utilizam imediatamente as alterações de propriedades. Contudo, se ocorrerem alterações de propriedades numa célula de gestão de implementação, estas tomam efeito em cada nó, enquanto este é sincronizado. Para além disso, os fluxos de mediação que estão em curso continuam a utilizar valores anteriores.

**Nota:** Da consola administrativa só é possível alterar os valores de propriedade e não os grupos, nomes ou tipos de propriedades. Se for desejável alterar os grupos, nomes ou tipos de propriedades, é necessária a utilização do Integration Designer.

- Um módulo de mediação ou biblioteca dependente também podem definir sub-fluxos. Um sub-fluxo encapsula um conjunto de primitivas de mediação ligadas como uma peça reutilizável de lógica de integração. Uma primitiva pode ser adicionada a um fluxo de mediação para chamar um sub-fluxo.

## Implementar módulos de mediação

Os módulos de mediação são criados utilizando o Integration Designer, e são, geralmente, implementados no IBM Business Process Manager, dentro de um ficheiro de arquivo de empresa (enterprise archive, EAR).

O valor de propriedades promovidas pode ser alterado na altura de implementação.

Pode ser exportado um módulo de mediação do Integration Designer, causando com que o Integration Designer armazene o módulo de mediação dentro de um ficheiro de arquivo Java (Java archive) e o ficheiro JAR dentro de um ficheiro EAR. O ficheiro EAR pode ser implementado com a instalação de uma nova aplicação a partir da consola administrativa.

Os módulos de mediação podem ser visto como uma entidade. Contudo, os módulos SCA são definidos por um número de ficheiros XML armazenados num ficheiro JAR.

### Exemplo de ficheiro EAR, que contém um módulo de mediação

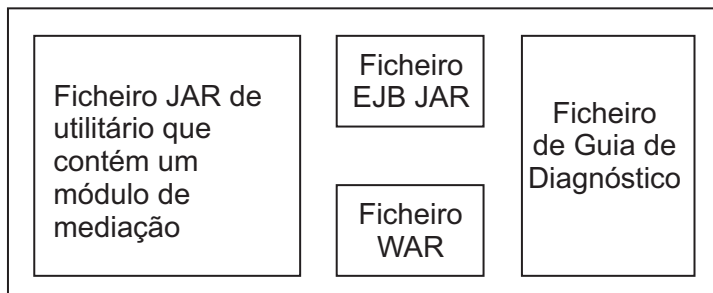


Figura 89. Um exemplo simplificado de um ficheiro EAR que contém um módulo de mediação. O ficheiro EAR contém JARs. O ficheiro JAR utilitário contém um módulo de mediação.

## Primitivas de mediação

Os componentes de fluxo de mediação operam em fluxos de mensagens entre componentes de serviços. As funções de um componente de mediação são implementadas por *primitivas de mediação*, que implementam tipos de implementação padrão.

Um componente de fluxo de mediação tem um ou mais fluxos. Por exemplo, um para pedidos e um para respostas.

O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação, que implementam funções de mediação padrão para módulos de mediação ou módulos implementados no IBM Business Process Manager. Se necessitar de funções de mediação especiais, pode desenvolver as suas próprias primitivas de mediação personalizadas.

Uma primitiva de mediação define uma operação “in” que processa ou conduz mensagens que são representadas por objectos de serviços de mensagens (SMOs). Uma primitiva de mediação pode também definir operações “out” que enviam mensagens para outro componente ou módulo.

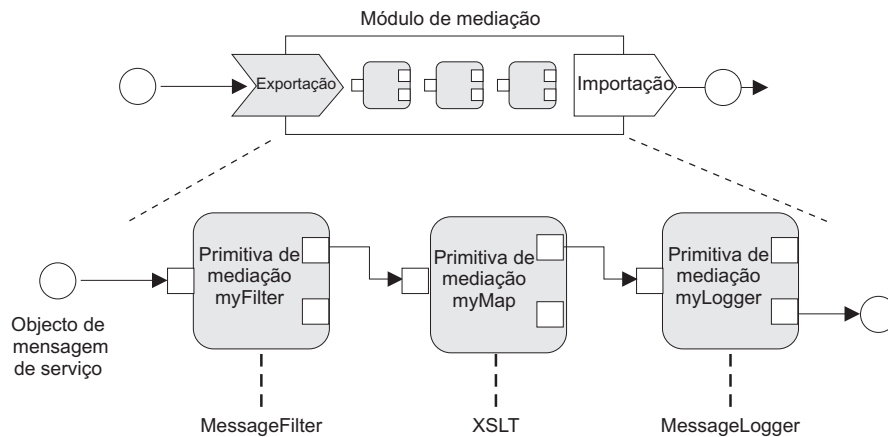


Figura 90. Módulo de mediação que contém três primitivas de mediação

Pode utilizar o Integration Designer para configurar as primitivas de mediação e definir as suas propriedades. Algumas destas propriedades podem ser tornadas visíveis ao administrador de tempo de execução ao promovê-las. Qualquer propriedade primitiva de mediação que possa ser promovida, pode também ser uma propriedade dinâmica. Uma propriedade dinâmica pode ser substituída, no tempo de execução, utilizando um ficheiro de política.

O Integration Designer também lhe permite modelar e montar graficamente componentes de fluxo de mediação a partir das primitivas de mediação e módulos de montagem ou módulos dos componentes de fluxo de mediação. A consola administrativa refere-se a módulos de mediação e módulos como os módulos SCA.

O Integration Designer também permite a definição dos sub-fluxos em módulos ou nas suas bibliotecas independentes. Um sub-fluxo pode conter qualquer primitiva de mediação, à excepção da primitiva de mediação de Resolução de Política. Um sub-fluxo é chamado um fluxo de pedido ou resposta, ou de outro sub-fluxo, utilizando a primitiva de mediação de Sub-fluxos. As propriedades promovidas das primitivas de mediação num sub-fluxo são expostas como propriedades nas primitivas de mediação de Sub-fluxos. Estes podem então ser promovidos outra vez até chegarem ao nível de módulo, altura em que podem ser modificados pelo administrador de tempo de execução.

## Primitivas de mediação suportadas

O seguinte conjunto de primitivas de mediação é suportado pelo IBM Business Process Manager:

### Mapa de objecto de negócio

Converte mensagens.

- Define conversões de mensagens utilizando um mapa de objecto de negócio, que pode ser re-utilizado.
- Permite-lhe definir graficamente conversões de mensagens, utilizando o editor de mapa de objecto de negócio.
- Pode alterar o conteúdo de uma mensagem.
- Pode converter um tipo de mensagem de entrada num tipo diferente de mensagem de saída.

### **Mediação Personalizada**

Permite-lhe implementar a sua lógica de mediação em código Java. A primitiva de Mediação Personalizada combina a flexibilidade de uma primitiva de mediação definido por um utilizador com a simplicidade de uma primitiva de mediação pré-definida. Pode criar conversões complexas e padrões de encaminhamento ao:

- Criar código Java.
- Criar as suas próprias propriedades.
- Adicionar novos terminais.

Pode chamar um serviço a partir de uma primitiva de Mediação Personalizada, mas a primitiva de mediação de Invocação de Serviços é concebida para chamar serviços e fornece funcionalidade adicional, como a de tentar novamente.

### **Rotina de Tratamento de Dados**

Permite converter uma parte de uma mensagem. É utilizado para converter um elemento de uma mensagem de um formato físico para uma estrutura lógica ou de um a estrutura lógica para um formato físico. A utilização primária da primitiva é a de converter um formato físico, como uma cadeia de texto dentro de um objecto de Mensagem de Texto JMS, para uma estrutura lógica de um Objecto de Negócio e vice-versa. Esta mediação é normalmente utilizada para:

- Transforme a secção da mensagem de entrada de uma estrutura definida para outra - um exemplo disto seria em que SMO inclui um valor de cadeia que é delimitado por vírgulas e que pretende analisar num objecto de negócios específico.
- Alterar o tipo de mensagem - um exemplo seria quando uma exportação JMS fosse configurada para utilizar uma associação básica de dados JMS e, dentro do módulo de mediação, o programador de integração decidisse que o conteúdo devesse ser inflacionado para uma estrutura BO específica.

### **Procura da Base de Dados**

Modifica mensagens, utilizando informação de uma base de dados fornecida pelo utilizador.

- Deve definir uma base de dados, uma origem de dados e quaisquer definições de autenticação do servidor para serem utilizados pela primitiva de mediação Procura de Base de Dados. Utilize a consola administrativa para o ajudar a fazê-lo.
- A primitiva de mediação Procura da Base de Dados pode ler a partir de apenas uma tabela.
- A coluna chave especificada deve conter um valor único.
- Os dados nas colunas de valores devem ser de um tipo de esquema XML simples ou de um tipo de esquema XML que expande um tipo de esquema XML simples.

### **Procura do Ponto de Terminação**

Permite o encaminhamento dinâmico de pedidos, ao procurar pontos de terminação de serviços no repositório.

- A informação de pontos de terminação de serviços é obtida de um WebSphere Service Registry and Repository (WSRR). O registo WSRR pode ser local ou remoto.
- Podem ser efectuadas alterações do registo a partir da consola administrativa do WSRR.
- O IBM Business Process Manager necessita de saber que registo utilizar e, por isso, devem ser criadas definições de acesso ao WSRR, utilizando a consola administrativa do IBM Business Process Manager.

### **Emissor de Eventos**

Melhora a supervisão ao deixá-lo enviar eventos de dentro de um componente de fluxo de mediação.

- A acção de mediação pode ser suspensa ao desmarcar a caixa de verificação.
- Pode ver eventos do Emissor de Eventos utilizando o navegador de Eventos de Base Comum no IBM Business Process Manager.
- Por razões de desempenho, só deverá conseguir enviar eventos numa altura significativa de um fluxo de mediação.

- É possível definir as partes da mensagem que o evento contém.
- Os eventos são enviados na forma de Eventos de Base Comum para o servidor da Infra-Estrutura de Eventos Comuns.
- Para utilizar completamente a informação do Emissor de Eventos, os consumidores de eventos necessitam de compreender a estrutura dos Eventos de Base Comum. Os Eventos de Base comum têm um esquema geral, mas isto não modela os dados específicos da aplicação, que é contida nos elementos de dados expandidos. Para modelar os elementos de dados expandidos, as ferramentas do Integration Designer geram um ficheiro de definição de catálogo de eventos de Infra-Estrutura de Eventos Comuns para cada uma das primitivas de mediação do Emissor de Eventos. Os ficheiros de definição de catálogo de eventos são artefactos de exportação fornecidos para ajudar; não são utilizados pelo Integration Designer ou pelo tempo de execução do IBM Business Process Manager. Os ficheiros de definição de catálogo de evento devem ser consultados quando são criadas aplicações para consumir os eventos de Emissão de Eventos.
- Pode ser especificada outra supervisão a partir do IBM Business Process Manager. Por exemplo, podem ser supervisionados eventos que serão emitidos das importações e exportações.

**Falha** Pára um caminho em particular no fluxo e gera uma excepção.

### **Concentrar**

Ajuda a agregar (combinar) mensagens.

- Pode apenas ser utilizado em conjunto com a primitiva de mediação Dispersar.
- Juntas, as primitivas de mediação Dispersar e Concentrar permitem a agregação de dados numa mensagem de saída.
- A primitiva de mediação Concentrar recebe mensagens até ser atingido um ponto de decisão, em que uma mensagem se torna de saída.
- O contexto partilhado deve ser utilizado para reter os dados de agregação.

### **Dispersar**

Ajuda a dividir e agregar (combinar) mensagens.

- Juntas, as primitivas de mediação Dispersar e Concentrar permitem a agregação de dados numa mensagem de saída.
- No modo iterativo, a primitiva de mediação Dispersar permite iterar uma única mensagem de entrada que contém um elemento repetido. Para cada ocorrência do elemento repetido é enviada uma mensagem.
- O contexto partilhado deve ser utilizado para reter os dados de agregação.

### **Compositor de Cabeçalhos HTTP**

Fornece um mecanismo para gerir cabeçalhos de mensagens HTTP.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens HTTP.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos HTTP.

### **Mapeamento**

Converte mensagens.

- Permite-lhe executar transformações Extensible Stylesheet Language (XSL) (linguagem de folha de estilos extensível) ou transformações Business Object Map (mapeamento do objecto de negócios).
- Se transformar mensagens ao utilizar uma transformação XSLT 1.0 ou XSLT 2.0 ou uma transformação Business Object Map (mapa de objecto de negócios). As transformações XSL funcionam numa serialização XML da mensagem, em que a transformação de Business Object Map funciona nos Service Data Objects (SDO) (objectos de dados de serviços).

### **Compositor de Elementos de Mensagem**

Fornece um mecanismo simples para definir o conteúdo de mensagens.

- Pode alterar, adicionar ou apagar elementos de mensagens.
- Não altera o tipo de mensagem.
- Os dados nas colunas de valores devem ser de um tipo de esquema XML simples ou de um tipo de esquema XML que expande um tipo de esquema XML simples.

### **Filtro de Mensagens**

Encaminha mensagens por caminhos diferentes, com base no conteúdo das mensagens.

- A ação de mediação pode ser suspensa ao desmarcar a caixa de verificação.

### **Registador de Mensagens**

Regista as mensagens numa base de dados relacional ou através do próprio registador de mensagens personalizado. As mensagens são guardadas como XML e, por isso, os dados podem ser depois processados por aplicações que detectem XML.

- A ação de mediação pode ser suspensa ao desmarcar a caixa de verificação.
- O esquema de base de dados relacional (estrutura de tabela) é definido pela IBM.
- Por defeito, a primitiva de mediação Registador de Mensagens utiliza a base de dados Comum. O ambiente em tempo de execução correlaciona a origem de dados em **jdbc/mediation/messageLog** com a base de dados Comum.
- Podem ser definidas classes de implementação de Processador para personalizar o comportamento do registador personalizado. Opcionalmente, podem ser fornecidas classes de implementação de Formatador, classes de implementação de Filtro ou ambas, para personalizar o comportamento do registador personalizado.

### **Compositor de Cabeçalhos MQ**

Fornece um mecanismo para gerir cabeçalhos em mensagens MQ.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens MQ.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos MQ.

### **Resolução de Política**

Permite a configuração dinâmica de pedidos, ao procurar, num repositório, pontos de terminação de serviços e ficheiros de política associados.

- Pode ser utilizado um ficheiro de política para substituir de forma dinâmica as propriedades promovidas de outras primitivas de mediação.
- A informação de pontos de terminação de serviços e informação de política são obtidas de um WebSphere Service Registry and Repository (WSRR). O registo WSRR pode ser local ou remoto.
- Podem ser efectuadas alterações do registo a partir da consola administrativa do WSRR.
- O IBM Business Process Manager necessita de saber que registo utilizar e, por isso, devem ser criadas definições de acesso ao WSRR, utilizando a consola administrativa do IBM Business Process Manager.

### **Chamada de Serviços**

Chama um serviço dentro de um fluxo de mediação, em vez de esperar até ao fim do fluxo de mediação e utilizar o mecanismo de chamada.

- Se o serviço devolver uma falha, pode ser feita uma nova tentativa ou chamar outro serviço.
- A primitiva de mediação de Chamada de Serviços pode ser utilizada sozinha para chamadas simples de serviço ou em conjunto com outras primitivas de mediação para mediações complexas.

### **Definir Tipo de Mensagem**

Durante o desenvolvimento da integração, permite tratar campos de mensagem pouco tipificados como se fossem muito tipificados. Um campo é pouco tipificado se puder conter mais de um tipo de dados. Um campo é muito tipificado se o seu tipo e estrutura interna forem conhecidos.

- No tempo de execução, a primitiva de mediação Definir Tipo de Mensagens permite verificar se o conteúdo de uma mensagem corresponde aos tipos de dados esperados.



## Compositor de Cabeçalhos SOAP

Fornecer um mecanismo para gerir cabeçalhos de mensagens SOAP.

- Pode criar, definir, copiar ou apagar cabeçalhos de mensagens SOAP.
- Pode definir acções múltiplas para alterar múltiplos cabeçalhos SOAP.

**Parar** Pára um caminho específico do fluxo sem gerar uma excepção.

## Filtro de Tipos

Permite direccionar mensagens num caminho diferente de um fluxo, com base no seu tipo.

## Obter WebSphere eXtreme Scale

Pode obter informações de um ambiente de memória cache do servidor eXtreme Scale.

- Pode procurar valores na memória cache e armazená-los como elementos na mensagem através de uma mensagem.
- Ao combinar as primitivas de mediação de armazenamento e de obtenção eXtreme Scale, pode colocar na memória cache a resposta de um sistema emissor. Os pedidos futuros não irão requerer acesso a esse sistema emissor.
- Deverá criar as definições eXtreme Scale ao utilizar a consola administrativa WebSphere ESB, pelo que pode especificar o servidor eXtreme Scale a utilizar.

## Armazém WebSphere eXtreme Scale

Pode armazenar informações num ambiente de memória cache do servidor eXtreme Scale.

- Pode armazenar informação numa memória cache eXtreme Scale ao utilizar uma chave e um objecto.
- Ao combinar as primitivas de mediação de armazenamento e de obtenção eXtreme Scale, pode utilizar a primitiva de mediação de armazenamento para armazenar dados na memória cache e utilizar a primitiva de mediação de obtenção para obter dados anteriormente gravados na memória cache.
- Deverá criar as definições eXtreme Scale ao utilizar a consola administrativa WebSphere ESB, pelo que pode especificar o servidor eXtreme Scale a utilizar.

## Encaminhamento dinâmico

É possível encaminhar mensagens de várias formas utilizando pontos de terminação definidos no tempo de integração ou pontos de terminação determinados, de forma dinâmica, no tempo de execução.

O encaminhamento dinâmico cobre dois casos de encaminhamento de mensagens:

- O encaminhamento de mensagens em que o fluxo é dinâmico, mas todos os pontos finais possíveis são predefinidos num nóduo Service Component Architecture (SCA)(arquitectura de componente de serviço).
- O encaminhamento de mensagens em que o fluxo é dinâmico e a selecção de pontos finais também é dinâmica. Os pontos finais do serviço são seleccionados em tempo de execução, a partir de uma fonte externa

## Seleção de ponto de terminação dinâmico

O tempo de execução tem a função de encaminhar mensagens de pedido e resposta para um endereço de ponto de terminação identificado por um elemento de cabeçalho de mensagem. Este elemento de cabeçalho de mensagem pode ser actualizado por primitivas de mediação, num fluxo de mediação. O endereço de ponto de terminação pode ser actualizado com informação de um registo, uma base de dados ou da própria mensagem. O encaminhamento de mensagens de resposta aplica-se apenas quando a resposta está a ser enviada por uma exportação de serviço da web WAX-WS.

Para que o tempo de execução implemente um encaminhamento dinâmico num pedido ou numa resposta, o módulo SCA deve ter o conjunto de propriedades Use dynamic endpoint if set in the message header (Utilizar pontos de terminação dinâmico se definidos no cabeçalho de mensagens). Os programadores de integração podem definir a propriedade Utilizar o ponto de terminação dinâmico

se for definido no cabeçalho da mensagem ou podem promovê-la (torná-la visível no tempo de execução), para que o administrador de tempo de execução a possa definir. As propriedades de módulos podem ser vistas na janela Module Properties (Propriedades do Módulo). Para ver a janela, é necessário fazer clique em **Applications (Aplicações) > SCA Modules (Módulos SCA) > Module Properties (Propriedades de Módulos)**. O programador de integração atribui nomes alternativos às propriedades promovidas e estes nomes são apresentados na consola administrativa.

## **Registo**

O IBM WebSphere Service Registry and Repository (WSRR) pode ser utilizado para armazenar informação de ponto de terminação e depois criar módulos SCA para obter pontos de terminação do registo do WSRR.

Quando se desenvolvem módulos SCA, é utilizado a primitiva de mediação Pesquisa de Pontos de Terminação para permitir que um fluxo de mediação consulte um registo WSRR sobre um ponto de terminação de serviço ou um conjunto de pontos de terminação de serviço. Se um módulo SCA obtiver um conjunto de pontos de terminação, este deve utilizar outra primitiva de mediação para seleccionar o preferido.

## **Controlo da política de mediação dos pedidos de serviço**

É possível utilizar as políticas de mediação para controlar fluxos de mediação entre solicitações de serviços e fornecedores de serviços.

É possível controlar fluxos de mediação utilizando políticas de mediação guardadas no IBM WebSphere Serviço de Registo e Repositório (WSRR, Service Registry and Repository). A implementação da gestão da política de serviço do WSRR é baseada no Web Services Policy Framework (Política WS).

Para controlar os pedidos de serviço utilizando políticas de mediação, é necessário possuir módulos de Arquitectura de Componente de Serviço (SCA, Service Component Architecture) e documentos de política de mediação no registo WSRR.

## **Procedimento para anexar uma política de mediação a um pedido de serviço**

Quando é desenvolvido um módulo SCA que necessite de utilizar uma política de mediação, deve ser incluído uma primitiva de mediação Resolução de Política (Policy Resolution) no fluxo de mediação. Durante o tempo de execução, a mediação da Resolução Política (Policy Resolution) primitiva, obtém as informações da política de mediação a partir do registo. Assim, um módulo SCA deve conter um componente de fluxo de mediação para suportar o controlo da política de mediação dos pedidos de serviço.

No registo, é possível anexar uma ou mais políticas de mediação a um módulo SCA ou a um serviço destino utilizado pelo módulo SCA. As políticas de mediação anexadas podem ser utilizadas (dentro do âmbito) para todas as mensagens de serviço processadas por esse módulo SCA. As políticas de mediação podem conter anexos de políticas para definir condições. As condições de políticas de mediação permitem a aplicação de diferentes políticas de mediação dependendo em diferentes contextos. Além disso, as políticas de mediação podem ter classificações, que podem ser utilizadas para especificar o estado de controlo.

## **Registo e Repositório de Serviços WebSphere**

O produto de Registo e Repositório de Serviços WebSphere (WSRR) permite armazenar, aceder e gerir informação acerca dos pontos de terminação de serviços e políticas de mediação. O WSRR pode ser utilizado para tornar as aplicações de serviço mais dinâmicas e mais adaptáveis a condições de negócios em constante alteração.

## Introdução

Os fluxos de mediação podem utilizar o WSRR como um mecanismo de procura dinâmica, fornecendo informação acerca dos pontos de terminação de serviço ou políticas de mediação.

Para configurar o acesso ao WSRR, criam-se documentos de definição do WSRR utilizando a consola administrativa. Alternativamente, os comandos de administração do WSRR podem ser utilizados a partir do cliente de script wsadmin. As definições do WSRR e as suas propriedades de ligação são o mecanismo utilizado para ligar a uma instância de registo e obter um ponto de terminação de serviço ou política de mediação.

## Pontos de Terminação de serviços

O WSRR pode ser utilizado para armazenar informação acerca de serviços já utilizados, que poderão ser utilizados ou que sejam de interesse. Estes serviços podem estar em sistemas próprios ou noutros sistemas. Por exemplo, uma aplicação pode utilizar o WSRR para localizar o serviço mais apropriado para satisfazer as suas necessidades funcionais e de desempenho.

Quando é desenvolvido um módulo SCA que necessita de aceder pontos de terminação de serviços a partir do WSRR, é necessário incluir a primitiva de mediação Procurar Pontos de Terminação no fluxo de mediação. No tempo de execução, a primitiva de mediação Pesquisa de Pontos de Terminação obtém pontos de terminação de serviços a partir do registo.

## Políticas de mediação

O WSRR também pode ser utilizado para armazenar informação de política de mediação. As políticas de mediação podem ajudar a controlar pedidos de serviços, ao substituir propriedades de módulos de forma dinâmica. Se o WSRR contiver políticas de mediação que estão ligadas a um objecto que representa o módulo SCA ou o serviço destino, então as políticas de mediação podem substituir as propriedades de módulos. Se for desejado que diferentes políticas de mediação se apliquem a contextos diferentes, é possível criar condições de política de mediação.

**Nota:** As políticas de mediação dizem respeito ao controlo de fluxos de mediação e não à segurança.

Quando é desenvolvido um módulo SCA que necessite de utilizar uma política de mediação, deve ser incluído uma primitiva de mediação Resolução de Política (Policy Resolution) no fluxo de mediação. Durante o tempo de execução, a mediação da Resolução Política (Policy Resolution) primitiva, obtém as informações da política de mediação a partir do registo.

## WebSphere eXtreme Scale

Ao utilizar o produto WebSphere eXtreme Scale (eXtreme Scale) pode disponibilizar um sistema de memória cache que pode integrar com uma aplicação IBM Business Process Manager. Ao utilizar eXtreme Scale com IBM Business Process Manager pode melhorar os tempos de resposta e a fiabilidade do serviço e disponibilizar uma função de integração adicional.

eXtreme Scale age como uma grelha de dados na memória, elástica e escalável. A grelha de dados coloca na memória cache, particiona, duplica e gere dados de aplicação e lógica de negócios em vários servidores. Com eXtreme Scale, também pode obter qualidades de serviço como a integridade transaccional, a elevada disponibilidade e tempos de resposta previsíveis.

Pode utilizar os fluxos de mediação para aceder à função de memória cache eXtreme Scale ao incluir as primitivas de mediação WebSphere eXtreme Scale no seu fluxo. Quando desenvolve um módulo de arquitectura de componente de serviço (SCA) que necessita de armazenar informação numa memória cache eXtreme Scale, deverá incluir a primitiva de mediação de armazenamento WebSphere eXtreme Scale no fluxo de mediação. Se pretende recuperar informação de uma memória cache eXtreme Scale, deverá incluir a primitiva de mediação de recuperação WebSphere eXtreme Scale. Ao combinar duas

primitivas de mediação num fluxo de mediação, pode obter a resposta de memória cache de um sistema de sistema emissor, para que os pedidos futuros possam obter a resposta da memória cache.

Para configurar o acesso a eXtreme Scale, deverá criar uma definição WebSphere eXtreme Scale através da consola administrativa. Como alternativa, pode utilizar os comandos administrativos WebSphere eXtreme Scale do cliente de script wsadmin. Uma definição eXtreme Scale é um mecanismo utilizado pelas primitivas de mediação de obtenção e de armazenamento WebSphere eXtreme Scale para ligar a um servidor eXtreme Scale.

## Clientes do Message Service

Os clientes do Message Service estão disponíveis para C/C++ e .NET de forma a permitir que aplicações que não sejam Java se liguem ao Enterprise Service Bus.

O Message Service Clients for C/C++ and .NET fornece um API chamado XMS que tem o mesmo conjunto de interfaces que o API de Serviço de Mensagens Java (JMS). O Message Service Client for C/C++ contém duas implementações de XMS, uma para utilização por aplicações C e outra para utilização por aplicações C++. O Message Service Client for .NET contém uma implementação totalmente gerida de XMS, que pode ser utilizada por qualquer linguagem utilizada compatível com .NET.

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

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

Pode também instalar e utilizar o suporte de cliente Java EE de WebSphere Application Server Network Deployment, incluindo o cliente de serviços da web, o cliente EJB e o cliente JMS.

