

IBM WebSphere Commerce



Loja de Exemplo NewFashion: Arquivos de Ajuda Online

Versão 54

IBM WebSphere Commerce



Loja de Exemplo NewFashion: Arquivos de Ajuda Online

Versão 54

Nota!

Antes de utilizar estas informações e o produto suportado por elas, leia as informações gerais em Avisos.

Índice

Capítulo 1. Criando uma Loja Utilizando um Exemplo 1

Criando uma Loja Utilizando um Exemplo	1
Criando um Archive de Loja de Exemplo	1
Alterando Recursos do Banco de Dados da Loja	3
Alterando Recursos da Web	17
Publicando um Archive de Loja de Exemplo a Partir de Store Services	17
Publicando um Archive de Loja Utilizando a Linha de Comandos (Windows)	19
Configurando o Payment Manager para sua Loja	21
Criando Jobs Programados para uma Loja de Exemplo	24
Configurando Notificação de E-mail para uma Loja de Exemplo	26

Capítulo 2. Recursos do Banco de Dados da Loja de Exemplo 27

Recursos do banco de dados da loja	27
Recursos do Banco de Dados do Catálogo da NewFashion	28
Recursos do Banco de Dados da Política de Negócios e de Contrato da NewFashion	30
Recursos do Banco de Dados de Envio da NewFashion	31
Recursos de banco de dados de controle de acesso da loja de exemplo	33

Capítulo 3. Loja de Exemplo NewFashion 35

Loja de exemplo NewFashion	35
--------------------------------------	----

Capítulo 4. Páginas da Loja NewFashion 37

Páginas de pesquisa da loja de exemplo NewFashion	37
Páginas do Catálogo de Endereços da Loja de Exemplo	39
Páginas do catálogo de endereços da loja de exemplo NewFashion	40
Páginas de exibição de pacote Configurável das lojas de exemplo WebFashion e NewFashion	43
Páginas do grupo de catálogos da Loja de Exemplo	44
Páginas da loja de exemplo: técnicas comuns de implementação	46
Página Entre em Contato Conosco da Loja de Exemplo	48
Rodapé da Loja de Exemplo	49
Cabeçalho da Loja de Exemplo	50
Página de Ajuda da Loja de Exemplo	52
Home page da loja de exemplo	52
Páginas de Login da Loja de Exemplo	53
Página de privacidade da loja de exemplo	56
Páginas de Produtos da Loja de Exemplo	56

Página de registro da loja de exemplo	58
Página de Novidades da Loja de Exemplo	60
Página de confirmação de pedido da loja de exemplo NewFashion	61
Página Notificação de E-mail da WebFashion e da NewFashion	62
Página de Resumo do Pedido da Loja de Exemplo	63
Página Verificar Disponibilidade do Produto da Loja de Exemplo NewFashion	64
Páginas de exibição de pacote das lojas de exemplo WebFashion e NewFashion	65
Quadro de Navegação Esquerdo da Loja de Exemplo	66
Quadro de navegação esquerdo da loja de exemplo NewFashion	67
Página Selecionar Endereço para Cobrança da Loja de Exemplo	70
Página Selecionar Endereço para Cobrança da Loja de Exemplo NewFashion	71
Carrinho de Compras da Loja de Exemplo	72
Carrinho de compras da loja de exemplo NewFashion	74
Página Método de Envio da Loja de Exemplo	76
Página Selecionar Método de Envio da Loja de Exemplo NewFashion	77
Página Selecionar Endereço de envio da Loja de Exemplo	78
Página Selecionar Endereço de Envio da Loja de Exemplo NewFashion	79
Página de resumo de compra rápida da loja de exemplo	80
Páginas de exibição de pedidos das lojas de exemplo WebFashion e NewFashion	82
Páginas da Lista de Itens Desejados das Lojas de Exemplo WebFashion e NewFashion	84
Casos de utilização da loja de exemplo	86

Capítulo 5. Casos de Utilização da Loja de Exemplo. 89

Caso de utilização Incluir Novo Endereço (Business Edition)	89
Caso de utilização Incluir item na lista de itens de interesse da WebFashion e da NewFashion	90
Caso de utilização Exibir categoria de produto	90
Caso de Utilização Registrar Saída do Carrinho de Compras	91
Caso de utilização da página Exibir Pacote Configurável da WebFashion e NewFashion	94
Caso de utilização da página Exibir Pacote da WebFashion e NewFashion	95
Caso de utilização Editar um Endereço (Business Edition)	96
Caso de utilização Home page	97
Caso de utilização de Logon	98
Caso de utilização Gerenciar Conta Pessoal	98

Caso de utilização Exibir Pedidos da WebFashion e NewFashion	100
Caso de utilização Exibir página de produtos.	100
Caso de utilização Alterar informações pessoais	101
Caso de utilização do registro	102
Caso de utilização Exibir carrinho de compras	103
Caso de utilização da página Exibir Lista de Itens de Interesse da WebFashion e NewFashion	104

Fluxo do Carrinho de Compras da Loja de Exemplo NewFashion	107
Fluxo do Carrinho de Compras da Loja de Exemplo NewFashion	107


Avisos 109

Capítulo 6. Fluxo de Compra NewFashion 107

Capítulo 1. Criando uma Loja Utilizando um Exemplo

Criando uma Loja Utilizando um Exemplo

Para criar uma loja utilizando uma das lojas de exemplo, faça o seguinte:

1. (Opcional)  Crie uma organização para atuar como vendedor. Para determinar se você deseja criar uma nova organização para agir como vendedora, consulte Recursos de dados compartilhados.
2. Crie usuários para as seguintes funções:
 - Administrador do Site (se você não estiver utilizando o Administrador do Site padrão)
 - Administrador de Vendedores
 - Administrador de Loja
 - Desenvolvedor de Loja

Importante: Para publicar um archive de loja, você deve ter acesso de Administrador do Site, Administrador de Loja ou Desenvolvedor de Loja para todas as lojas. Se você estiver criando um usuário com autoridade de Administrador de Loja, certifique-se de que o grupo de acesso aplique-se a todas as lojas.
3. Criar um archive de loja utilizando os Store Services
4. Altere recursos do banco de dados da loja.
5. Altere os recursos da Web.
6. Publique um archive de loja.
7. (Opcional) Configurando lojas.
8. Configure o Payment Manager para sua loja.

Importante:

1. Em algumas instâncias, as lojas de exemplo precisam de alguma configuração para que todos os recursos funcionem corretamente. Se você está baseando sua loja em uma loja de exemplo, pode precisar concluir a configuração. Para obter mais informações, consulte Configurando uma loja de exemplo.
2. Determinados recursos de dados da loja, como catálogos e centros de distribuição, são compartilhados entre as lojas. Como resultado, se você publicar várias lojas na mesma loja de exemplo, o catálogo e o centro de distribuição serão os mesmos para cada loja. Se você tiver feito alterações em um catálogo, publicar outra loja no mesmo exemplo irá sobrepor essas alterações. Para obter mais informações e saber como evitar a substituição das alterações, consulte Recursos de dados compartilhados.

Criando um Archive de Loja de Exemplo

Para criar um archive de loja que possa ser utilizado como exemplo com o Store Services, consulte o *IBM WebSphere - Manual do Desenvolvedor de Loja*.

Alterando Recursos do Banco de Dados da Loja

Se você criar o archive de loja utilizando as ferramentas do Store Services, seu novo archive de loja conterá inicialmente os mesmos recursos do banco de dados de loja que o archive de loja de exemplo no qual você o baseou, como `infashion.sar`. Em um archive de loja, os recursos do banco de dados da loja assumem a forma de arquivos XML.

Na maioria dos casos, para alterar os recursos do banco de dados da loja, você deve editar diretamente os arquivos XML. Em alguns casos, você pode utilizar as ferramentas do Store Services para editar os recursos do banco de dados.

Você também tem a opção de editar recursos do banco de dados diretamente, ou seja, depois de ter publicado o archive de loja no Commerce Server, você pode editar o banco de dados utilizando o WebSphere Commerce Accelerator, o Loader package ou diretamente através de inserções de SQL. Se você optar por editar o banco de dados em vez dos recursos no archive de loja, deve atualizar o archive de loja para corresponder às alterações no banco de dados ou interromper o uso do archive de loja.

Suas opções estão listadas na tabela a seguir.

Importante:

1. As ferramentas em Store Services pesquisam pelos nomes dos recursos listados na tabela abaixo. Como resultado, se desejar editar seu archive de loja utilizando as ferramentas nos Store Services, você deve utilizar os mesmos nomes de recursos em seu archive de loja.
2. Quando um archive de loja é publicado para o WebSphere Commerce Server, as informações do banco de dados são carregadas na ordem especificada na coluna Recursos abaixo. Assim, a ordem de seus recursos, conforme especificada no arquivo `sarinfo.xml`, deve corresponder à ordem dos recursos especificada abaixo.
3. A ordem das informações do banco de dados em cada arquivo XML não precisa necessariamente corresponder à ordem especificada na coluna de Tabelas de Banco de Dados a seguir. Entretanto, as informações para uma tabela pai devem preceder as informações para uma tabela filho.
4. As informações marcadas como opcionais não são requeridas para a criação de uma loja funcional.
5. Para a coluna Opções de Edição do Banco de Dados, observe que a menos que seja especificada outra coisa, todos os recursos do banco de dados podem ser editados utilizando inserções de SQL ou o Loader package. Assim, a coluna reflete quais recursos podem ser editados pelo WebSphere Commerce Accelerator.

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
atendimento	FFMCENTER (0...1)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Novas páginas Centro de Distribuição e Alterando Centro de Distribuição no WebSphere Commerce Accelerator. • Consulte Criando um centro de distribuição e Alterando um centro de distribuição
loja	STOREENT	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. • Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - IDENTIFIER - DIRECTORY - SETCURR 	
	STADDRESS	<ul style="list-style-type: none"> • Alterando informações da loja utilizando o formulário Perfil da Loja. 	<ul style="list-style-type: none"> • Novas páginas Centro de Distribuição e Alterando Centro de Distribuição no WebSphere Commerce Accelerator. • Consulte Criando um centro de distribuição e Alterando um centro de distribuição
	STOREENTDS	<ul style="list-style-type: none"> • Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - DESCRIPTION - DISPLAYNAME 	
	STORELANG	<ul style="list-style-type: none"> • Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - LANGUAGE_ID 	
	STORELANGDS	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
loja	STORE	<ul style="list-style-type: none"> Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> PHONE1 CITY STORE_ID COUNTRY STATE EMAIL1 ADDRESS1 ADDRESS2 FAX ZIPCODE 	
	DISPENTREL (modelo CATENTRY padrão, CATENTRY_ID = -1)	<ul style="list-style-type: none"> Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita a coluna do banco de dados PAGENAME. 	
	DISPCGPREL (modelo CATEGORY padrão, CATGROUP_ID = -1)	<ul style="list-style-type: none"> Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita a coluna do banco de dados PAGENAME. 	
	VENDOR	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
catálogo	CATGROUP	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATGRPATTR	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	CATGRPDESC	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATALOG	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	CATALOGDSC	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	CATTOGRP	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	CATGRPREL	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATENTRY	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATENTDESC	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	ATTRIBUTE	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	ATTRVALUE	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
catálogo	CATGPENREL	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATENTREL	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	BASEITEM	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	ITEMSPC	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	VERSIONSPC	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	DISTARRANG	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	RECEIPT	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	RCPTAVAIL	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	
	STOREITEM	<ul style="list-style-type: none"> Alterando informações do catálogo Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
imposto	JURSTGROUP (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – JURSTGROUP_ID – DESCRIPTION – SUBCLASS – STOREENT_ID – CODE 	
	JURST (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – JURST_ID – COUNTRY – STOREENT_ID – CODE – SUBCLASS – STATE 	
	JURSTGPREL (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – JURST_ID – JURSTGROUP_ID – SUBCLASS 	
	CALMETHOD (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	TXCDCLASS (opcional, somente para categorização de CALCODE, utilizado pelo WebSphere Commerce Accelerator)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
imposto	TAXCGRY	<ul style="list-style-type: none"> Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> TAXCGRY_ID STOREENT_ID NAME 	
	CALCODE (comum entre imposto e envio)	<ul style="list-style-type: none"> Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> CALCODE_ID CODE CALUSAGE_ID STOREENT_ID GROUPBY CALMETHOD_id CALMETHOD_id_app CALMETHOD_id_qfy 	
	CALCODEDSC (comum entre imposto e envio, opcional)	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
	CALRULE (comum entre imposto e envio)	<ul style="list-style-type: none"> Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> CALRULE_ID CALCODE_ID TAXCGRY_ID CALMETHOD_ID 	
	CALSCALE (comum entre imposto e envio)	<ul style="list-style-type: none"> Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> CALSCALE_ID CODE STOREENT_ID CALUSAGE_ID SETCURR CALMETHOD_ID 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
imposto	CALSCALEDS (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CALRANGE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – CALRANGE_ID – CALSCALE_ID – CALMETHOD_ID – RANGESTART 	
	CALRLOOKUP (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – CALSCALE_ID – CALRULE_ID 	
	CRULESCALE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – CALSCALE_ID – CALRULE_ID 	
	CALRULEMGP (opcional)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CALCODTXEX (isenção de imposto, utilizada por descontos)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	STENCALUSG (comum, CALCODE padrão para loja, 2 entradas: 1 para imposto sobre venda, 1 para imposto sobre envio)	<ul style="list-style-type: none"> • Alterando definições de imposto utilizando o formulário Imposto. 	
preenchimento de imposto	TAXJCRULE	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. • Alterando definições de imposto utilizando o formulário Imposto. O formulário Imposto edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – CALRULE_ID – FFMCENTER_ID – JURSTGROUP_ID 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
loja-catálogo-imposto	CATENCALCD (opcional)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATGPCALCD (opcional)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
envio	JURSTGROUP (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - JURSTGROUP_ID - DESCRIPTION - SUBCLASS - STOREENT_ID - CODE 	
	JURST (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - JURST_ID - COUNTRY - STOREENT_ID - CODE - SUBCLASS - STATE 	
	JURSTGPREL (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - JURST_ID - JURSTGROUP_ID - SUBCLASS 	
	SHIPMODE	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CODE - CARRIER - SHIPMODE_ID 	
	SHPMODEDSC	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - SHIPMODE_ID - LANGUAGE_ID 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
envio	CALMETHOD (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CALCODE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALCODE_ID - CODE - CALUSAGE_ID - STOREENT_ID - GROUPBY - CALMETHOD_ID 	
	CALCODEDSC (opcional)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. 	
	CALRULE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALRULE_ID - CALCODE_ID - TAXCGRY_ID - CALMETHOD_ID 	
	CALSCALE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALSCALE_ID - CODE - STOREENT_ID - CALUSAGE_ID - SETCURR - CALMETHOD_ID 	
	CALRULEMGP (opcional)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
envio	CALSCALEDS (comum entre imposto e envio)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CALRANGE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALRANGE_ID - CALSCALE_ID - CALMETHOD_ID 	
	CALRLOOKUP (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALRLOOKUP_ID - SETCURR - CALRANGE_ID - VALUE 	
	CRULESCALE (comum entre imposto e envio)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> - CALSCALE_ID - CALRULE_ID 	
	STENCALUSG (comum, calcode padrão para loja. 1 entrada para envio de caluage,1.)	<ul style="list-style-type: none"> • Alterando definições de envio utilizando o formulário Envio. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
preenchimento de envio	SHPJCRULE (pelo menos uma regra padrão para loja)	<ul style="list-style-type: none"> Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – CALRULE_ID – SHPARRANGE_ID – JURSTGROUP_ID 	
	SHPARRANGE	<ul style="list-style-type: none"> Alterando definições de envio utilizando o formulário Envio. O formulário Envio edita as seguintes colunas do banco de dados: <ul style="list-style-type: none"> – SHARRAND_ID – STORE_ID – FFMCENTER_ID – SHIPMODE_ID 	
loja-catálogo	STORECAT	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
	STORECENT	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
	STORECGRP	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
	DISPENTREL	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
	DISPCGPREL	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	
preenchimento da loja	INVENTORY	<ul style="list-style-type: none"> Editar arquivos XML de archive de loja. 	

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
oferta	TRADEPOSCN (1)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	MGTRDPSCN (opcional, para grupo de clientes)	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	OFFER	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	OFFERDESC	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	OFFERPRICE	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	LISTPRICE	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
comando	URLREG	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CMDREG	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	VIEWREG	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
moeda	CURCONVERT	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CURFORMAT	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CURFMTDESC	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CURCVLIST	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	CURLIST	<ul style="list-style-type: none"> • Alterando informações da loja utilizando o formulário Perfil da Loja. O formulário Perfil da Loja edita CURRSTR. 	
campanha	EMSPOT	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Ponto de e-Marketing no WebSphere Commerce Accelerator
	CAMPAIGN	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Campanhas no WebSphere Commerce Accelerator

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
loja-catálogo-envio	CATENCALCD	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	<ul style="list-style-type: none"> • Ferramentas de Gerenciamento de Produtos no WebSphere Commerce Accelerator
	CATGPCALCD	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
loja-padrões	STOREDEF	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
verificação de consistência		<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
pagamento	CMDREG	<ul style="list-style-type: none"> • Alterando definições de pagamento. • Editar arquivos XML de archive de loja. 	
	VIEWREG	<ul style="list-style-type: none"> • Alterando definições de pagamento. • Editar arquivos XML de archive de loja. 	
política	POLICY	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	POLICYCMD	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
Business organização	ORGENTITY	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	MBRREL	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	ADDRBOOK	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
	ADDRESS	<ul style="list-style-type: none"> • Editar arquivos XML de archive de loja. 	
Business conta de negócios	TERMCOND	Os recursos da conta não podem ser atualizados e republicados através do archive de loja. Se você precisa alterar os recursos de conta, deve editá-los no banco de dados utilizando as ferramentas de Gerenciamento de Relacionamentos de Negócios no WebSphere Commerce Accelerator.	Ferramentas de Gerenciamento de Relacionamentos de Negócios no WebSphere Commerce Accelerator Importante: Não é possível carregar contas utilizando o Loader Package.
	ACCOUNT		
	TRADING		
	TCDESC		
	PATTRVALUE		
	CREDITLINE		
	TRDESC		
	POLICYTC		
	BUYERPO		
	PARTICIPNT		
	ATTACHMENT		
TRDATTACH			

Recursos do Banco de Dados da Loja	Tabelas do Banco de Dados	Opções de Edição do Store Archive	Opções de Edição do Banco de Dados
<div style="background-color: #c00000; color: white; padding: 2px; display: inline-block;">Business</div> contrato	CONTRACT TERMCOND PRODUCTSET TRADING TCDESC PATTRVALUE TRDDESC POLICYTC PARTICIPNT TRADEPOSCN ATTACHMENT OFFER TRDATTACH OFFERPRICE STORECNTR PURCHASELT STOREDEF (utilizando pelo Store Services) APRVSTATUS FLINSTANCE	<ul style="list-style-type: none"> Editando arquivos XML de contrato no archive de loja 	Ferramentas de Gerenciamento de Relacionamentos de Negócios no WebSphere Commerce Accelerator Importante: Não é possível carregar contratos utilizando o Loader Package.

Alterando Recursos da Web

Para alterar os recursos da Web em seu archive de loja, existem as seguintes opções:






- Importar o archive de loja no WebSphere Studio e fazer as alterações necessárias nos recursos da Web, utilizando o Page Designer ou uma ferramenta de sua preferência. Se necessário, crie novas páginas da loja. Em seguida, exporta os recursos da Web de volta para o archive de loja ou para a loja em execução.
- Fazer download dos recursos da Web do archive de loja em uma localização de sua preferência, utilizando o diálogo Recurso da Web no Store Services e alterá-los, utilizando uma ferramenta de sua preferência. Ou substituir os recursos da Web no archive de loja por seus recursos existentes da Web, utilizando o diálogo Recurso da Web no Store Services. Se necessário, crie novas páginas de loja.
- Abrir manualmente os arquivo archive compactado de recursos da Web no archive de loja e fazer alterações nos arquivos existentes ou incluir novos arquivos.

Publicando um Archive de Loja de Exemplo a Partir de Store Services

Publicar o archive de uma loja de exemplo em um WebSphere Commerce Server permite criar uma loja funcional. Para publicar um archive de loja, conclua as seguintes etapas:

1. Certifique-se de possuir acesso de Administrador do Site ou Administrador de Loja. Se você tiver acesso de Administrador de Loja, verifique se o acesso é para todas as lojas.
2. Certifique-se de que tenha concluído as tarefas em Certificar-se de que os serviços e servidores apropriados estejam em execução.
3. Abra o Store Services.
4. Na lista **Archive de Loja**, selecione o archive de loja que deseja publicar.
5. Clique em **Publicar**.
É exibida a página Publicar Archive de Loja.
6. Selecione as opções de publicação desejadas. Para obter mais informações sobre opções de publicação, clique em **Ajuda**.
Dica: Para criar uma loja totalmente funcional, selecione todas as opções de publicação, incluindo a opção de dados de produtos, na primeira vez em que publicar um archive de loja.
7. Selecione **OK**.
Enquanto a loja publica você retorna para a página de listas do Archive de Loja. O estado da publicação é refletido na coluna Status da publicação. Clique em **Atualizar** para atualizar o status.
8. Selecione o archive de loja a partir da lista e clique em **Resumo da Publicação** para consultar os resultados da publicação.
9. Quando a publicação estiver concluída, clique em **Lançar Loja** para exibir e testar sua loja. Quando tiver terminado, marque o site e feche o navegador.

Importante:

1. Se alterar o caminho da Web do aplicativo da Web ou a raiz do documento dele, você deverá assegurar que eles correspondam aos caminhos definidos pelo WebSphere Commerce Server.
2. Pode ser publicado apenas um archive de loja por vez. A publicação simultânea não é suportada e faz com que a publicação das duas lojas falhe.
3. Durante a publicação, o verificador de consistência confirma se os arquivos mencionados pelo archive de loja existem. Se o verificador de consistência encontrar um erro, ele será gravado no log. A publicação continua normalmente.
4. Antes de publicar a loja novamente, exclua os arquivos do seguinte diretório:
 -  `NT unidade:\WebSphere\CommerceServer\instances\instancename\cache`
 -  `2000 unidade:\Arquivos de Programas\WebSphere\CommerceServer\instancename\cache`
 -  `AIX /usr/WebSphere/CommerceServer/instances/instancename/cache`
 -  `Solaris /opt/WebSphere/CommerceServer/instances/instancename/cache`
 -  `400 /QIBM/UserData/WebCommerce/instances/instancename/cache`
 Enquanto estiver na fase de desenvolvimento da loja, talvez você deseje desativar a cache. Para obter mais informações, consulte Configurar a cache.
5. Quando você inicia a loja no Serviços da Loja, você inicia sessão na loja com o mesmo nome e senha de usuário utilizados para iniciar sessão no Store Services. Se você alterar a senha na loja, ela também será alterada para o Store Services. Ao invés disso, para testar os recursos na loja, inclusive a troca da senha, salve o endereço do site, feche todas as janelas do navegador e efetue logon na loja novamente.
6. Se você tiver efetuado logon como administrador padrão, não poderá navegar em uma loja com base na loja de exemplo business-to-business. Em vez disso, crie um novo usuário pertencente à organização padrão e depois navegue pela loja.

7. Determinados recursos de dados da loja, como catálogos e centros de distribuição, são compartilhados entre as lojas. Como resultado, se você publicar várias lojas na mesma loja de exemplo, o catálogo e o centro de distribuição serão os mesmos para cada loja. Se você tiver feito alterações em um catálogo, publicar outra loja no mesmo exemplo irá sobrepor essas alterações. Para obter mais informações e saber como evitar a substituição das alterações, consulte Recursos de dados compartilhados.

Publicando um Archive de Loja Utilizando a Linha de Comandos (Windows)

Embora o método principal de se publicar um archive da loja seja pelo Store Archive Services, você também pode publicar um archive da loja utilizando a linha de comandos. Para publicar utilizando a linha de comandos, faça o seguinte:

1. Certifique-se de possuir acesso de Administrador do Site ou Administrador de Loja. Se você tiver acesso de Administrador de Loja, verifique se o acesso é para todas as lojas.
2. Edite o seguinte comando, utilizando parâmetros válidos para o archive de loja: `publishstore sarName hostname logonId logonPwd {insert|update} destination1=webapp.zip,destination2=properties.zip` em que:
 - `sarName` é o nome do archive de loja. `sarName` faz distinção entre maiúsculas e minúsculas. Certifique-se de utilizar a caixa correta.
 - `hostname` em que `hostname` é o nome TCP/IP completo do WebSphere Commerce Server e o número da porta de ferramentas da sua instância. Você pode localizar o número da porta de ferramentas no Gerenciador de Configuração, sob **Propriedades da instância > WebSphere**. Por padrão, este é o `hostname:8000`.
 - `logonId` é o ID do usuário do WebSphere Commerce.
 - `logonPwd` é a senha de logon do usuário do WebSphere Commerce
 - `insert|update` determina se a loja está sendo criada (`insert`) ou atualizada (`update`).
 - `{ALL|NOCATLG}` determina quais arquivos XML no SAR devem ser publicados. Para publicar todos, utilize `ALL`. Para publicar tudo exceto o catálogo, utilize `NOCATLG`.

- `destination1=webapp.zip,destination2=properties.zip`, é a lista dos arquivos de recursos de arquivo no SAR, por exemplo `webapp.zip`, e os caminhos nos quais eles serão publicados, por exemplo, `destination1is`

```
> NT unidade:\Websphere\AppServer\installedApps\
WC_Enterprise_App_instancename.ear\wcstores.war
> 2000 unidade:\Arquivos de
Programas\WebSphere\AppServer\installedApps\
WC_Enterprise_App_instancename.ear\wcstores.war
```

`destination2` é

```
> NT
unidade:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
wcstores.war\WEB-INF\classes
> 2000 unidade:\Arquivos de
Programas\WebSphere\AppServer\installedApps\
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes
```

A seguir é fornecido um exemplo do comando:

```
publishstore mysar.sar myhost wcsadmin wcsadmin insert ALL
```

```
"d:\websphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
wcstores.war=webapp.zip,d:\websphere\AppServer\installedApps\
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes=properties.zip"
```

3. Abra o Internet Explorer. Vá para o seguinte endereço da Web:
`http://hostname/webapp/wcs/stores/store_directory/index.jsp`, em que *store_directory* é o diretório da loja que você acabou de publicar.
É exibida a sua loja.

Nota: Se estiver publicando um archive de loja criado com o WebSphere Commerce Suite, versão 5.1, será necessário concluir as seguintes etapas antes de lançar a loja por meio da URL:

1. Se estiver utilizando o DB2, faça o seguinte. Se estiver utilizando o Oracle, vá para a etapa 2.
 - a. Quando o comando for executado, no menu **Iniciar**, selecione **Programas, DB2 para Windows NT e Janela de Comando**.
 - b. Na janela **DB2 CLP**, digite `db2 connect to dbname`, em que *dbname* é o banco de dados no qual você está publicando a loja. Pressione **Enter**.
 - c. Na linha de comandos, digite `db2 select * from store`. Pressione **Enter**. É exibida uma lista de lojas. Anote o número da loja que você criou.
 - d. Na linha de comandos, digite `db2 select * from catalog`. Pressione **Enter**. Uma lista de catálogos é exibida. Anote o número do catálogo da loja de exemplo.
 - e. Vá para a etapa 3.
2. Se estiver utilizando o Oracle, faça o seguinte:
 - a. Quando o comando terminar de executar, no menu **Iniciar**, selecione **Programas, Oracle - HomeOra81, Desenvolvimento de Aplicativos e SQL Plus**.
 - b. Na janela, digite seu nome de usuário e senha e cadeia do host.
 - c. Na janela **SQL Plus**, digite `select * from store;`. Pressione **Enter**. É exibida uma lista de lojas. Anote o número da loja que você criou.
 - d. Na janela **SQL Plus**, digite `select * from catalog;`. Pressione **Enter**. Uma lista de catálogos é exibida. Anote o número do catálogo da loja de exemplo.
 - e. Vá para a etapa 3.
3. Abra o Internet Explorer. Vá para a seguinte URL:
`http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=storeId de step1c ou 2c&langId=-1&catalogId=catalogId de step1d ou 2d`
É exibida a sua loja.
Se estiver tendo problemas para exibir a loja, consulte Resolução de problemas de publicação.

Importante:

1. Durante a publicação, o verificador de consistência confirma se os arquivos mencionados pelo archive de loja existem. Se o verificador de consistência encontrar um erro, ele será gravado no log. A publicação continua normalmente.
2. Antes de publicar a loja novamente, exclua os arquivos do seguinte diretório:

```
> NT unidade:\WebSphere\CommerceServer\instances\instancename\cache
> 2000 unidade:\Arquivos de
Programas\WebSphere\CommerceServer\instances\instance name\cache
```

3. Enquanto estiver na fase de desenvolvimento da loja, desative o armazenamento em cache de gatilhos e a cache. Deixar a cache ativada pode resultar no seguinte:
 - Alterações feitas no JSP podem não ser exibidas no navegador.
 - O armazenamento em cache de gatilhos será chamado durante a publicação, quando o banco de dados é atualizado. O armazenamento em cache de gatilhos pode gerar atividade desnecessária no banco de dados que pode resultar em estouro do log de transações do banco de dados. Para obter mais informações, consulte Configurar a cache.
4. Se você tiver efetuado logon como administrador padrão, não poderá navegar em uma loja com base na loja de exemplo business-to-business. Em vez disso, crie um novo usuário pertencente à organização padrão e depois navegue pela loja.

Configurando o Payment Manager para sua Loja

Você pode concluir a configuração do Payment Manager para sua loja utilizando o Administration Console ou a interface com o usuário do Payment Manager. Se você utilizar o Administration Console, os itens de menu aparecem no menu **Payment Manager**. Se você utiliza a interface com o usuário do Payment Manager, os itens de menu aparecem sob **Administração** no quadro de navegação.

Se você criar a loja utilizando o archive de loja de exemplo (recomendado), o Payment Manager será parcialmente configurado.

Para concluir a configuração do Payment Manager para a loja, faça o seguinte:

1. Abra o Administration Console ou a interface com o usuário do Payment Manager.
2. Atribua funções de usuário do Payment Manager a usuários do WebSphere Commerce, conforme necessário. Para atribuir funções de usuário do Payment Manager, selecione **Usuários**.

O Administrador do Site padrão do WebSphere Commerce, `wcsadmin`, é atribuído, por padrão, à função de administrador do Payment Manager. Você pode querer atribuir várias funções do Payment Manager a outros usuários do WebSphere Commerce.

3. Autorize cassetes para sua loja procedendo como segue:
 - a. Selecione **Definições do Comerciante**.
 - b. Clique no nome de sua loja na coluna **Nome do comerciante**.
 - c. Selecione os cassetes que deseja autorizar para sua loja.
 - d. Clique em **Atualizar**.

Importante: Se criou sua loja manualmente, você deverá incluir um novo comerciante (sua loja) para poder autorizar cassetes para sua loja. Ao criar um novo comerciante, o número do comerciante especificado deve corresponder ao ID de Loja do WebSphere Commerce. Você pode criar um novo comerciante selecionando **Definições do Comerciante** clicando, em seguida, em **Incluir um Comerciante**.

4. Configure cassetes para sua loja procedendo como segue:
 - a. Selecione **Definições do Comerciante**.
 - b. Selecione um cassete para ser configurado para sua loja clicando no ícone que aparece na linha e clique na coluna para o cassetes que você deseja configurar.

- c. Clique em **Contas** na página do cassete para sua loja e siga um dos procedimentos abaixo:
- Para alterar contas existentes, clique no nome da conta.
 - Para criar uma nova conta, clique em **Incluir uma nova Conta**.


Para obter mais informações sobre a configuração do Cassete para BankServACH, consulte o *IBM WebSphere Payment Manager for Multiplatforms, Cassette for BankServACH Supplement, Versão 3.1*.

Para obter informações sobre a configuração dos cassetes OfflineCard e CustomOffline, consulte o *IBM WebSphere Payment Manager para Multiplatforms, Manual do Administrador, Versão 3.1*.

Para obter informações sobre como configurar o Cassete para SET, consulte o *IBM WebSphere Payment Manager for Multiplatforms, Cassette for SET Supplement, Versão 3.1*.

Para obter informações sobre como configurar o Cassete para CyberCash, consulte o *IBM WebSphere Payment Manager for Multiplatforms, Cassette for CyberCash Supplement, Versão 3.1*.

Para obter informações sobre como configurar o Cassete para VisaNet, consulte o *IBM WebSphere Payment Manager for Multiplatforms, Cassette for VisaNet Supplement, Versão 3.1*.

Para obter ajuda quando estiver utilizando o Payment Manager no Administration Console ou na interface com o usuário do Payment Manager, clique em  no canto superior direito da página do Payment Manager na qual você está trabalhando.

Para obter mais informações sobre as tarefas acima ou sobre outras tarefas administrativas do Payment Manager, consulte o *IBM WebSphere Payment Manager for Multiplatforms, Administrator's Guide, Versão 3.1*.


Para obter detalhes sobre a instalação do WebSphere Payment Manager, consulte o seguinte:

- *IBM WebSphere Commerce Business Edition, Installation Guide, Versão 5.4*
- *IBM WebSphere Payment Manager for Multiplatforms, Install Guide, Versão 3.1*.

Criando Jobs Programados para uma Loja de Exemplo

Depois de publicar uma loja de exemplo, é necessário criar jobs programados para a loja. A tabela a seguir mostra os jobs programados que devem ser criados para cada loja de exemplo:

Loja de exemplo	Jobs programados requeridos
InFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

Loja de exemplo	Jobs programados requeridos
NewFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAreallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebFashion	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan
 ToolTech	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAreallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebAuction	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

Aqui está uma descrição breve dos jobs:

BalancePayment

Este job chama o comando de tarefa DoDepositCmd para capturar pagamento quando o pedido é enviado. Esse comando implementa a função de captura de pagamento automática para o WebSphere Commerce.

PayCleanup

Este job cancela pedidos do WebSphere Commerce com pedidos de autorização de pagamento rejeitados pelas respectivas instituições financeiras durante um período de tempo maior que o configurado de uma loja.

ProcessBackorders

Este job aloca inventário a pedidos em espera que foram criados quando o inventário não estava disponível.

RAreallocate

(Redistribuir alocações em relação ao inventário esperado) Este job redistribui EIR (Expected Inventory Records, Registros de Inventário Esperado) em relação a pedidos em espera existentes. Isto é necessário para prever com maior precisão quando itens de pedidos em espera estarão disponíveis à medida que as informações de EIR são incluídas ou modificadas e itens anteriormente em espera são excluídos ou alocados.

ReleaseExpiredAllocations

Este job retorna inventário alocado de volta para a tabela de recebimentos a partir de um item de linha de pedido anteriormente alocado que excedeu seu limite de tempo de expiração.

ReleaseToFulfillment

Este job libera itens alocados em um pedido para atendimento.

ReturnCreditAndCloseScan

Este job varre em busca de autorizações de devoluções de mercadorias que são elegíveis para serem creditadas e marcadas como fechadas.

Para criar um job programado, siga as instruções encontradas em Programando um job em nível de loja para cada job. A tabela a seguir lista os parâmetros recomendados para cada job:

Nome do job programado	Hora de início recomendada	Intervalo recomendado (segundos)	Prioridade recomendada
BalancePayment	00:00	86400	1
PayCleanup	00:00	86400	1
ProcessBackorders	00:00	43200	8
RAllocate	00:00	86400	1
ReleaseExpiredAllocations	00:00	3600	8
ReleaseToFulfillment	00:00	600	10
ReturnCreditAndCloseScan	00:00	86400	1

A definição da hora de início como 00:00 inicia o job programado imediatamente.

Nota: O campo **Parâmetros do Job** na janela Programar Job não precisa ser preenchido para esses jobs.

Configurando Notificação de E-mail para uma Loja de Exemplo

O procedimento a seguir ativa procedimento notificação por e-mail para o cliente quando o pagamento é autorizado, quando um pedido é autorizado e quando um pedido é cancelado. Observe que você deve ter um servidor de correio configurado para enviar mensagens a clientes.

Nota: Se você não tem um servidor de correio, não será capaz de enviar notificações por e-mail a partir da loja, mas os demais recursos da loja de exemplo ainda irão funcionar.

As diferentes lojas de exemplo suportam diferentes notificações por e-mail. A tabela a seguir mostra as notificações por e-mail suportadas para cada loja:

Loja de Exemplo	Notificações por e-mail suportadas	Tipo de Mensagem
InFashion	Redefinição de senha	Mensagem de notificação para redefinição de senha

Loja de Exemplo	Notificações por e-mail suportadas	Tipo de Mensagem
NewFashion	Pedido autorizado	Mensagem de notificação para um pedido autorizado
	Redefinição de senha	Mensagem de notificação para redefinição de senha
	Pedido de submissão	Mensagem de notificação para um pedido recebido
	Pedido cancelado	Mensagem de notificação para um pedido cancelado
	Notificação de envio	Mensagem para notificar o cliente sobre uma manifestação de liberação ed pedido
	Lista de itens de interesse	Uma mensagem de divulgação
WebFashion	Pedido autorizado	Mensagem de notificação para um pedido autorizado
	Redefinição de senha	Mensagem de notificação para redefinição de senha
	Lista de itens de interesse	Uma mensagem de divulgação

Nota: A loja de exemplo WebAuction baseia-se na WebFashion. Para configurar a notificação por e-mail, você precisa executar todas as etapas de WebFashion, além de etapas adicionais relacionadas a leilão. Para obter mais informações sobre as etapas de notificação por e-mail para WebAuction, consulte as Tarefas Relacionadas abaixo.

Para ativar a notificação por e-mail, faça o seguinte:

1. Assegure que o servidor de administração IBM WebSphere Application Server esteja iniciado.
2. Abra o Administration Console utilizando um ID de Administrador do Site.
3. Na página Seleção de Site/Loja do Administration Console, selecione **Loja**. A seção Selecionar Loja e Idioma é exibida.
4. Na lista drop down **Nome**, selecione a loja.
5. Na lista drop down **Idioma** selecione o idioma. Clique em **OK**. A home page Loja do Administration Console é exibida.
6. No menu **Configuração**, clique em **Transportes**. A página Configuração de Transporte é exibida.
 - a. Verifique se o transporte de **E-mail** tem o status Ativo. Se o e-mail estiver inativo, selecione-o e clique em **Alterar Status**.
 - b. Selecione **E-mail** e clique em **Configurar**. A página Parâmetros de Configuração de Transporte é exibida.
 - c. No campo **Host**, digite o nome completo do servidor de correio, como `myserver.ibm.com`.
 - d. No campo **Protocolo**, digite `smtp` em letras minúsculas ou o protocolo de sua escolha. Clique em **OK**.
7. No menu **Configuração**, clique em **Tipos de Mensagens**. A página Configuração do Tipo de Mensagem é exibida.

8. Crie a notificação a ser enviada quando o pagamento for autorizado da seguinte maneira:
 - a. Clique em **Nova**. A página Atribuição de Transporte de Mensagens é exibida.
 - b. Selecione o tipo de mensagem na lista drop down **Tipo de Mensagem**. Consulte a tabela acima para ver os tipos de mensagem a serem utilizadas na loja.
 - c. No campo **Gravidade da Mensagem**, digite 0 a 0
 - d. Na lista drop down **Transporte**, selecione **E-mail**.
 - e. Na lista drop down **Formato do Dispositivo**, selecione **Formato de Dispositivo Padrão**.
 - f. Clique em **Avançar**. A página Parâmetros de Atribuição de Transporte é exibida.
 - g. Preencha os campos a seguir:

Host	O nome completo do servidor de correio, como <code>example.ibm.com</code>
Protocolo	Digite <code>smtp</code> (você deve utilizar letras minúsculas) ou o protocolo que você está utilizando.
Destinatário	Digite um endereço de e-mail válido. Este endereço será substituído pelo endereço de e-mail do cliente durante a execução.
Remetente	Digite um endereço de e-mail que você queira utilizar como remetente da mensagem, como <code>orders@example.ibm.com</code> . O endereço deve ser um endereço de e-mail de um usuário válido no servidor de correio.
Assunto	Digite o texto que você deseja exibir como linha de assunto da mensagem, por exemplo, <code>Seu pedido foi aceito</code> .

- h. Clique em **Concluir**. A página Configuração do Tipo de Mensagem é exibida.
9. Para cada tipo de mensagem na loja de exemplo, repita a etapa 8.

Notas:

- Frequentemente será necessário um longo tempo para que o e-mail de notificação de pedido cancelado seja enviado. O tempo pode ser encurtado:
 1. Diminuindo o valor da coluna `REJECTEDORDEREXPIRY` na tabela do banco de dados `STORE`.
 2. Alterando o job programado `PayCleanup` para ter um intervalo de programação menor.
- Para obter mais informações sobre notificações de envio, consulte a mensagem `ReleaseShipNotify`

Capítulo 2. Recursos do Banco de Dados da Loja de Exemplo

Recursos do banco de dados da loja

Dados da loja são as informações carregadas no banco de dados do WebSphere Commerce Server, que permitem que sua loja funcione. Para operar corretamente, uma loja deve ter os dados no lugar para suportar todas as atividades do cliente. Por exemplo, para um cliente efetuar uma compra, sua loja deve conter um catálogo de mercadorias à venda (dados do catálogo), os dados associados ao processamento de pedidos (dados de impostos e de envio) e o inventário para atender o pedido (dados de inventário e de distribuição).

Os dados podem ser exclusivos de uma loja ou compartilhados entre lojas. Para obter mais informações, consulte Recursos de dados compartilhados.

Os recursos de banco de dados da loja nos archives da loja de exemplo fornecidos com o WebSphere Commerce são arquivos XML válidos, bem formados, para o pacote Loader, com as seguintes exceções: os arquivos XML do archive da loja são destinados a serem portáteis e não devem conter chaves principais geradas que são específicas de uma determinada instância do banco de dados. Em vez disso, eles utilizam os aliases (descritos em Convenções do Loader do archive de loja) que são resolvidos pelo IDResolver no momento da publicação. Os archives de loja também utilizam um conjunto de macros de DTD (conhecidas em XML como entidades). As macros agem como marcadores para valores que você seleciona no Store Services durante a criação da loja. A utilização dessas duas convenções permite que os archives de loja de exemplo sejam copiados e publicados várias vezes.

O exemplo de archive de loja inclui todos os recursos de banco de dados necessários para criar uma loja funcional. Você pode modificar estes arquivos para utilização em seu próprio archive de loja, ou utilizá-los como guia para criar seus próprios arquivos XML. O WebSphere Commerce requer que determinados dados sejam carregados no banco de dados do WebSphere Commerce para criar uma loja funcional e que estes dados devem ser carregados na ordem determinada pelo esquema. Por exemplo, a tabela FFMCENTER deve ser ocupada antes da tabela STOREENT. Como as lojas de exemplo incluem todos os dados mandatórios no pedido e a estrutura requerida pelo WebSphere Commerce, ao utilizar os recursos do banco de dados como base, ou guia, para sua própria loja, você economiza muito tempo durante o período inicial de criação.

Para obter uma lista dos arquivos de recursos de banco de dados utilizados no exemplo de archive de loja, consulte Recursos de banco de dados do exemplo de archive de loja. Para obter informações mais detalhadas sobre dados da loja, consulte o *IBM WebSphere Commerce Store - Manual do Desenvolvedor*.

Nota: Os DTDs para os arquivos XML de recursos do banco de dados da loja de exemplo não estão nos arquivos archive de loja. Eles estão localizados no seguinte diretório:

▶ NT	<code>unidade:\WebSphere\CommerceServer\xml\sar</code>
▶ 2000	<code>unidade:\Arquivos de Programas\WebSphere\CommerceServer\xml\sar</code>
▶ AIX	<code>/usr/WebSphere/CommerceServer/xml/sar</code>
▶ Solaris	<code>/opt/WebSphere/CommerceServer/xml/sar</code>
▶ 400	<code>/QIBM/ProdData/WebCommerce/xml/sar</code>

Recursos do Banco de Dados do Catálogo da NewFashion

O arquivo `catalog.xml` armazena as informações do catálogo para as lojas de exemplo do WebSphere Commerce. Cada loja de exemplo tem seu próprio arquivo `catalog.xml`. Para obter mais informações, consulte o arquivo `catalog.xml`.

Os recursos de banco de dados de catálogo da NewFashion são divididos nas seguintes seções:

- Grupos de catálogos
- Entidades do catálogo
- Recurso ATP

Grupos de catálogos

Grupos de catálogos são grupos de categorias e produtos. Uma categoria é um grupo de catálogos. Por exemplo, Moda Masculina é um agrupamento das categorias que formam a Moda Masculina, como Calças e Camisas, enquanto a categoria Calças é um agrupamento de produtos.

Na loja de exemplo NewFashion, há vários grupos de catálogos:

- Moda Masculina
- Moda Feminina
- Novidades
- Promoções Domésticas

Os grupos de catálogos são criados na tabela `CATGROUP` e associados a um catálogo específico na tabela `CATTOGRP`. Os grupos de catálogos podem ter categorias principais e subcategorias. Por exemplo, uma categoria principal pode ser Moda masculina e uma subcategoria Calças. Se você tiver uma subcategoria, você deverá colocá-la sob a categoria principal na tabela `CATGRPREL`.

Nota: Produtos das categorias Promoções Domésticas ou Novidades aparecem somente nas categorias especiais. Depois das promoções terminarem, esses produtos precisam ser movidos de volta a suas categorias regulares.

Entidades do catálogo

O catálogo da loja de exemplo NewFashion é formado por entidades do catálogo. Essas entidades são formadas pelo seguinte:

- produtos
- itens
- pacotes
- pacotes configuráveis

As informações sobre entidades do catálogo são armazenadas na tabela `CATENTRY`. A tabela `CATENTREL` armazena relacionamentos entre entidades do catálogo, como relacionamentos de produto-item, pacote e pacote configurável. Para obter mais informações sobre as entidades, consulte [Pacotes e pacotes configuráveis](#).

Produtos

Para cada produto da loja NewFashion há uma entrada na tabela BASEITEM. Um item base é um produto. Cada produto pode ter somente uma versão que é determinada por uma data de expiração. Portanto, você pode ter somente uma camisa feminina vermelha tamanho 12 que expira em 1 de janeiro de 2010. As informações sobre versão por produto são armazenadas na tabela ITEMVERSN.

Na loja NewFashion, inventário não é compartilhado com outros comerciantes. Como resultado, cada produto do catálogo tem apenas uma entrada na tabela DISTARRANG. Cada linha da tabela DISTARRANG representa uma disposição de distribuição, permitindo que uma loja venda seu próprio inventário.

Itens

Um item é uma instância específica de um produto, definida por atributos. As informações contidas sobre um item específico são armazenadas na tabela ITEMSPC. As informações de inventário de cada item são mantidas na tabela RECEIPT. A data na qual o recebimento de um item em espera é esperado está na tabela RADETAIL.

Nota: Na loja NewFashion, o catálogo utiliza preços para itens na tabela OFFERPRICE. Para obter mais informações, consulte o arquivo `offering.xml`.

Pacotes

Pacotes são produtos que são vendidos juntos, como um pacote de golfe. Um pacote aparece como uma linha no carrinho de compras. Produtos de um pacote não podem ser comprados separadamente. Os produtos são configurados no arquivo `product.xml` e são agrupados em pacotes na tabela CATENTRY. Cada produto do pacote tem uma entrada separada.

Pacotes Configuráveis

Pacotes configuráveis são produtos que são agrupados juntos, mas que podem ser vendidos separadamente. Cada produto de uma pacote configurável aparece como uma linha no carrinho de compras. Os pacotes configuráveis são armazenados na tabela CATENTRY e os produtos associados a um pacote configurável são armazenados na tabela CATGPENREL. O relacionamento pacote configurável-produto é definido na tabela CATENTREL.

Os atributos para entidades de catálogo, pacotes e pacotes configuráveis são definidos nas tabelas PKGATTR e PKGATTRVAL. A tabela PKGATTR faz referência a uma tabela de atributos já existentes que pertence ao produto no pacote. A tabela PKGATTRVAL faz referência ao valor de atributo do produto no pacote.

Recurso ATP

A loja NewFashion tem alguns itens especiais que apresentam o recurso ATP, como os seguintes:

- Masculino, Esportivo, Camisa de Time tem inventário de pronta entrega igual a zero.
- Feminino, Acessórios, Bolsa Social tem pedido em espera forçado.
- Masculino, Esportivo, Camisa de Time tem RADetails.

Se um item for zero para pronta entrega ou pedido em espera forçada, não importa quantos desses itens os clientes pedirem, eles irão automaticamente para pedido em espera. Se não houver RADetail significa que a data esperada para o pedido em espera será a data atual mais o deslocamento padrão de pedido em espera na loja, (coluna DEFAULTBOOFFSET na tabela da loja), definido como 30 dias.

Recursos do Banco de Dados da Política de Negócios e de Contrato da NewFashion

No WebSphere Commerce, todas as lojas precisam ter um contrato padrão. O contrato padrão é formado por termos e condições que são armazenadas no arquivo `contract.xml`. Se os termos e condições fizerem referência a uma política de negócios, as informações da política de negócios são armazenadas no arquivo `businesspolicy.xml`.

As informações dos recursos de banco de dados de política de contratos e negócios da NewFashion são divididas nas seguintes seções:

- Contrato
- Política de negócios

Contrato

Termos e condições

Todos os contratos das lojas de exemplo devem ter termos e condições. Os termos e condições são o conjunto de regras acordados em um contrato e controlam o processo de compras entre um comprador e um vendedor. Para cada contrato, é necessário ter um conjunto de termos e condições de preço.

Você também pode ter termos e condições para outros tipos de encargos, como encargos para envio. Cada contrato deve ter somente um termo e condição para encargo para envio.

O comprador e o vendedor devem ser consultados no `contract.xml`.

Política de negócios

Termos e condições

Uma política delinea as regras seguidas por um negócio para um processo de negócios específico. Se os termos e condições de seu contrato fizer referência a uma política de negócios, você deve ocupar o `businesspolicy.xml` antes de importar o arquivo `contract.xml`.

Nota: Algumas políticas mencionadas em `contract.xml` não estão contidas em `businesspolicy.xml`. Essas políticas fazem parte dos dados bootstrap. Para obter mais informações sobre os dados bootstrap, consulte Arquivos Bootstrap.

Recursos do Banco de Dados de Envio da NewFashion

Os recursos do banco de dados de envio da NewFashion são armazenados nos arquivos XML a seguir:

- shipping.xml
- store-catalog-shipping.xml
- store-defaults.xml
- shipfulfill.xml

Os recursos de banco de dados de envio da NewFashion são divididos nas seguintes seções:

- Jurisdições
- Modos de envio
- Códigos de cálculo
- Regras de cálculo
- Escala de cálculo
- Intervalo de cálculo
- Procura de cálculo
- Combinações de cálculo
- Atendimento de envio

Jurisdições

O arquivo shipping.xml identifica jurisdições para envio. As jurisdições são definidas na tabela JURST, JURSTGROUP atribui a jurisdição a um grupo e uma subclasse e JURSTPREL atribui a jurisdição e o grupo de jurisdições à mesma subclasse.

Modos de envio

Um modo de envio é uma combinação de uma transportadora e seu serviço de envio. Por exemplo, Transportadora XYZ, Entrega 24 horas é um modo de envio. Informações sobre os modos de envio podem ser armazenadas na tabela SHIPMODE.

Códigos de cálculo

Os códigos de cálculo são utilizados para calcular descontos, encargos para envio, impostos sobre vendas e impostos sobre envio. O arquivo shipping.xml contém todos os envios de códigos de cálculo. A tabela CALCODE configura os códigos de cálculo para envio. O campo displaylevel mostra um número com o valor calculado.

- 0 = Item de Pedido
- 1 = Pedido
- 2 = Produto
- 3 = Item
- 4 = Contrato

Regras de cálculo

Cada código de cálculo tem um conjunto de regras de cálculo que define como o cálculo será feito. Por exemplo, se estiver enviando mercadorias para uma região,

pode aplicar determinadas regras para o cálculo e se estiver enviando mercadorias para outra região, pode aplicar regras diferentes para o cálculo. A tabela CALRULE armazena as regras de cálculo para envio. O campo flag especifica se o CalculationCodeQualifyMethod do CalculationCode específico deve ser chamado.

0 = 0 método não será chamado.

1 = 0 método será chamado.

Escala de cálculo

Uma escala de cálculo é o conjunto de intervalos que se aplica ao cálculo. Por exemplo, para encargos para envio você pode ter um conjunto de intervalos de peso, cada um correspondendo a um encargo específico. Ou seja, um produto que pesa entre 0 e 5 kg pode custar R\$10,00 para ser enviado e um produto que pesa entre 5 e 10 kg pode custar R\$15,00 para ser enviado. A tabela CALSCALE armazena o código de escala para envio, um para por pedido e um para por item.

Nota: A tabela CALSCALE armazena o código de escala para as moedas se elas se aplicarem ao intervalo correspondente.

Intervalo de cálculo

O intervalo para os códigos de escala é armazenado na tabela CALRANGE. calmethod_id_10 é utilizado para envio por pedido e calmethod_id_11 para envio por item.

Procura de cálculo

Os valores de procura de cálculo são os valores associados à escala de cálculo. Os valores da procura de cálculo, para um produto que pesa entre 0 e 5 kg pode custar R\$10,00 para ser enviado e um produto que pesa de 5 a 10 kg pode custar R\$15,00 para ser enviado, seriam R\$10,00 e R\$15,00 respectivamente. Há um valor de procura por moeda para um determinado CALRANGE ID. A tabela CARLOOKUP define o ID de procura e o valor.

Combinações de cálculo

As regras de cálculo e os intervalos de escala são combinados na tabela CRULESCALE, como visto no código a seguir: Métodos e regras de cálculo são combinados na tabela STENCALUSG. O padrão da loja para cálculos também é armazenado nessa tabela. O campo usageflag controla a forma como o comando OrderPrepare utiliza o cálculo.

1 = utilizar - utiliza este CalculationUsage.

2 = verificar - lança um ECAApplicationException se este cálculo não produzir um valor para um item de pedido.

Atendimento de envio

Os recursos de atendimento de envio associam um grupo de jurisdições de envio às regras de cálculo e o centro de distribuição a shipmode da loja. As informações

sobre atendimento do envio são armazenadas nas tabelas SHPJCRULE e SHPARRANGE e podem ser vistas a seguir como exibidas em `shipfulfill.xml`.

Recursos de banco de dados de controle de acesso da loja de exemplo

No WebSphere Commerce, todas as lojas têm políticas de controle de acesso. Uma política de controle de acesso autoriza usuários ou um grupo de usuário a executar determinadas ações. Existem dois arquivos de política de controle de acesso para cada loja:

- `samplestorenameAccessPolicies.xml`
- `samplestorenameAccessPolicies_locale.xml`

O `AccessPolicies.xml` e o `AccessPolicies_locale.xml` são arquivos de controle de acesso nativo de alto nível. `AccessPolicies.xml` não depende do idioma nacional, enquanto `AccessPolicies_locale.xml` depende do idioma nacional. Cada arquivo inclui as possíveis ações, grupos de ações, recursos e definições de política utilizados nas lojas de exemplo. Esses arquivos são transformados nos arquivos `AccessPoliciesOut.xml` e `AccessPoliciesOut_locale.xml`, respectivamente. Cada arquivo transformado ocupa o banco de dados. Para obter mais informações sobre como transformar os arquivos de política de acesso, consulte os links relacionados a seguir.

Nota: Somente os arquivos transformados podem ser carregados em massa ou utilizados diretamente no arquivo SAR; os arquivos pré-transformados não podem.

As informações sobre recurso de banco de dados das lojas de exemplo podem ser divididas nas seguintes seções:

- Ações
- Categorias de Recursos
- Grupos de Recursos
- Grupos de Ações
- Definições de Políticas

Ações

Ações que podem ser executadas sob a política de controle de acesso são definidas no arquivo `AccessPolicies.xml` de cada loja de exemplo.

Categorias de Recursos

As categorias de recursos definem recursos que podem ser protegidos.

Grupos de Recursos

Os grupos de recursos contêm os recursos controlados pela política de controle de acesso. Um grupo de recursos pode incluir objetos de negócios como “contrato” ou “posição de troca” ou um conjunto de comandos relacionados. O arquivo `AccessPolicies.xml` de cada loja de exemplo define os grupos de recursos na política.

Grupos de Ações

Os grupos de ações definem as ações que podem ser executadas nos grupos de recursos na política de controle de acesso. Esses grupos são definidos no arquivo `AccessPolicies.xml` para cada loja.

Definições de Políticas

A política de cada loja de exemplo é definida no arquivo `AccessPolicies.xml` da loja individual. A loja de exemplo ToolTech tem duas políticas.

Capítulo 3. Loja de Exemplo NewFashion

Loja de exemplo NewFashion

A NewFashion é uma das lojas de modas online business-to-consumer fornecidas com o WebSphere Commerce. Ela implementa muitos dos recursos mais utilizados nos principais sites varejistas atuais. Alguns dos recursos incluídos na loja de exemplo NewFashion são:

- Suporte multicultural
- Datas de disponibilidade para itens de pedidos
- Pedidos em espera de itens que não estão no estoque
- Divisão de pedidos com base na preferência do cliente
- Rastreamento de status de pedidos
- Notificação por e-mail do status do pedido
- Capacidade de pesquisa
- Colaboração através de suporte ao cliente em tempo real

A loja NewFashion fornece todas as páginas e recursos necessários para uma loja funcional. Ela é fornecida no pacote do WebSphere Commerce como um archive de loja e, portanto, não é necessária uma instalação adicional. Para exibir a loja de exemplo basta criar um novo archive de loja com base na NewFashion, utilizando as ferramentas do Store Services, e depois publicá-la no WebSphere Commerce Server. Para obter mais informações, consulte Criar um archive de loja utilizando o Store Services.

Como a criação de loja no WebSphere Commerce está baseada na seleção de um archive de loja de exemplo e na modificação dele, a NewFashion foi desenvolvida para agir como a base de sua loja. Tem como base um fluxo de compras único, já comprovado e todas as páginas da loja de exemplo podem ser facilmente personalizadas.

Para obter mais informações sobre o fluxo de compras da NewFashion, consulte o diagrama do fluxo de compras da loja de exemplo e os casos de utilização. Os casos de utilização detalham o fluxo de interação de cada usuário na loja, por exemplo, registro ou exibição de um produto. Para obter mais detalhes técnicos sobre como cada página funciona, consulte as informações de referência correspondentes para cada página.

Capítulo 4. Páginas da Loja NewFashion

Páginas de pesquisa da loja de exemplo NewFashion

A página de resultados da pesquisa, `resultlist.jsp` é exibida depois que um cliente digita um pedido de pesquisa na home page, `sidebar.jsp`, ou na página de pesquisa avançada, `advancedsearch.jsp`.

Nota: A página `subcategory.jsp` não inclui `sidebar.jsp` como a barra lateral da página. Como resultado, a funcionalidade de pesquisa foi construída diretamente nessa página e os recursos de pesquisa são os mesmos que os de `sidebar.jsp`.

Beans

`resultlist.jsp` utiliza os seguintes beans:

- `CatalogDataBean`
- `CategoryDataBean`
- `CatEntrySearchListDataBean`
- `CatalogEntryDataBean`

`advancedsearch.jsp` utiliza os seguintes beans:

- `CatalogDataBean`
- `CategoryDataBean`

Comandos

`resultlist.jsp` utiliza os seguintes comandos:

- `ProductDisplay`
- `AdvancedSearchView`
- `CatalogSearchResultView`

`advancedsearch.jsp` utiliza os seguintes comandos:

- `CatalogSearchResultView`

`sidebar.jsp` utiliza os seguintes comandos:

- `CatalogSearchResultView`
- `AdvancedSearchView`
- `LogonForm` (apenas Pesquisa)
- `HelpView` (apenas Pesquisa)
- `StoreCatalogDisplay` (apenas Pesquisa)
- `SetCurrencyPreference` (apenas Pesquisa)

`subcategory.jsp` utiliza os seguintes comandos:

- `CatalogSearchResultView`
- `AdvancedSearchView`

Nota: A página `subcategory.jsp` não inclui `sidebar.jsp` como a barra lateral da página. Como resultado, a funcionalidade de pesquisa foi construída diretamente nessa página e os recursos de pesquisa são os mesmos que os de `sidebar.jsp`.

Implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando o cliente clica em **IR** na tela de navegação esquerda, `sidebar.jsp` ou `subcategory.jsp`, ou clica em **Enviar** na página de pesquisa avançada `advancedsearch.jsp`, `resultlist.jsp` será exibido.

Quando `CatEntrySearchListDataBean` é ativado, ele vai ao banco de dados e recupera todos os resultados da pesquisa, conforme é visto no código abaixo.

```
com.ibm.commerce.beans.DataBeanManager.activate(catEntSearchListBean, request);
```

O parâmetro `request` é a cadeia de URL que é passada para a página `resultlist.jsp`. Em `CatEntrySearchListDataBean`, o método `getResultList()` retorna uma matriz de `CatalogEntryDataBeans`. Cada `CatalogEntryDataBean` é um resultado que é listado na página de resultados da pesquisa. O método `getResultList()` retorna um número específico de resultados por página. Esse número é especificado no endereço de pedido da URL com a variável `pageSize` quando `CatEntrySearchListBean` for ativado. Quando `result.jsp` for exibido e os clientes clicarem nos botões **Anterior** ou **Próximo**, `CatalogSearchResultView` lista os resultados e o parâmetro de URL `beginindex` é passado para `CatalogSearchResultView`. `CatalogSearchResultView` informa `CatEntrySearchListDataBean` em qual resultado de número na lista de resultados de pesquisa a exibição deve começar.

Na loja de exemplo `NewFashion`, somente produtos são exibidos nos resultados de pesquisa. Quando o comando `CatalogSearchResultView` é chamado, a variável `resulttype` é passada para o comando como um argumento de URL. `resulttype` informa `CatEntrySearchListBean` para listar os produtos, itens ou ambos. Os valores de `resulttype` são:

- 1 = Listar somente itens.
- 2 = Listar somente produtos.
- 3 = Listar itens e produtos.

O código a seguir lista somente produtos.

```
<input type="hidden" name="resultType" value="2">
```

Limitações

O catálogo da loja segue uma estrutura hierárquica com categorias principais no topo, como **Masculino**, **Feminino** e **Novidades**, e subcategorias entre elas, como **Camisas** e **Calças**.

Quando os clientes selecionam **Pesquisa Avançada**, podem pesquisar somente na categoria principal ou na subcategoria escolhida. Se o cliente pesquisar na categoria **Masculino**, somente os produtos que estão diretamente sob essa categoria serão exibidos. Os produtos na categoria **Camisas** ou **Calças Masculinas** não serão exibidos. Para pesquisar na subcategoria **Calças** ou **Camisas**, os clientes devem selecionar essa categoria específica, para que os resultados da pesquisa sejam exibidos.

Páginas do Catálogo de Endereços da Loja de Exemplo

As páginas do catálogo de endereços da loja de exemplo permitem que clientes registrados incluam endereços, inclusive endereços para cobrança e de envio, em um catálogo de endereços.

Um cliente registrado efetua login e clica em **Editar meu catálogo de endereços** a partir da página Minha Conta (`account.jsp`). A página Catálogo de Endereços (`addressbookform.jsp`) será exibida, a partir da qual um cliente pode incluir um novo endereço ou editar um existente. Para obter mais informações, consulte o Caso de utilização Incluir novo endereço e o Caso de utilização Editar um endereço.

Os procedimentos do catálogo de endereços da loja de exemplo utilizam os seguintes arquivos JSP:

- `account.jsp` (página Minha Conta)
- `addressbookform.jsp` (página Catálogo de Endereços)
- `addressform.jsp` (inclui os parâmetros para o comando `AddressForm`. Não é exibida para o cliente)
- `address.jsp` (página Incluir Endereço e Atualizar Endereço)
Nota: `address.jsp` é utilizada para ambas as páginas Incluir Endereço e Atualizar Endereço. Se o `addressId` for fornecido, a página `address.jsp` será carregada como Atualizar Endereço. Caso contrário, ela carrega como uma página Incluir Endereço. Se o `addressId` for fornecido como parâmetro para o comando `AddressAdd`, este comando atualiza o endereço do `addressId` especificado. Caso contrário, um novo endereço será criado.

Comandos

`account.jsp` utiliza os seguintes comandos:

- `UserRegistrationForm`
- `AddressBookForm`

`addressbook.jsp` utiliza os seguintes comandos:

- `AddressForm`
- `AddressDelete`

`address.jsp` utiliza os seguintes comandos:

- `AddressAdd`
- `PrivacyView`

Beans

`addressbook.jsp` utiliza os seguintes beans:

- `AddressAccessBean`

`address.jsp` utiliza os seguintes beans:

- `ErrorDataBean`
- `AddressDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte Páginas da InFashion: técnicas comuns de implementação.

Quando um cliente clica em **Editar meu catálogo de endereços** a partir da página Minha Conta, o comando AddressBookForm é chamado. AddressBookForm carrega então a página Catálogo de Endereços (addressbook.jsp). Se um cliente clica em **Incluir um novo endereço** o comando AddressForm é chamado. AddressForm é registrado com AddressForm.jsp no banco de dados e verifica o parâmetro page. Se page for definido como newshipaddressa página Compra 1: Incluir Endereço de Cobrança (billingaddress.jsp) será carregada; caso contrário, a página Incluir Endereço (address.jsp) será carregada.

Nota: AddressForm carregará a página billingaddress.jsp se estiver definido para newshipaddress por causa do tratamento de erros para incluir endereço de cobrança. Se page for igual a newshipaddress, significa que o cliente estava criando um novo endereço a partir da página de endereço de cobrança e tentando entrar na página de endereço de envio, mas ocorreu um erro. Como resultado, o cliente será enviado de volta para a página Endereço de Cobrança.

Quando o cliente tiver preenchido os campos na página Incluir Endereço (address.jsp), ele verifica se o addressId existe. Se o addressId existir, o catálogo de endereços será atualizado. Caso contrário, será criado um novo endereço.

Se o addressId for fornecido, a página address.jsp será carregada como Atualizar Endereço. Caso contrário, ela carrega como uma página Incluir Endereço. Na página Atualizar Endereço, os campos de entrada estão preenchidos com os valores digitados anteriormente, conforme visto abaixo:

Antes de um novo endereço ser criado, um pseudônimo (um identificador exclusivo para o endereço, incluindo data e hora) é criado para o endereço, utilizando o seguinte JavaScript:

Quando um cliente preenche o endereço e clica em **Enviar** na página Incluir Endereço (address.jsp) e Atualizar endereço (address.jsp), o comando AddressAdd é chamado. A página do catálogo de endereços (addressbook.jsp) exibirá os endereços já existentes.

Tratamento de erros

Se o cliente não preenche os campos necessários na página Incluir Endereço (address.jsp) ou Atualizar Endereço (address.jsp), o sistema pede que ele preencha os campos novamente.

Páginas do catálogo de endereços da loja de exemplo NewFashion

As páginas do catálogo de endereços da loja de exemplo permitem que clientes registrados incluam endereços para cobrança e de envio em um catálogo de endereços.

Um cliente registrado efetua logon e clica em **Editar meu catálogo de endereços** na página Minha Conta (myaccount.jsp). A página Catálogo de Endereços (addressbookform.jsp) será exibida. Nela, um cliente pode incluir um novo endereço ou editar um existente. Para obter mais informações, consulte o Caso de utilização Incluir novo endereço e o Caso de utilização Editar um endereço.

O catálogo de endereços da loja de exemplo utiliza os seguintes arquivos JSP:

- `account.jsp` (página Minha Conta)
- `addressbookform.jsp` (página Catálogo de Endereços)
- `addressform.jsp` (inclui os parâmetros para o comando `AddressForm`. Não é exibida para o cliente)
- `address.jsp` (página Incluir Endereço e Atualizar Endereço)
Nota: `address.jsp` é utilizada para ambas as páginas Incluir Endereço e Atualizar Endereço. Se o `addressId` for fornecido, a página `address.jsp` será carregada como Atualizar Endereço. Caso contrário, será carregada como Incluir Endereço. Se o `addressId` for fornecido como parâmetro para o comando `AddressAdd`, o comando atualizará o endereço do `addressId` especificado. Caso contrário, um novo endereço será criado.

Comandos

`account.jsp` utiliza os seguintes comandos:

- `AddressBookForm`
- `InterestItemDisplay`
- `TrackOrderStatus`

`addressbookform.jsp` utiliza os seguintes comandos:

- `AddressForm`
- `AddressDelete`
- `OrderItemDisplay`

`address.jsp` utiliza os seguintes comandos:

- `AddressAdd`
- `PrivacyView`

Beans

`addressbook.jsp` utiliza os seguintes beans:

- `AddressAccessBean`
- `AddressBookDataBean`

`address.jsp` utiliza os seguintes beans:

- `ErrorDataBean`
- `AddressDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente clica em **Editar meu catálogo de endereços** na página Minha Conta, o comando `AddressBookForm` é chamado. `AddressBookForm` carrega então a página Catálogo de Endereços (`addressbookform.jsp`). Se um cliente clica em **Incluir um novo endereço**, o comando `AddressForm` é chamado. `AddressForm` é registrado com `AddressForm.jsp` no banco de dados e verifica o parâmetro `page`. Se

page for definido para newshipaddress, a página Compra 1: Incluir Endereço de Cobrança (billingaddress.jsp) será carregada; caso contrário, a página Incluir Endereço (address.jsp) será carregada.

Nota: AddressForm carregará a página billingaddress.jsp se estiver definido para newshipaddress por causa do tratamento de erros para incluir endereço de cobrança. Se page for igual a newshipaddress, significa que os clientes estavam criando um novo endereço a partir da página de endereço de cobrança e tentando ir para a página de endereço de envio, mas ocorreu um erro. Como resultado, os clientes são enviados de volta à página Endereço de Cobrança.

Depois que o cliente preenche os campos na página Incluir Endereço (address.jsp), o sistema verifica se addressId existe. Se addressId existe, o catálogo de endereços é atualizado. Caso contrário, um novo endereço será criado. Se o addressId for fornecido, a página address.jsp será carregada como Atualizar Endereço. Caso contrário, ela carrega como uma página Incluir Endereço. Na página Atualizar Endereço, os campos de entrada estão preenchidos com os valores digitados anteriormente, conforme visto abaixo:

```
if (addressId != null)
bUpdateAddress = true;
else
bUpdateAddress = false;
```

Quando um cliente preenche o endereço e clica em **Enviar** na página Incluir Endereço (address.jsp) e Atualizar endereço (address.jsp), o comando AddressAdd é chamado. A página de catálogo de endereços (addressbookform.jsp) exibe os endereços existentes.

Os clientes podem clicar em **Editar Catálogo de Endereços** na página Selecionar endereço de envio para ir para o Catálogo de endereços. Se os clientes forem para a página Catálogo de endereços a partir da página Selecionar endereço de envio, a página Catálogo de endereços exibe um link **Retornar à Compra**. Esses clientes não têm permissão para remover um endereço da página Catálogo de Endereços. Como resultado, novo código é incluído para verificar se os clientes estão vindo da página Selecionar endereço de envio, conforme visto abaixo:

```
String mode = jhelper.getParameter("mode"); if
(mode.equals("AddressBookReturnToCheckout"))
```

Tratamento de erros

Se o cliente não preencher os campos obrigatórios na página Incluir Endereço (address.jsp) ou Atualizar Endereço (address.jsp), o sistema solicita que o cliente digite os campos novamente. O seguinte código trata do erro:

```
TypedProperty hshErrorProperties = bnError.getExceptionData();

if (hshErrorProperties != null) { //Há um erro no endereço enviado.
strErrorCode = hshErrorProperties.getString(ECConstants.EC_ERROR_CODE, "");
if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_NICKNAME))
strErrorMessage = infashiontext.getString("ERROR_MESSAGE1");

...

}
```

Páginas de exibição de pacote Configurável das lojas de exemplo WebFashion e NewFashion

Uma página de exibição de pacote configurável apresenta grupos promocionais de produtos em uma loja online. Normalmente, ela inclui uma descrição, uma lista dos componentes que compõem o pacote configurável, o preço de cada componente, uma imagem e uma lista de atributos (tamanho e cor), se os produtos que compõem o pacote configurável tiverem variações, e uma lista de valores para cada atributo (vermelho e azul para cor, e Grande e Extra-Grande para tamanho).

Os clientes podem incluir todos os produtos do pacote configurável no carrinho de compras ou na lista de itens de interesse com um único clique. Cada produto no pacote configurável é exibido como um item de linha separado no carrinho de compras, ao contrário do pacote, que exibe um único item de linha para o pacote inteiro.

Para obter mais informações sobre as páginas de exibição de pacotes configuráveis da loja de exemplo e como elas funcionam, consulte o Caso de utilização da página de exibição de pacotes configuráveis.

Comandos

O `bundledisplay.jsp` utiliza os seguintes comandos:

- `OrderItemAdd`
- `InterestItemAdd`

Beans

O `bundledisplay.jsp` utiliza os seguintes beans:

- `BundleDataBean`
- `CompositeProductDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

O `Bundledisplay.jsp` é registrado no banco de dados (na tabela `DISPENTREL`) para exibir todos os produtos na loja. O `Bundledisplay.jsp` exibe o seguinte:

- Descrição (inclusive uma lista de componentes que compõem o pacote configurável) e imagem do pacote configurável
- Uma imagem, preço e uma lista de atributos (tamanho e cor) para cada produto do pacote configurável. Exibe também uma lista de valores para cada atributo (vermelho e azul para cor e Grande e Extra-Grande para tamanho.)
- Link Incluir no carrinho de compras e Incluir na lista de itens de interesse.
- Campo Quantidade (apenas NewFashion)

Descrição (inclusive uma lista de componentes que compõem o pacote configurável) e imagem do pacote configurável

O `BundleDataBean` recupera a descrição e a imagem do pacote configurável.

Imagem do produto, preço, atributos e valores de atributos

O BundleDataBean chama o ProductDataBean para recuperar as informações sobre cada produto que compõe o pacote configurável. Para obter mais informações sobre como o ProductDataBean funciona, consulte Páginas de produtos da loja de exemplo.

Incluir no carrinho de compras e Incluir na lista de itens de interesse

Para obter mais informações, consulte Páginas de produtos da loja de exemplo.

Campo Quantidade (apenas NewFashion)

Há um campo de quantidade para cada produto no pacote configurável, que permite que o cliente especifique o número de cada produto a incluir no carrinho de compras.

Páginas do grupo de catálogos da Loja de Exemplo

As páginas de grupos de catálogos ajudam os clientes a navegar através dos vários departamentos ou grupos de produtos ou serviços disponíveis em uma loja. As primeiras páginas de grupos de catálogos levam os clientes às áreas nas quais eles desejam fazer compras e geralmente exibem o nome de cada grupo de catálogos, uma descrição breve e uma imagem. As páginas de grupos de catálogos subsequentes estreitam ainda mais o tipo de produto selecionado que o cliente deseja procurar. A última página do grupo de catálogos em um caminho de compras contém links para páginas de produtos.

Geralmente existem três tipos de páginas de grupos de catálogos:

- páginas que fazem link com páginas de grupos de subcatálogos (páginas de grupos de catálogos)
- páginas que fazem link com páginas de produtos (páginas de listas de produtos)
- páginas que fazem link com páginas de grupos de subcatálogos e páginas de produtos

Para obter mais informações sobre as páginas de grupos de catálogos da loja de exemplo e como elas funcionam, consulte Exibir caso de utilização de categorias de produtos.

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte páginas da Loja de exemplo: técnicas comuns de implementação.

Exibição de páginas de grupos de catálogos e exibição de páginas de listas de produtos

As lojas de exemplo apresentam páginas de grupos de catálogos, as quais exibem as subcategorias em cada uma das categorias de nível superior (Masculinas e Femininas) bem como um produto de destaque especial e páginas de listas de produtos, as quais exibem uma lista de produtos em cada uma das subcategorias (Calças e Camisas).

Todos os hyperlinks para as categorias de nível superior contêm um parâmetro extra chamado top que é definido como Y. O arquivo da página JavaServer categorydisplay.jsp é registrado no banco de dados (na tabela DISPCGREL) como a página que exibe todas as categorias da loja. Nessa página, verificamos a existência do parâmetro top. Se ele existir, a página

inclui `topcategory.jsp`. Caso contrário, `subcategory.jsp` será incluído. `topcategory.jsp` é a página de grupos de catálogos e `subcategory.jsp` é a página de listas de produtos.

Página de grupos de catálogos

`topcategory.jsp` recupera uma lista de subcategorias utilizando o método `getSubCategories()` localizado em `CategoryDataBean`. Os produtos especiais contidos na InFashion foram criados no catálogo, em vez de como parte de uma campanha com o WebSphere Commerce Accelerator. Os produtos especiais apresentados foram incluídos no grupo de catálogos e, em seguida, recuperados por `topcategory.jsp` utilizando o método `getProducts()` localizado em `CategoryDataBean`.

Na WebFashion e na NewFashion, o `topcategory.jsp` da moda masculina exibe um pacote configurável e o `topcategory.jsp` da moda feminina exibe um pacote. O pacote configurável é recuperado utilizando o método `getBundles()` no `CategoryDataBean` e o pacote é recuperado utilizando o método `getPackages()` no `CategoryDataBean`.

Página de listas de produtos

`subcategory.jsp` exibe uma lista ou todos os produtos no grupo de catálogos e uma lista de todas as categorias semelhantes (todas as outras categorias sob a mesma categoria de nível superior) na barra de navegação esquerda. Para cada produto, `subcategory.jsp` exibe a descrição resumida do produto, a imagem normal e o preço, utilizando os seguintes métodos: `getDescription().getShortDescription()`, `getCalculatedContractPrice()` e `getDescription().getFullImage()`. O ID da categoria pai é necessário para exibir a lista de categorias semelhantes. Para recuperar o ID da categoria pai, o parâmetro `parent_category_rn` é fornecido no hyperlink, o qual constrói então um `CategoryDataBean` para a categoria pai. Por padrão, `CategoryDataBean` recupera o ID da categoria a partir do parâmetro `CategoryId`. Entretanto, neste caso, o ID é armazenado em `parent_category_rn`. Como resultado, o ID da categoria deve ser definido explicitamente como segue:

```
String parentCategoryId =
request.getParameter("parent_category_rn"); parentCategory = new
CategoryDataBean (); parentCategory.setCategoryId(parentCategoryId);
com.ibm.commerce.beans.DataBeanManager.activate(parentCategory,
request);
```

Uma lista de produtos que pertencem a este grupo de catálogos é então recuperada utilizando o método `getProducts()` no `CategoryDataBean`.

Nota: Na loja NewFashion, a página `subcategory.jsp` também contém um link para o recurso de assistência ao cliente de colaboração, se estiver ativado no Store Services. A página `subcategory.jsp` tem o link incluído porque ela tem sua própria barra lateral e não inclui `sidebar.jsp` como as outras páginas.

Se os clientes clicarem em **Bate-papo ao Vivo com Assistência ao Cliente**, será exibida uma janela popup na tela e eles poderão bater papo online em tempo real com um representante de atendimento ao cliente. Este link somente será exibido se este recurso estiver ativado no Store Services. O link **Bate-papo ao Vivo com a Assistência ao Cliente** está contida no corpo de um par de tags personalizadas (a tag `<flow:ifEnabled feature="customerCare">` e a tag `</flow:ifEnabled>`) e pode ser ativado ou desativado com base na opção selecionada em Store Services. Enquanto a tag personalizada permanece no local, o Store Services pode

ser utilizado para alternar automaticamente entre um site que tem suporte à colaboração e um que não tem, sem precisar modificar a página JavaServer. Para ativar ou desativar permanentemente o suporte à colaboração na página, as tags personalizadas e o link **Bate-papo ao Vivo com Assistência ao Cliente** podem ser removidos da página JavaServer clicando em **Aplicar Permanentemente**, na GUI do Store Services. Não é recomendável remover manualmente ou alterar as tags personalizadas ou o código contido nelas. Em vez disso, utilize o botão **Aplicar Permanentemente** no Store Services.

Nota: A tag personalizada que contém o link **Bate-papo ao Vivo com Assistência ao Cliente** não pode ser copiada para as páginas JavaServer de outras lojas. Essas tags destinam-se apenas a funcionar na loja que as contém originalmente. Para obter mais informações sobre colaboração, consulte os links relacionados abaixo.

Páginas da loja de exemplo: técnicas comuns de implementação

A maior parte das páginas da loja de exemplo utiliza as técnicas de implementação a seguir. Esta página utiliza a loja InFashion como exemplo, mas pode ser aplicada a todas as lojas de exemplo. Para obter mais informações sobre as técnicas específicas para uma página individual, consulte o arquivo de referência para essa página.

Conteúdo multicultural

As páginas da loja de exemplo também são utilizadas para exibir conteúdo multicultural. Isto é, o mesmo conjunto de páginas pode ser utilizado para vários locais diferentes. A maior parte do código para ativar uma exibição multicultural está em `getResource.jsp`. `getResource.jsp` executa as ações a seguir:

- Recupera o locale atual do contexto de comando e o armazena como a variável do locale.
- Recupera o diretório da loja e o armazena como a variável de `storeDir`.
- Recupera o nome da loja e o armazena como a variável de `storeName`.
- Carrega o arquivo de propriedades específicas do idioma utilizando a API `ResourceBundle` e o armazena como a variável de `infashiontext`.

Para que um arquivo JavaServer Page acesse as variáveis acima, `getResource.jsp` deve ser incluído no arquivo JavaServer Page, utilizando a ação de inclusão de tempo de compilação:

```
<%@ include file="include/getResource.jsp"%>
```

Como `getResource.jsp` está incluído em quase todos os arquivos JavaServer Page das lojas de exemplo, ele pode ser executado várias vezes em uma única solicitação. Para evitar duplicação do esforço, muitas das informações recuperadas nesta página são armazenadas no contexto do pedido. Por exemplo:

```
String storeDir = (String) request.getAttribute("storeDir"); String  
includeDir = (String) request.getAttribute("includeDir"); String fileDir =  
(String) request.getAttribute("fileDir"); String bundleDir = (String)  
request.getAttribute("bundleDir");
```

```
String storeName = "";
```

```

if (storeDir == null) {

storeDir = sdb.getJspPath(); fileDir = sdb.getFilePath(); includeDir =
storeDir + "include" + "/"; bundleDir = sdb.getDirectory(); storeName =
sdb.getDescription(cmdcontext.getLanguageId()).getDisplayname();
request.setAttribute("storeName", storeName);
request.setAttribute("storeDir", storeDir); request.setAttribute("fileDir",
fileDir); request.setAttribute("includeDir", includeDir);
request.setAttribute("bundleDir", bundleDir); }

```

Mensagens específicas do idioma

Mensagens específicas do idioma como “Agradecemos o seu pedido!” são armazenadas em arquivos de propriedades de pacotes configuráveis de recursos. Estes arquivos estão localizados no seguinte diretório:

▶ NT

unidade: \WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storedir

▶ 2000

unidade: \Arquivos de Programas\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storedir

▶ AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir

▶ Solaris

/opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir

▶ 400

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir

Por exemplo, se o diretório de sua loja for “storedir”, então o arquivo de propriedade em inglês será:

▶ NT

unidade: \WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storedir\infashiontext_en_US.properties.

▶ 2000

unidade: \Arquivos de Programas\WebSphere\AppServer\installedApps\WC_Enterprise_App_instance_name.ear\wcstores.war\WEB-INF\classes\storedir

▶ AIX

/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir/infashiontext_en_US.properties.

▶ Solaris

/QIBM/UserData/WebASAdv4/WAS_instance_name/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/WEB-INF/classes/storedir/infashiontext_en_US.properties

O conteúdo deste arquivo é carregado utilizando a API `java.util.ResourceBundle` do Java a partir do arquivo JSP `getResource.jsp`. O pacote é armazenado como a variável `infashiontext`. Uma mensagem específica do idioma é exibida como segue:

```
<title><%=infashiontext.getString("REGISTER_TITLE")%></title>
```

Codificação do conteúdo da definição

A maior parte dos navegadores modernos interpretam dados HTML codificados em codificação UTF-8. Entretanto, alguns navegadores mais antigos podem entender apenas codificação nativa. Por exemplo, um navegador japonês antigo pode interpretar apenas dados HTML codificados em codificação "Shift_JIS". Para resolver esse problema, a loja de exemplo especifica a codificação específica do idioma nos arquivos de propriedades do pacote configurável de recursos como a propriedade de nome ENCODESTATEMENT. Por exemplo, o arquivo de propriedade em inglês `infashiontext_en_US.properties` contém a entrada a seguir:

```
ENCODESTATEMENT = text/html; charset=ISO_8859-1
```

Para arquivos JSP individuais, a codificação é definida utilizando o objeto de pedido do JSP como segue:

```
<% response.setContentType(infashiontext.getString("ENCODESTATEMENT")); %>
```

A definição do tipo de codificação deve ser feita o mais cedo possível no arquivo JSP, pois o cabeçalho HTTP é enviado antes de qualquer outro conteúdo HTML. Como resultado, se você definir o tipo de conteúdo (que é enviado como cabeçalho HTTP) após qualquer conteúdo HTML ele poderá não apresentar o efeito desejado. O navegador pode não ser capaz de exibir os dados adequadamente.

Incluindo o cabeçalho, rodapé e quadro de navegação esquerdo

Quase todas as páginas da loja de exemplo exibem as páginas de cabeçalho (`header.jsp`), rodapé (`footer.jsp`) e quadro de navegação esquerdo (`sidebar.jsp`). Elas são incluídas nos outros arquivos JSP utilizando a diretiva de inclusão de tempo de execução a seguir:

```
<% String incfile;
```

```
incfile = includeDir + "header.jsp"; %> <jsp:include page="<%=incfile%>"  
flush="true"/>
```

Se souber a localização exata dos arquivos JSP, você poderá simplificar o processo de inclusão utilizando o seguinte:

```
<jsp:include page="/storedir/include/header.jsp"/>
```

em que `header.jsp` está localizado no diretório `storedir` sob a raiz de documento do aplicativo da Web.

Página Entre em Contato Conosco da Loja de Exemplo

Quando um cliente clica no link **Entre em Contato Conosco**, a página Entre em Contato Conosco (`contact.jsp`) é exibida. Ela exibe os diversos tipos de informações de contato para a loja, incluindo números de telefone e localizações para lojas "não virtuais".

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte páginas da Loja de exemplo: técnicas comuns de implementação.

Uma nova exibição chamada `ContactView` é criada na tabela `VIEWREG` e associada a `contact.jsp`. A URL da página de privacidade é:
`http://machine_name/webapp/wcs/stores/servlet/ContactView?parameter_list`

Rodapé da Loja de Exemplo

As lojas de exemplo incluem um rodapé na parte inferior de cada página (`footer.jsp`) que inclui links para o seguinte:

- Início
- Carrinho de compras
- Minha conta
- Entre em contato conosco
- Política de privacidade
- Ajuda

Comandos

`footer.jsp` utiliza os seguintes comandos:

- `StoreCatalogDisplay`
- `OrderItemDisplay`
- `LogonForm`
- `ContactView`
- `PrivacyView`
- `HelpView`

Beans

`footer.jsp` utiliza os seguintes beans:

- `UserRegistrationDataBean`
- `ErrorDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte Páginas da loja de exemplo: técnicas de implementação comuns.

Carrinho de compras

O link **CARRINHO DE COMPRAS** no rodapé liga ao comando `OrderItemDisplay` que retorna o comando de exibição `OrderItemDisplayViewShiptoAssoc`. O comando de exibição `OrderItemDisplayViewShiptoAssoc` está registrado com `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` carrega `shoppingcart.jsp` para exibir a página do Carrinho de Compras.

Para obter mais informações sobre os comandos, consulte Comandos e “WebSphere Commerce - Manual do Programador.”

Minha conta

Se você é um cliente registrado, clicar em **Minha conta** leva você à página Minha Conta (`account.jsp`). Se não for um cliente registrado, clicando em **Minha conta** levará você à página Registrar ou Efetuar Logon (`myaccount.jsp`). Isso é realizado por meio do seguinte código:

```
if (userType.equalsIgnoreCase("G")){ %>
  <font class="buttonson"><a
  href="LogonForm?langId=<%=languageId%>
  &storeId=<%=storeId%>&catalogId=
  <%=catalogId%>" style="color:
  #CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%
  ></a></font></td>
<%} else {%>
  <font class="buttonson"><a
  href="LogonForm?langId=<%=languageId%>
  &storeId=<%=storeId%>&catalogId=
  <%=catalogId%>&page=account" style="color:
  #CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%
  ></a></font></td>
```

Entre em contato conosco

Clicar em Entre em Contato Conosco chama o comando `ContactView`, que carrega a página Entre Em Contato Conosco (`contact.jsp`).

Ajuda Clicar em Ajuda chama o comando `HelpView`, que carrega a página de Ajuda (`help.jsp`).

Política de privacidade

Clicar em Política de Privacidade chama o comando `PrivacyView`, que carrega a página de Política de Privacidade (`privacy.jsp`).

Cabeçalho da Loja de Exemplo

As lojas de exemplo incluem um cabeçalho na parte superior de cada página (`header.jsp`) que inclui links para o seguinte:

- Carrinho de compras
- Minha conta
- Entre em contato conosco
- Ajuda
- Masculino
- Feminino
- Novidades

Comandos

`header.jsp` utiliza os seguintes comandos:

- `OrderItemDisplay`
- `LogonForm`
- `ContactView`
- `HelpView`
- `StoreCatalogDisplay`
- `CategoryDisplay`

Beans

header.jsp utiliza os seguintes beans:

- UserRegistrationDataBean
- CatalogDataBean
- CategoryDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte páginas da Loja de exemplo: técnicas comuns de implementação.

CARRINHO DE COMPRAS

O link **CARRINHO DE COMPRAS** no cabeçalho liga ao comando `OrderItemDisplay`, que retorna o comando de exibição `OrderItemDisplayViewShiptoAssoc`. O comando de exibição `OrderItemDisplayViewShiptoAssoc` está registrado com `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` carrega `shoppingcart.jsp` para exibir a página Carrinho de Compras.

MINHA CONTA

Se você é um cliente registrado, clicando em **MINHA CONTA** levará você à página Minha Conta (`account.jsp`). Se não for um cliente registrado, clicando em **MINHA CONTA** levará você à página Registrar ou Efetuar Logon (`myaccount.jsp`). Isto é realizado por meio do seguinte código:

```
if (userType.equalsIgnoreCase("G")){ %>
  <font class="buttonson"><a
  href="LogonForm?langId=<%=languageId%>
  &storeId=<%=storeId%>&catalogId=
  <%=catalogId%>" style="color:
  #CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%
  ></a></font></td>
<%} else {%>
  <font class="buttonson"><a
  href="LogonForm?langId=<%=languageId%>
  &storeId=<%=storeId%>&catalogId=
  <%=catalogId%>&page=account" style="color:
  #CCCC99"><%=infashiontext.getString("MY_ACCOUNT")%
  ></a></font></td>
```

ENTRE EM CONTATO CONOSCO

Clicar em **ENTRE EM CONTATO CONOSCO** chama o comando `ContactView`, que carrega a página Entre em Contato Conosco (`contact.jsp`).

AJUDA

Clicar em **AJUDA** chama o comando `HelpView`, que carrega a página de Ajuda (`help.jsp`).

Categorias de nível superior (Masculino, Feminino, Novidades)

Quando um cliente clica em uma das categorias de nível superior no cabeçalho, o comando `CategoryDisplay` é chamado. `CategoryDisplay` é registrado com `CategoryDisplay.jsp` no banco de dados. Quando o parâmetro `top` é definido como `Y`, como no exemplo a seguir, `topcategory.jsp` é carregado, exibindo a página de categoria apropriada, da seguinte maneira:

```
<a
href="CategoryDisplay?catalogId=<%=catalogId%>&storeId=<%=storeId%>
&categoryId=<%=category.getCategoryId()%>
&langId=<%=languageId%>&top=Y">
```

Página de Ajuda da Loja de Exemplo

A página Ajuda (`help.jsp`) é exibida quando um cliente clica em **Ajuda**.

Detalhes da implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Uma nova exibição chamada `HelpView` é criada na tabela `VIEWREG` e associada a `help.jsp`. A URL da página de privacidade é:
`http://machine_name/webapp/wcs/v5/stores/HelpView?parameter_list`.

Home page da loja de exemplo

A home page (`storecatalogdisplay.jsp`) age como a fachada da loja, trazendo os clientes para dentro de sua loja. A home page da loja de exemplo exibe todas as categorias de nível superior da loja, como Masculino e Feminino, e promove alguns itens especiais. Para obter mais informações sobre a home page da loja de exemplo, consulte o Caso de utilização da home page.

Comandos

`storecatalogdisplay.jsp` utiliza os comandos a seguir:

- `CategoryDisplay`
- `ProductDisplay`

Beans

`storecatalogdisplay.jsp` utiliza os beans a seguir:

- `CatalogDataBean`
- `CategoryDataBean`
- `ProductDataBean`
- `EMarketingSpotBean` (apenas `WebFashion`)

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte páginas da Loja de exemplo: técnicas comuns de implementação.

O `storecatalogdisplay.jsp` é lançado pela página `index.jsp` que fornece uma URL para chamar a home page da loja de exemplo. `index.jsp` chama o arquivo `parameters.jsp` que contém os parâmetros necessários para lançar a loja. `storecatalogdisplay.jsp` exibe o seguinte:

Categorias superiores

As categorias superiores são registradas na `CATTOGRP` tabela.

storecatalogdisplay.jsp recupera as categorias superiores com o método `getTopCategories()` localizado em `CatalogDataBean`.

Produtos especiais

Os produtos especiais contidos na InFashion e na NewFashion foram criados no catálogo, em vez de com o WebSphere Commerce Accelerator. Para criar as promoções, uma categoria especial de nível superior foi incluída no catálogo, com o identificador `HOMEPAGE_PROMO`. Os produtos pertencentes a esta categoria são, então, exibidos como produtos especiais apresentados.

Na WebFashion os produtos direcionados na home page fazem parte de uma campanha. A campanha é criada utilizando Pontos de e-Marketing e o WebSphere Commerce Accelerator. O Ponto de e-Marketing na home page é chamada `StoreHomePage` e é utilizada para criar promoções baseadas em sexo. Para ativar o Ponto de e-Marketing, é necessário criar uma campanha utilizando o WebSphere Commerce Accelerator.

Clientes convidados e clientes registrados que não forneceram informações sobre sexo verão os produtos padrão (`HOMEPAGE_PROMO`).

Nota: Produtos das categorias Promoções Domésticas ou Novidades aparecem somente nas categorias especiais. Depois das promoções terminarem, esses produtos precisam ser movidos de volta a suas categorias regulares.

Páginas de Login da Loja de Exemplo

As páginas de login da loja de exemplo permitem que clientes registrados efetuem login.

Quando clientes registrados clicam no link **Registre-se agora e receba aviso prévio de promoções!** na barra lateral, a página Registrar ou Login (`account.jsp`) é exibida. Os clientes então fornecem seus endereços de e-mail e senhas e clicam em **Login** e a página Minha Conta (`myaccount.jsp`) é exibida. Para obter mais informações, consulte Casos de utilização de Logon.

O procedimento de login na loja de exemplo utiliza os seguintes arquivos JSP:

- `account.jsp` (página Registrar ou Login)
- `myaccount.jsp` (página Minha Conta)
- `LoginForm.jsp` (Inclui os parâmetros do comando Logon. Não é exibida para o cliente.)
- `Logoff.jsp` (Inclui os parâmetros do comando Logoff. Não é exibida para o cliente.)
- `forgetpassword.jsp` (página Esqueceu sua Senha)
- `ResetPasswordForm.jsp` (Inclui os parâmetros do comando de redefinição da senha. Não é exibida para o cliente.)
- `password.jsp` (exibe uma mensagem aos clientes de que a senha foi definida para eles)
- `ChangePasswordForm.jsp` (página Alterar Senha)
- `ResetPasswordError.jsp` (chamada se houver um problema para redefinir a senha. Não é exibida para o cliente.)

Comandos

`account.jsp` utiliza os seguintes comandos.

- Logon
- Logoff

myaccount.jsp utiliza os seguintes comandos:

- UserRegistrationForm (InFashion, WebFashion, NewFashion e WebAuction)
- AddressBookForm (InFashion, WebFashion, NewFashion e WebAuction)
- InterestItemDisplay (WebFashion, NewFashion e WebAuction)
- ProfileFormView (WebFashion e WebAuction)
- TrackOrderStatus (WebFashion, NewFashion e WebAuction)

forgetpassword.jsp utiliza os seguintes comandos:

- Logoff
- ResetPassword

ChangePasswordForm.jsp utiliza os seguintes comandos:

- ResetPassword

forgetpassword_err.jsp utiliza os seguintes comandos:

- ResetPassword
- Logoff

password.jsp utiliza os seguintes comandos:

- LogonForm

Beans

forgetpassword.jsp utiliza os seguintes beans:

- ErrorDataBean

forgetpassword_err.jsp utiliza os seguintes beans:

- ErrorDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Depois que os clientes tiverem registrado na página Registrar ou tiverem digitado seus endereços de e-mail e senhas na página Login (account.jsp), os valores serão convertidos para letras minúsculas utilizando o seguinte código:

```
function prepareSubmit(form)
{
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
form.submit()
}
```

account.jsp também define os campos esperados pelo comando Logon, por exemplo:

```
<INPUT TYPE="hidden" NAME="URL"
VALUE="LogonForm?page=account">
```

Quando os clientes clicam em **Login** a partir da página Registrar ou Login, o comando Logon é chamado. Logon é registrado com LogonForm.jsp no banco de dados. LogonForm.jsp utiliza o parâmetro page para determinar qual página (myaccount.jsp ou account.jsp) carregar.

```
String state = request.getParameter("page");
.
.
.
if (state == null)
{
incfile = "/" + storeDir + "/myaccount.jsp";

}
else if (state.equals("account"))
{
incfile = "/" + storeDir + "/account.jsp";

}
```

Se uma combinação correta de endereço de e-mail e senha tiver sido digitada, LogonForm.jsp carrega a página Minha Conta (myaccount.jsp). Se uma combinação de endereço de e-mail e senha incorreta tiver sido digitada, LogonForm.jsp recarrega a página Registrar ou Login (account.jsp).

Se os clientes esquecerem suas senhas, e clicarem em **Esqueceu sua Senha?**, o comando Logoff é chamado, com o parâmetro state=forgetpassword. O comando Logoff é registrado com Logoff.jsp no banco de dados. Logoff.jsp verifica o estado do parâmetro, conforme descrito abaixo:

```
if (state == null) { String [] arrstate = (String [])
request.getAttribute("state"); if (arrstate != null) state = arrstate[0]; }

if (state == null || state.length() == 0) { incfile = "/" + storeDir +
"/UserRegistrationForm.jsp";

} else if (state.equals("forgetpassword")) { incfile = "/" + storeDir +
"/forgetpassword.jsp";

}
```

A página Esqueceu sua Senha (forgetpassword.jsp) é carregada se o estado é igual a forgetpassword. Quando um cliente preencher os campos na página e clicar em **Envie-me minha senha**, o comando ResetPassword será chamado. A senha antiga é definida como expirada no banco de dados e a nova senha é enviada por e-mail aos clientes. Quando os clientes efetuam login utilizando a nova senha, eles são forçados a alterar a senha e são levados à página Alterar Senha (ChangePasswordForm.jsp).

Nota: Se as senhas dos clientes estiverem definidas para expirar, eles serão automaticamente levados para a página Alterar Senha da próxima vez que efetuarem login.

Tratamento de erros

Se os clientes fornecerem um endereço de e-mail ou senha incorreto ou não preencherem os campos, uma mensagem de erro será exibida e o comando Logon recarrega a página Registrar ou Login sem definir o parâmetro page. Se uma senha incorreta tiver sido digitada, os clientes terão de aguardar por alguns segundos antes de efetuar login ou o seguinte erro será exibido:

Aguarde alguns segundos antes de tentar efetuar login novamente.

Página de privacidade da loja de exemplo

A página de Privacidade (`privacy.jsp`) é exibida quando um cliente clica em **Política de Privacidade**.

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Uma nova exibição chamada PrivacyView é criada na tabela VIEWREG e associada a `privacy.jsp`. A URL para exibir a página de privacidade é `http://machine_name/webapp/wcs/v5/stores/PrivacyView?parameter_list`.

Nota: Você deve criar sua própria política de privacidade e incluí-la em sua loja online. Para obter mais informações, consulte o site de privacidade da IBM `http://www.ibm.com/privacy/`.

Páginas de Produtos da Loja de Exemplo

Uma página de produto caracteriza um determinado produto em uma loja online. Geralmente, ela inclui uma descrição, preço e uma imagem e, se o produto tiver atributos (por exemplo, tamanhos e cores diferentes), permite que os clientes escolham um atributo.

Para obter mais informações sobre as páginas de produtos e como elas funcionam, consulte o Caso de utilização da página de exibição de produto.

Comandos

`productdisplay.jsp` utiliza os seguintes comandos:

- `OrderItemAdd`
- `InterestItemAdd` (apenas WebFashion e NewFashion)

Beans

`productdisplay.jsp` utiliza os seguintes beans:

- `CategoryDataBean`
- `ProductDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

ProductDisplay.jsp é registrado no banco de dados (na tabela DISPENTREL) para exibir todos os produtos na loja. ProductDisplay.jsp exibe o seguinte:

- Descrição, imagem, atributos e valores de atributos do produto.
- Descrição resumida da categoria pai
- Link Incluir no carrinho de compras
- Caixa de texto Quantidade (apenas NewFashion)
- Link Incluir na lista de itens de interesse (apenas WebFashion e NewFashion)

Descrição, imagem, atributos e valores de atributos do produto.

A descrição e a imagem do produto são exibidas utilizando-se propriedades ProductDataBean.

Atributos de produtos são recuperados utilizando o método getAttributes() para ProductDataBean. Os valores para cada atributo são recuperados utilizando o método getDistinctAttributeValues() no AttributeAccessBean. Se o ProductDataBean localizar informações nessa coluna, a página Exibição do Produto exibirá a imagem de Hotmedia ao invés da imagem completa.

Descrição resumida da categoria pai

O ID da categoria pai é fornecido a páginas de produtos através do parâmetro parent_category_rn. Uma descrição resumida da categoria pai é recuperada por CategoryDataBean. Por padrão, CategoryDataBean obtém o ID da categoria do parâmetro categoryId. No exemplo a seguir, o nome do parâmetro é parent_category_rn e o ID da categoria é definido explicitamente:

```
String parentCategoryId =
request.getParameter("parent_category_rn"); parentCategory = new
CategoryDataBean (); parentCategory.setCategoryId(parentCategoryId);
com.ibm.commerce.beans.DataBeanManager.activate(parentCategory,
request);
```

Incluir no carrinho de compras

O link **Incluir no carrinho de compras** é implementado criando um formulário que chama o comando OrderItemAdd. Na InFashion e na WebFashion, a quantidade de produtos pedidos é definida como 1 por padrão, utilizando um campo oculto da seguinte maneira:

```
<input type="hidden" name="quantity" value="1">
```

Você pode substituir o campo oculto por uma caixa de texto para que os clientes possam informar uma quantidade diferente.

Na loja NewFashion, a quantidade de produtos pedidos é definida como 1 por padrão, utilizando um campo de texto da seguinte maneira:

```
<input type="text" name="quantity" value="1",size="2">
```

O campo de texto permite que os clientes digitem uma quantidade diferente:

Incluir no carrinho de compras e Incluir na lista de itens de interesse

Quando um cliente seleciona **Incluir no carrinho de compras** ou **Incluir na lista de itens de interesse**, o seguinte javascript é chamado.

```
<SCRIPT language="javascript"> function Add2ShopCart(form){
form.action='OrderItemAdd'
form.URL.value='OrderItemDisplay' form.submit() }
```

```
function Add2WishList(form){
form.action='InterestItemAdd'
form.URL.value='InterestItemDisplay'      form.submit()    }
</SCRIPT> Quando o cliente inclui no carrinho de compras, o comando
OrderItemAdd é chamado. Quando inclui na lista de itens de interesse, o
comando InterestItemAdd é chamado.
```

Nota: Você pode utilizar o WebSphere Commerce Accelerator para criar produtos. Ao criar produtos, você deve criar um preço para o produto, a fim de exibi-lo nas páginas do produto na loja de exemplo. Se você não criar um preço, deverá remover o método `getCalculatedContractPrice` do `ProductDataBean`.

Página de registro da loja de exemplo

A página de registro da loja de exemplo permite que os clientes se registrem na loja de exemplo. Quando um cliente se registra, ele deve fornecer seu nome e sobrenome, um endereço de e-mail e, em seguida, criar uma senha.

Quando os clientes clicam em **Registrar agora e comprar!** a página Registrar ou Logon é exibida. Eles clicam então em **Registrar** e a página Registro (`register.jsp`) é exibida.

Quando os clientes querem atualizar suas informações de registro, eles clicam em **Alterar informações pessoais** na página Minha Conta. A página Alterar Informações Pessoais (`edit_registration.jsp`) é exibida.

Comandos

`register.jsp` utiliza os seguintes comandos:

- `UserRegistrationAdd`
- `PrivacyView`

Beans

`register.jsp` utiliza os seguintes beans:

- `ErrorDataBean`

`edit_registration.jsp` utiliza os seguintes beans:

- `DemographicsAccessBean`
- `UserRegistrationDataBean`
- `ErrorDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de exemplos de loja, incluindo informações multiculturais, consulte Páginas de loja de exemplo: técnicas de implementação comuns.

Registro

O link **Registrar** na página Registrar ou Login (`account.jsp`) exibe o formulário de registro. O WebSphere Commerce utiliza o comando `UserRegistrationAdd` para criar um novo registro. Se o cliente efetuou logon, `UserRegistrationAdd` comporta-se como `UserRegistrationUpdate`. Ou seja, o cliente não pode registrar-se novamente e criar uma nova conta. Como a loja de exemplo permite que o usuário registre-se várias vezes, um

cliente que efetuou logon deve efetuar logoff antes de criar um novo registro. Para obter este efeito, o link Registrar é definido automaticamente como o comando Logoff para clientes que efetuaram logon e como a exibição UserRegistrationForm para clientes convidados. O comando Logoff chama automaticamente a tarefa LogoffView. A tarefa LogoffView é registrada com Logoff.jsp no banco de dados. Logoff.jsp verifica o parâmetro state da URL. Se estiver definido como forgetpassword, forgetpassword.jsp é carregado; caso contrário, UserRegistrationForm.jsp é carregado. A exibição UserRegistrationForm está associada ao arquivo UserRegistrationForm.jsp na tabela VIEWREG. UserRegistrationForm.jsp verifica o parâmetro "new" da URL. Se estiver definido como Y o novo formulário de registro será exibido incluindo register.jsp. Caso contrário, o formulário de registro atualizado será exibido incluindo edit_registration.jsp.

register.jsp contém um novo formulário de registro. O novo formulário de registro é enviado ao comando UserRegistrationAdd. Por padrão, o comando UserRegistrationAdd espera vários campos obrigatórios que não são requeridos para a loja de exemplo. Como resultado, esses campos são configurados como campos HTML ocultos e os valores são definidos para "-", como no exemplo a seguir:

```
<INPUT TYPE="hidden" NAME="personTitle" Value="-">
```

Nota: Na loja NewFashion, o par nome-valor não é submetido. Se o par nome-valor não for submetido, o comando UserRegistrationAdd não verificará o parâmetro URL. Quando os clientes registram-se e preenchem seus perfis, o perfil não pode corresponder a um dos perfis do WebSphere Commerce Accelerator. Se o perfil do WebSphere Commerce Accelerator tiver um desconto associado a ele, os clientes receberão esse desconto.

Endereço de e-mail como o login

As lojas de exemplo requerem que os clientes informem seus endereços de e-mail como IDs de logon, mas o WebSphere Commerce requer um ID como um login. Como solução, a loja de exemplo pede aos clientes que informem seus endereços de e-mail no formulário de registro. Em seguida, antes de enviar o formulário, os campos de e-mail e ID do usuário são definidos com o valor do endereço de e-mail utilizando o seguinte JavaScript:

```
function prepareSubmit(form) {  
form.<%=ECUserConstants.EC_ADDR_EMAIL1%>.value = form.<%=  
ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()  
form.<%=ECUserConstants.EC_UREG_LOGONID%>.value = form.<%=  
ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase() form.submit()  
}
```

Alterar informações pessoais

Clicar em **Alterar informações pessoais** chama o comando UserRegistrationForm, que está associado no banco de dados a UserRegistrationForm.jsp. Se o parâmetro new não for nulo, ele irá carregar register.jsp. Caso contrário, ele irá carregar edit_registration.jsp.

O comando UserRegistrationUpdate é utilizado em edit_registration.jsp para atualizar as informações de registro do usuário. Se o cliente não preenche os campos de senha ou verificação de senha, o sistema fornece a senha de registro, utilizando o seguinte código:

```

function prepareSubmit(form)
{
    form.<%= ECUserConstants.EC_ADDR_EMAIL1
    %>.value=form.<%=ECUserConstants.EC_UREG_LOGONID%>.value
    form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
    form.<%= ECUserConstants.EC_UREG_LOGONID%>.
        value.toLowerCase()
    if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.
        value.length == 0)
    {
        form.<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>.value =
            '<%=strPassword%>'
    }
    if (form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>.
        value.length == 0)
    {
        form.<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>
            .value =
            '<%=strPassword%>'
    }
    form.submit()
}

```

Se um cliente digitou anteriormente as informações de sexo e idade, os campos de idade e sexo estarão previamente preenchidos. O DemographicsAccessBean extrai as informações de sexo e idade do banco de dados.

Tratamento de Erros

Em qualquer erro, o comando UserRegistrationAdd chama UserRegistrationErrorView, que é registrado com UserRegistrationForm.jsp no banco de dados. Contudo, a mesma exibição de erro é invocada por UserRegistrationUpdate. Para diferenciar um novo formulário de registro de um formulário de edição, o novo formulário de registro contém um parâmetro oculto chamado new. Se esse parâmetro estiver presente, UserRegistrationForm.jsp incluirá register.jsp. Caso contrário, ele carregará edit_registration.jsp.

register.jsp é utilizado em condições normais e de erro. ErrorDataBean e verificação de erros são utilizados para determinar sob quais condições register.jsp está sendo executado. Se existir um erro, register.jsp exibirá a mensagem de erro.

Nota: Erros de senha são identificados através dos comandos UserRegistrationAdd e UserRegistrationUpdate, no entanto, eles chamam AuthenticationPolicyErrorView quando ocorre um erro de senha.

Página de Novidades da Loja de Exemplo

A página Novidades (newarrivals.jsp) é exibida quando o cliente clica em **Novidades**.

Comandos

newarrivals.jsp utiliza os seguintes comandos:

- ProductDisplay

Beans

`newarrivals.jsp` utiliza os seguintes beans:

- `CategoryDataBean`
- `ProductDataBean`
- `CatalogDataBean`
- `EMarketingSpotBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de exemplos de loja, incluindo informações multiculturais, consulte Páginas de loja de exemplo: técnicas de implementação comuns.

A página Novidades (`newarrivals.jsp`) destina-se a exibir uma lista de itens em promoção. As páginas Novidades contidas na InFashion foram criadas no catálogo. Assim, uma categoria de nível superior chamada Novidades foi criada e todos os produtos em promoção foram incluídos nessa categoria. Como a InFashion exibe o conteúdo dessa categoria de modo diferente das outras categorias de nível superior, o arquivo `newarrivals.jsp` é registrado no banco de dados (na tabela `DISPCGPREL`) para exibir essa categoria. Quando o comando `CategoryDisplay` é invocado com este ID de categoria, `newarrivals.jsp` é carregado para exibir os produtos designados como novidades.

A página `newarrivals.jsp` utiliza `CategoryDataBean` para exibir informações sobre categoria. Para a NewFashion e a InFashion, os produtos dessa página vêm de uma categoria especial com o identificador “O que há de Novo.” A lista de produtos em promoção é recuperada através do método `getProducts()` de `CategoryDataBean`.

Na WebFashion, a página Novidades exibe a campanha Super Liquidação. O objetivo dessa campanha é vender as principais tendências da estação. A campanha Super Liquidação utiliza uma iniciativa de venda sugestiva, destinada a clientes registrados, sugerindo itens com base na idade do cliente. Clientes registrados com mais de 29 anos vêm um conjunto de três produtos; clientes com menos de 29 anos vêm outro conjunto. Clientes convidados e clientes registrados que não forneceram informações sobre idade verão os produtos padrão. (Qual Promoção)

A campanha Super Liquidação é criada com o WebSphere Commerce Accelerator e um Ponto de e-Marketing. O Ponto de e-Marketing é chamada de `NewArrivalsPage` e fica posicionada na página Novidades. Para ativar o Ponto de e-Marketing, é necessário configurar uma campanha no WebSphere Commerce Accelerator.

Página de confirmação de pedido da loja de exemplo NewFashion

Depois de concluir o pedido na quarta etapa do processo de compra da loja de exemplo, o cliente clica em **Pedir Agora**. A página Confirmação do pedido (`confirmation.jsp`) é exibida, permitindo que o cliente exiba o número do pedido, o sub-total, o total de impostos, os encargos para envio, o total geral e o preço do pedido total. Se dois pedidos forem processados, as informações dos dois pedidos serão exibidas na página de confirmação.

Para obter mais informações, consulte o Caso de utilização registrar saída do Carrinho de compras.

Beans

O `confirmation.jsp` utiliza os seguintes beans:

- `PayStatusPMDataBean`
- `OrderDataBean`

Implementação

O `PayStatusPMDataBean` obtém as informações do pedido do sistema. O `OrderDataBean` obtém as informações, formata-as e exibe-as para o sistema.

Página Notificação de E-mail da WebFashion e da NewFashion

Depois de um cliente registrado ter concluído um pedido clicando em **Pedir Agora** (para obter mais detalhes consulte o Caso de utilização registrar saída do Carrinho de compras) e o pedido é autorizado pelo Payment Manager, o sistema envia um e-mail ao cliente, `OrderAuthorized.jsp`, informando que o pedido foi aceito e o pagamento está autorizado.

Na loja NewFashion, os seguintes avisos de e-mail são enviados:

- E-mail de pedido cancelado (`OrderCanceledNotification.jsp`). O pedido não pode ser concluído porque o cartão de crédito foi rejeitado.
- E-mail de notificação de envio (`ReleaseShipNotify.jsp`). O pedido foi enviado para o comprador.
- E-mail de envio do pedido (`OrderReceived.jsp`). Depois de um cliente registrado ter concluído um pedido clicando em **Pedir Agora**, o e-mail de submissão do pedido será enviado.

Beans

`OrderAuthorized.jsp` utiliza os seguintes beans:

- `StoreDataBean`
- `OrderDataBean`
- `OrderItemDataBean`
- `OrderBean`
- `StoreEntityDescriptionAccessBean`
- `AddressDataBean`

Nota: Os beans acima são utilizados apenas na loja NewFashion.

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

O sistema recupera as informações sobre o pedido do cliente utilizando o `OrderDataBean`. As informações da loja são recuperadas com o `StoreDataBean`. O `StoreEntityDescriptionAccessBean` fornece o nome da loja ao sistema. O `AddressDataBean` fornece o endereço de envio ao sistema.

Página de Resumo do Pedido da Loja de Exemplo

Na quarta etapa do processo de compra da loja de exemplo, a página Compra 4, Resumo do Pedido, (`orderdisplaypending.jsp`), o cliente pode rever as informações detalhadas do pedido, incluindo uma descrição dos itens comprados, assim como quantidade, preço unitário e preço total, endereço de envio e custos de envio. O cliente deve então concluir o pedido fornecendo informações sobre pagamento e clicando em **Pedir Agora**. Na loja NewFashion, a data de envio esperada é exibida.

Para obter mais informações, consulte o Caso de utilização registrar saída do carrinho de compras.

Comandos

`orderdisplaypending.jsp` utiliza os seguintes comandos:

- `OrderProcess`
- `PrivacyView`
- `ContactView`
- `MultiOrderProcess` (apenas NewFashion) (`MultiOrderProcess` está associado a `MultiOrderProcess.jsp`)

Beans

`orderdisplaypending.jsp` utiliza os seguintes beans:

- `OrderDataBean`
- `OrderItemDataBean`
- `AddressDataBean`
- `ErrorDataBean`
- `ShippingModeDescriptionAccessBean`
- `UsablePaymentTCListDataBean`

Detalhes de implementação

Apenas NewFashion

Quando os clientes clicam em **Pedir Agora**, o comando `MultiOrderProcess` é chamado. `MultiOrderProcess` é um comando de exibição registrado na tabela `VIEWREG` e está associado a `MultiOrderProcess.jsp`.

`MultiOrderProcess.jsp` executa `OrderProcess` várias vezes, dependendo do número de pedidos na página de resumo de pedidos. Na NewFashion, o fluxo da compra permite somente até dois pedidos na página de resumo de pedidos.

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Para todas as lojas de exemplo

Quando o cliente inicia o processo de compra clicando em **Registrar Saída** na página Carrinho de Compras, ele passa por uma série de páginas de compra. A quarta delas é a página Compra 4. Resumo do Pedido (`orderdisplaypending.jsp`). Esta página inclui um formulário que permite que os clientes enviem suas

informações de cartão de crédito. UsablePaymentTCListDataBean é utilizado para obter os nomes de cartão de crédito disponíveis do Payment Manager e a ação para o formulário é definida como OrderProcess. Depois que o formulário é enviado e se o processamento do pedido tem êxito, OrderOKView é chamado. O comando OrderOKView é registrado na tabela VIEWREG no banco de dados e é associado a confirmation.jsp, que exibe informações para confirmar o pedido. Na loja NewFashion, se dois pedidos forem processados, as informações dos dois pedidos serão mostradas na página de confirmação.

Se há um erro, DoPaymentErrorView é chamado. DoPaymentErrorView é associado no banco de dados a OrderDisplayPending.jsp. Assim, quando há um erro, a página Compra 4. Resumo do Pedido (orderdisplaypending.jsp) é exibida novamente com uma mensagem de erro.

Página Verificar Disponibilidade do Produto da Loja de Exemplo NewFashion

Na terceira etapa, parte 3a, do processo de compra da loja de exemplo, conforme descrito no Caso de utilização registrar saída do carrinho de compras, o cliente pode verificar a disponibilidade do produto e ver a data de envio esperada de cada item do pedido (ProductAvailability.jsp).

Nota: Essa página aparece somente se alguns dos itens do pedido estão disponíveis.

Se alguns dos itens não estão atualmente em estoque, os clientes podem escolher uma preferência de envio que permita dividir o pedido ou encomendar os itens. Os clientes também têm a opção de remover um item do pedido.

Comandos

ProductAvailability.jsp utiliza os seguintes comandos:

- ProductDisplay
- OrderItemDelete
- OrderPrepare
- OrderItemMove
- OrderItemDisplay

Beans

ProductAvailability.jsp utiliza os seguintes beans:

- OrderDataBean
- OrderItemDataBean
- CatalogEntryDataBean

Implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

A página Compra 3a, Verificar Disponibilidade do Produto (ProductAvailability.jsp) é exibida somente se alguns dos itens do pedido não

estiverem disponíveis. `AllocationCheck.jsp` verifica se algum dos itens não está disponível. Se todos os itens estiverem indisponíveis ou se todos estiverem disponíveis, encaminha para `OrderDisplayPending.jsp`; caso contrário, inclui `ProductAvailability.jsp`. `ProductAvailability.jsp` permite que os clientes vejam a data de disponibilidade de cada item do pedido. Se alguns itens do pedido não existirem no estoque, o cliente seleciona uma opção de envio no campo **Escolher preferência de envio** para o pedido.

Os clientes podem escolher três opções de envio:

1. Aguardar até que o pedido todo esteja pronto para envio.
2. Enviar os itens disponíveis no estoque agora e enviar o restante quando estiver disponível.
3. Enviar os itens disponíveis no estoque agora e deixar o restante no carrinho de compras.

Se o cliente escolher as opções 2 ou 3, o comando `OrderItemMove` divide o pedido em dois pedidos separados.

Páginas de exibição de pacote das lojas de exemplo WebFashion e NewFashion

Uma página de exibição de pacote apresenta grupos promocionais de produtos em uma loja online. Normalmente, ela inclui uma descrição, uma imagem do pacote, uma lista dos componentes que compõem o pacote e uma imagem de cada um, o preço do pacote e uma lista de atributos (tamanho e cor), se os produtos que compõem o pacote tiverem variações, e uma lista de valores para cada atributo (vermelho e azul para cor, e Grande e Extra-Grande para tamanho).

Os clientes podem incluir o pacote no carrinho de compras ou na lista de itens de interesse com um único clique. O pacote é exibido como um único item de linha no carrinho de compras, ao contrário do pacote configurável, que exibe uma linha separada para cada item do pacote configurável. Os produtos de um pacote não podem ser adquiridos separadamente.

Para obter mais informações sobre as páginas de exibição de pacotes da loja de exemplo e como elas funcionam, consulte o Caso de utilização da página de exibição de pacotes.

Comandos

`PackageDisplay.jsp` utiliza os seguintes comandos:

- `OrderItemAdd`
- `InterestItemAdd`

Beans

`PackageDisplay.jsp` utiliza os seguintes beans:

- `PackageDataBean`
- `CompositeProductDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

O `packagedisplay.jsp` é registrado no banco de dados (na tabela `DISPENTREL`) para exibir todos os produtos na loja. O `packagedisplay.jsp` exibe o seguinte:

- Descrição (incluindo a lista de componentes que compõem o pacote), imagem e preço do pacote
- Atributos e valores dos atributos dos produtos que compõem o pacote
- Links para Incluir no carrinho de compras e Incluir na lista de itens de interesse
- Caixa de texto Quantidade (apenas `NewFashion`)

Descrição (incluindo a lista de componentes que compõem o pacote), imagem e preço do pacote

A descrição, imagem e preço do pacote são recuperados com o `PackageDataBean`.

Atributos e valores dos atributos dos produtos que compõem o pacote

O `PackageDataBean` recupera `ProductDataBeans` para exibir as informações de cada produto no pacote. Para obter mais informações, consulte Páginas de produtos da loja de exemplo.

Para que o pacote seja exibido corretamente, é necessário utilizar o mesmo nome de parâmetro para o valor do atributo de cada produto no pacote. Por exemplo:

```
<SELECT NAME="attrValue" >
<!-- Exibir valores de atributos do produto !-->
<%
Object values[] = attribute.getDistinctAttributeValues();
for (int j = 0; j < values.length; j++)
{
%>
<option><%=values[j]%></option>
<% } %>
</select>
```

Incluir no carrinho de compras e Incluir na lista de itens de interesse

Para obter mais informações, consulte Páginas de produtos da loja de exemplo.

Caixa de texto Quantidade

Permite que os clientes especifiquem o número de pacotes a incluir no carrinho de compras ou lista de itens de interesse.

Quadro de Navegação Esquerdo da Loja de Exemplo

O quadro de navegação esquerdo (`sidebar.jsp`) nas lojas de exemplo permitem que os clientes selecionem o idioma em que a loja será exibida. Ele também inclui links para as páginas de Registro e Ajuda.

Para obter mais informações, consulte Caso de utilização da home page.

Comandos

sidebar.jsp utiliza os seguintes comandos:

- StoreCatalogDisplay
- LogonForm
- HelpView

Beans

sidebar.jsp utiliza os seguintes beans:

- StoreLanguageAccessBean
- LanguageDescriptionAccessBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente seleciona país/região e idioma na lista drop down ESCOLHA UM PAÍS/REGIÃO e clica em **IR!**, as páginas do catálogo da loja são recarregadas de acordo com o languageId, utilizando o seguinte código:

```
String storelangId = storeLang.getLanguageId();
.
.
.
<option value="<%=storelangId %%"
SELECTED><%=langDesc.getDescription()%></option>
```

O código JavaScript é criado dinamicamente para encadear os comandos StoreCatalogDisplay e SetCurrencyPreference. Isso é feito para efetuar o link de um idioma com uma moeda padrão. Quando o cliente seleciona um idioma, ele seleciona indiretamente a moeda padrão para o idioma e, como resultado, o comando OrderPrepare não tem de ser chamado na página shoppingcart.jsp. Isso é mostrado no código a seguir:

```
<select NAME="currency"> ... <option value="<%=
(String)currencyId.elementAt(iElementNum) %%" SELECTED> </option>
```

Segue um exemplo de um javascript gerado dinamicamente que efetua o link com um idioma, com base na moeda selecionada:

```
<SCRIPT language="javascript"> function ChangeLanguage(form) { if
(form.currency[0].selected == true) { form.URL.value =
"StoreCatalogDisplay?storeId=10151&catalogId=10151&langId=-1"; } if
(form.currency[1].selected == true) { form.URL.value =
"StoreCatalogDisplay?storeId=10151&catalogId=10151&langId=-5"; }
form.submit(); } </SCRIPT>
```

Quadro de navegação esquerdo da loja de exemplo NewFashion

O quadro de navegação esquerdo (sidebar.jsp) na loja NewFashion permite que os clientes selecionem o idioma e a moeda em que a loja será exibida e pesquisem itens no catálogo. O quadro de navegação também inclui links para as páginas de Registro, Ajuda, Pesquisa Avançada e Bate-papo ao Vivo com Assistência ao Cliente.

Nota: O link **Bate-papo ao Vivo com Assistência ao Cliente** somente pode ser visto se for ativado na loja por meio do Store Services.

Para obter mais informações, consulte Caso de utilização da home page.

Comandos

sidebar.jsp utiliza os seguintes comandos:

- StoreCatalogDisplay
- SetCurrencyPreferenceCmdImpl
- LogonForm
- HelpView
- CatalogSearchResultView
- AdvancedSearchView

Beans

sidebar.jsp utiliza os seguintes beans:

- SupportedLanguageAccessBean
- LanguageDescriptionAccessBean
- CurrencyDescriptionAccessBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas da loja de exemplo, inclusive informações multiculturais, consulte páginas da Loja de exemplo: técnicas comuns de implementação.

Quando o cliente seleciona um país/região e idioma na lista drop down ESCOLHA UM IDIOMA e clica em **IR!**, as páginas do catálogo da loja são recarregadas de acordo com o languageId. O código abaixo obtém os idiomas suportados da loja e os exibe na caixa suspensa:

```
<jsp:useBean id="supportedLanguageDataBean"
class="com.ibm.commerce.common.objects.SupportedLanguageAccessBean"
scope="page" />
<%
Enumeration enStoreLangList =
supportedLanguageDataBean.findByStore(new Integer(storeId));
while (enStoreLangList.hasMoreElements()) {
SupportedLanguageDataBean storeLang =
(SupportedLanguageDataBean) enStoreLangList.nextElement();
String storelangId = storeLang.getLanguageId();
//Obtém o nome de exibição do idioma no idioma
//atualmente selecionado pelo comprador.
LanguageDescriptionDataBean langDesc =
new LanguageDescriptionDataBean();
langDesc.setInitKey_languageId(languageId);
langDesc.setInitKey_descriptionLanguageId(storelangId);

//Se esse idioma estiver atualmente selecionado, seleciona-o
//na lista drop down.
if (languageId.equals(storelangId))
{
```

```

%>
<option value="<%= storelangId %>"
SELECTED><%=langDesc.getDescription()%></option>

```

Quando um cliente seleciona uma moeda na lista drop down ESCOLHA UMA MOEDA e clica em **IR!**, as páginas do catálogo da loja são recarregadas de acordo com supportedCurrencies. O código abaixo obtém as moedas suportadas e as exibe na caixa suspensa:

```

CurrencyManager cm = CurrencyManager.getInstance();
String [] supportedCurrencies = (String [])
cm.getSupportedCurrencies(cmdcontext.getStore());
for (int i = 0; i < supportedCurrencies.length; ++i)
{
CurrencyDescriptionDataBean currDesc = new CurrencyDescriptionDataBean();
currDesc.setInitKey_languageId(languageId);
currDesc.setInitKey_currencyCode(supportedCurrencies[i]);
String currency = (String) cmdcontext.getCurrency();
// pré-seleciona o valor apropriado na lista drop down.
if (currency.equals(supportedCurrencies[i]))
{
%>
<OPTION Value="<%=supportedCurrencies[i]%>"
SELECTED><%=currDesc.getDescription()%></OPTION>
<%

```

Quando o cliente digita uma palavra-chave no campo Pesquisar e clica em **IR**, o comando CatalogSearchResultView envia os critérios de pesquisa e a página resultlist.jsp é exibida com os resultados da pesquisa.

Se os clientes clicarem em **Bate-papo ao Vivo com Assistência ao Cliente**, será exibida uma janela popup na tela e eles poderão bater papo online em tempo real com um representante de atendimento ao cliente. Este link somente será exibido se este recurso estiver ativado no Store Services. O link **Bate-papo ao Vivo com a Assistência ao Cliente** está contida no corpo de um par de tags personalizadas (a tag <flow:ifEnabled feature="customerCare"> e a tag </flow:ifEnabled>) e pode ser ativado ou desativado com base na opção selecionada em Store Services. Enquanto a tag personalizada permanece no local, o Store Services pode ser utilizado para alternar automaticamente entre um site que tem suporte à colaboração e um que não tem, sem precisar modificar a página JavaServer. Para ativar ou desativar permanentemente o suporte à colaboração na página, as tags personalizadas e o link **Bate-papo ao Vivo com Assistência ao Cliente** podem ser removidos da página JavaServer clicando em **Aplicar Permanentemente**, na GUI do Store Services. Não é recomendável remover manualmente ou alterar as tags personalizadas ou o código contido nelas. Em vez disso, utilize o botão **Aplicar Permanentemente** no Store Services.

Nota: A tag personalizada que contém o link **Bate-papo ao Vivo com Assistência ao Cliente** não pode ser copiada para as páginas JavaServer de outras lojas. Essas tags destinam-se apenas a funcionar na loja que as contém originalmente. Para obter mais informações sobre colaboração, consulte os links relacionados abaixo.

Página Selecionar Endereço para Cobrança da Loja de Exemplo

Na primeira etapa do processo de compra da loja de exemplo, a página Compra 1. Selecionar Endereço para Cobrança (`billingaddress.jsp`), o cliente pode selecionar um endereço existente como o endereço para cobrança ou criar um novo endereço a ser utilizado como o endereço para cobrança.

Para obter mais informações, consulte o Caso de utilização registrar saída do carrinho de compras.

Comandos

O `billingaddress.jsp` utiliza os seguintes comandos:

- `AddressForm`
- `OrderItemDisplay`
- `OrderCopy`
- `AddBillAddressView`

Beans

O `billingaddress.jsp` utiliza os seguintes beans:

- `OrderDataBean`
- `AddressAccessBean`
- `OrderItemAccessBean`
- `ErrorDataBean`

Detalhes de implementação

Nota: Para obter informações sobre as técnicas de implementação comuns a todas as páginas da loja de exemplo, incluindo as informações multiculturais, consulte Páginas da loja de exemplo: técnicas de implementação comuns.

Quando um cliente inicia o processo de compra, clicando em **Registrar Saída** na página Carrinho de Compras, ele passa por uma série de páginas de compra. A primeira dessas páginas é a página Compra 1. Incluir Endereço para Cobrança (`billingaddress.jsp`). O `billingaddress.jsp` verifica se o cliente tem endereços existentes no catálogo de endereços. Se existirem endereços atualmente no catálogo de endereços, eles serão exibidos, permitindo que o cliente selecione um como endereço de cobrança.

O cliente também pode criar um novo endereço, clicando em **Criar novo endereço**. Clicar em **Criar novo endereço** chama o comando `AddressForm`, que está associado no banco de dados a `AddressForm.jsp`. `AddressForm.jsp` chama `address.jsp`, que carrega a página Incluir Endereço. `address.jsp` verifica o parâmetro `page` para determinar a próxima página a ser carregada. Se o valor `page` estiver definido como `billingaddress`, o valor da URL no formulário `AddressAdd` será definido como `OrderItemDisplay`. `OrderItemDisplay` chama `billingaddress.jsp`, que traz o cliente de volta para a página Compra 1. Selecionar Endereço de Cobrança quando o cliente clica em **Enviar**. Ele verifica o parâmetro `page` para determinar qual formulário de endereço carregar.

Se não houver endereços existentes no catálogo de endereços, o formulário Incluir endereço de cobrança será exibido, solicitando que o cliente digite um novo endereço. O formulário Incluir endereço de cobrança é gerado neste caso por

billingaddress.jsp. A ação para este formulário HTML é definida como AddBillAddressView, que está registrado na tabela VIEWREG. AddBillAddressView está associado a AddBillAddress.jsp. Quando o formulário com AddBillAddressView é enviado, AddBillAddress.jsp é chamado.

AddBillAddress.jsp executa os seguintes comandos:

- AddressAdd
- OrderCopy

Depois que AddressAdd é executado, AddBillAddress.jsp utiliza o ID de endereço retornado por AddressAdd como parte da entrada para o comando OrderCopy. Em seguida, OrderCopy atribui o ID de endereço para o endereço de cobrança do pedido atual e OrderItemDisplay.jsp é chamado. O valor do parâmetro page é definido como newshipaddress, de modo que OrderItemDisplay.jsp chama shipaddress.jsp.

Nota: Durante o registro, o WebSphere Commerce requer que um endereço seja criado. Como a loja de exemplo não pede um endereço durante o registro do cliente, alguns dos campos requeridos, como o address1, são definidos como unused (não utilizados). Ao verificar endereços, **billingaddress.jsp** verifica se o valor de address1 é unused. Se for, o endereço não será exibido.

Página Selecionar Endereço para Cobrança da Loja de Exemplo NewFashion

Na primeira etapa do processo de compra da loja de exemplo, a página Compra 1. Selecionar Endereço para Cobrança (billingaddress.jsp), o cliente pode selecionar um endereço existente como o endereço para cobrança ou criar um novo endereço a ser utilizado como o endereço para cobrança.

Para obter mais informações, consulte o Caso de utilização registrar saída do carrinho de compras.

Comandos

O billingaddress.jsp utiliza os seguintes comandos:

- AddressForm
- OrderItemDisplay
- OrderCopy
- AddressAdd

Beans

O billingaddress.jsp utiliza os seguintes beans:

- OrderDataBean
- AddressDataBean
- OrderItemDataBean
- ErrorDataBean

Detalhes de implementação

Nota: Para obter informações sobre as técnicas de implementação comuns a todas as páginas da loja de exemplo, incluindo as informações multiculturais, consulte Páginas da loja de exemplo: técnicas de implementação comuns.

Quando os clientes iniciam o processo de compra clicando em **Registrar Saída** na página Carrinho de Compras, passam por uma série de páginas de compra. A primeira dessas páginas é a página Compra 1. Selecionar Endereço de Cobrança (`billingaddress.jsp`). `billingaddress.jsp`, verifica se o cliente tem endereços existentes no catálogo de endereços. Se existirem endereços atualmente no catálogo de endereços, eles serão exibidos, permitindo que os clientes selecionem um como endereço de cobrança.

O comando `OrderItemDisplay` é utilizado para determinar qual página será carregada em seguida. `OrderItemDisplay` retorna `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` inclui diferentes arquivos JSP baseados no parâmetro `page`. Se um valor para `pagefor shipmethod`, a terceira página, Compra 3. Selecionar Método de Envio (`shipping.jsp`), será carregada.

Os clientes também podem criar um novo endereço, clicando em **Criar novo endereço**. Clicar em **Criar novo endereço** chama o comando `AddressForm`, que está associado no banco de dados a `AddressForm.jsp`. `AddressForm.jsp` chama `address.jsp`, que carrega a página Incluir Endereço. A página Incluir Endereço leva o cliente de volta à página Compra 1, Incluir Endereço de Cobrança, quando ele clica em **Enviar**.

Se não houver endereços existentes no catálogo de endereços, o formulário Incluir endereço de cobrança será exibido, solicitando que o cliente digite um novo endereço. O formulário Incluir endereço para cobrança, neste caso, também é gerado pelo `billingaddress.jsp`.

Depois que `AddressAdd` é executado e chama o comando `OrderCopy`, `OrderCopy` atribui o ID de endereço ao endereço de cobrança do pedido atual e chama o comando `OrderItemDisplay` que chama `OrderItemDisplay.jsp`. O valor do parâmetro `page` é definido como `newshipaddress`, de modo que `OrderItemDisplay.jsp` chama `shipaddress.jsp`.

Nota: Nas lojas `InFashion` e `WebFashion`, o `AddBillAddress.jsp` é utilizado para chamar os comandos `AddAddress` e `OrderCopy`. Na loja `NewFashion`, `AddBillAddress.jsp` foi eliminado, encadeando os comandos `AddAddress` e `OrderCopy` com o parâmetro `URL`. Este método é mais fácil de implementar, pois não exige JSP extra, entretanto, o caminho de execução no sistema é mais longo, devido ao redirecionamento extra.

Carrinho de Compras da Loja de Exemplo

Os clientes podem exibir e editar os itens que selecionaram no carrinho de compras (`shoppingcart.jsp`), conforme descrito no Caso de utilização Exibir carrinho de compras.

Comandos

`shoppingcart.jsp` utiliza os seguintes comandos:

- `OrderItemUpdate`
- `OrderItemDelete`
- `StoreCatalogDisplay`

- QuickCheckoutView

Beans

shoppingcart.jsp utiliza os seguintes beans:

- OrderDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente clica em **CARRINHO DE COMPRAS** no cabeçalho ou rodapé, o comando `OrderItemDisplay` é chamado, retornando o comando de exibição `OrderItemDisplayViewShiptoAssoc`. O comando de exibição `OrderItemDisplayViewShiptoAssoc` está registrado com `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` inclui diferentes arquivos JavaServer Page com base no parâmetro `page`. Se não for fornecido um valor para `page`, a página Carrinho de Compras (`shoppingcart.jsp`) será carregada.

Nota: O `OrderItemDisplay.jsp` executa o `OrderPrepare`, que recalcula o total do pedido e converte a moeda padrão na moeda que está sendo utilizada pelo cliente. As lojas de uma única moeda não precisam executar o `OrderPrepare`.

A página Carrinho de Compras inclui um botão **Atualizar Totais** e um botão **Registrar Saída**. Clicar em **Atualizar Totais** atualiza a quantidade de itens pedidos e em seguida exibe novamente a página Carrinho de Compras. Clicar em **Registrar Saída** atualiza a quantidade de itens pedidos e, em seguida, exibe a página Compra 1: Selecionar Endereço de Cobrança.

Ambos os botões **Registrar Saída** e **Atualizar Totais** utilizam o mesmo formulário HTML. Contudo, quando um cliente clica em **Registrar Saída**, antes que o formulário seja enviado, JavaScript é utilizado para incluir um parâmetro `page` adicional com o valor definido como `billingaddress`.

O comando utilizado para concluir a opção de compra rápida é `QuickCheckoutView`, que é um comando de exibição registrado em `VIEWREG` e associado a `QuickCheckout.jsp`. `QuickCheckout.jsp` executa os seguintes comandos do servidor:

- `OrderItemUpdate` (atualizar endereço de envio)
- `OrderItemUpdate` (atualizar endereço de envio)
- `OrderCopy` (atualizar endereço para cobrança e informações de pagamento)
- `OrderPrepare`

`QuickCheckout.jsp` recupera os endereços de cobrança e de envio, o método de envio e as informações sobre pagamento do perfil de compra rápida do cliente, utilizando `OrderAccessBean`. Em seguida, atribui essas informações ao pedido especificado no `orderId` e executa o comando `OrderPrepare`.

Depois que o processo de compra rápida é concluído, o comando avança para uma exibição especificada na URL. A `WebFashion` especifica o `QuickCheckoutSummaryView` como a URL do comando `QuickCheckout`. Portanto, quando a compra rápida é concluída, a página de resumo da compra rápida é exibida.

Tratamento de erros

Se a loja não tiver um centro de distribuição anexado ou se o produto não estiver em estoque, o comando `OrderItemAdd/OrderItemUpdate` chama `ResolveFulfillmentCenterErrorView`, que está registrado com `shoppingcart.jsp` no banco de dados. Se os clientes digitarem um caractere inválido no campo de quantidade, `InvalidInputErrorView` é chamado. `InvalidInputErrorView` também está registrado com `shoppingcart.jsp` no banco de dados.

Como erros podem ser causados pelo comando `OrderItemUpdate` e pelo comando `OrderItemAdd`, `shoppingcart.jsp` verifica qual comando causou o problema e exibe um erro de acordo. Se o último comando for `OrderItemUpdate`, a página Carrinho de compras será exibida novamente com uma mensagem de erro. Caso contrário, uma página de erro separada será exibida com uma mensagem de erro. Isso é mostrado no código abaixo:

```
String lastCmdName = cmdcontext.getCommandName().trim();
```

O `shoppingcart.jsp` é utilizado tanto em condições normais como em condições de erro. O `ErrorDataBean` e a verificação de erro determina em que condição o `shoppingcart.jsp` está sendo exibido. Se existir um erro, o `shoppingcart.jsp` exibe uma mensagem de erro apropriada.

Se `QuickCheckout.jsp` (`QuickCheckoutView`) não puder encontrar o perfil de compra rápida, ele chama `QuickCheckoutError.jsp` para mostrar uma mensagem de erro. Como resultado, o `quickcheckouterrorview.jsp` é exibido.

O `quickcheckouterrorview.jsp` verifica se o cliente está registrado. Se o cliente estiver registrado, o sistema exibirá uma mensagem solicitando que o cliente crie um perfil de compra rápida. Caso contrário, o sistema solicita que o cliente faça o registro primeiro e, em seguida, crie um perfil de compra rápida.

Carrinho de compras da loja de exemplo NewFashion

Os clientes podem exibir e editar os itens que selecionaram no carrinho de compras (`shoppingcart.jsp`), conforme descrito no Caso de utilização Exibir carrinho de compras. Até dois pedidos podem estar presentes no carrinho de compras.

Comandos

`shoppingcart.jsp` utiliza os seguintes comandos:

- `OrderItemUpdate`
- `ProductDisplay`
- `StoreCatalogDisplay`
- `OrderItemMove`

Beans

`shoppingcart.jsp` utiliza os seguintes beans:

- `OrderItemDataBean`
- `ErrorDataBean`

- CatalogEntryDataBean
- OrderDataBean
- FormattedMonetaryAmountDataBean
- OrderAccessBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

O carrinho de compras apresenta-se sempre como um pedido, no entanto, pode haver vários pedidos no carrinho de compras que podem ter sido divididos. Isso ocorre se o cliente tiver ido ao fluxo de compras anteriormente e dividido o pedido. Quando o cliente clica em **Registrar Saída**, OrderItemMove é chamado para mover todos os pedidos para um pedido. OrderItemMove utiliza o parâmetro deleteIfEmpty para excluir todos os pedidos vazios deixados depois da movimentação.

Quando um cliente clica em **CARRINHO DE COMPRAS** no cabeçalho ou rodapé, o comando OrderItemDisplay é chamado, que está registrado com o OrderItemDisplay.jsp no banco de dados. OrderItemDisplay.jsp inclui diferentes arquivos JSP baseados no parâmetro page. Se não for fornecido um valor para page, a página Carrinho de Compras (shoppingcart.jsp) será carregada.

O comando OrderItemUpdate atualiza a quantidade de cada item no carrinho de compras. O FormattedMonetaryAmountDataBean formata o preço total e as quantidades. O OrderDataBean obtém os itens do carrinho de compras para cada pedido. Cada item do carrinho de compras é um OrderItemBean.

A página Carrinho de Compras inclui um botão **Atualizar Totais** e um botão **Registrar Saída**. Clicar em **Atualizar Totais** atualiza a quantidade de itens pedidos e em seguida exibe novamente a página Carrinho de Compras. Clicar em **Registrar Saída** atualiza a quantidade de itens pedidos e, em seguida, exibe a página Compra 1: Selecionar Endereço de Cobrança.

Nota: Os botões **Registrar Saída** e **Atualizar Totais** utilizam o mesmo formulário HTML.

Tratamento de erros

Se a loja não tiver um centro de distribuição conectado a ela, ErrorDataBean fornecerá informações de erro. ErrorDataBean também fornece informações se o cliente digita uma quantidade inválida, como um caractere não numérico.

O shoppingcart.jsp é utilizado tanto em condições normais como em condições de erro. O ErrorDataBean e a verificação de erro determina em que condição o shoppingcart.jsp está sendo exibido. Se existir um erro, o shoppingcart.jsp exibe uma mensagem de erro apropriada. Para obter mais informações sobre o tratamento de erros, consulte os conceitos relacionados abaixo.

Página Método de Envio da Loja de Exemplo

Na terceira etapa do processo de compra da loja de exemplo, conforme descrito no Caso de utilização Registrar saída do carrinho de compras, o cliente deve selecionar um método de envio (`shipping.jsp`).

Comandos

`shipping.jsp` utiliza os seguintes comandos:

- `AddShipModeView`
- `OrderItemDisplay`

Beans

`shipping.jsp` utiliza os seguintes beans:

- `OrderBean`
- `OrderItemAccessBean`
- `ShipModeAccessBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente inicia o processo de compra clicando em **Registrar Saída** na página do carrinho de compras, ele passa por uma série de páginas de compra. O comando `OrderItemDisplay` é utilizado para determinar qual página será carregada em seguida. `OrderItemDisplay` retorna `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` inclui diferentes arquivos JSP baseados no parâmetro `page`. Se um valor para `pagefor shipmethod`, a terceira página, **Compra 3 Selecionar Método de Envio** (`shipping.jsp`), será carregada.

A página **Compra 3 Selecionar Método de Envio** (`shipping.jsp`) inclui um formulário que permite que os clientes selecionem um método de envio. A ação para o formulário é definida como `AddShipModeView`, que é registrada na tabela do banco de dados `VIEWREG` para associar a `AddShipMode.jsp`. Quando o formulário com `AddShipModeView` é enviado, `AddShipMode.jsp` é chamado.

`AddShipMode.jsp` executa os seguintes comandos:

- `OrderItemUpdate`
- `OrderPrepare`.

O comando `OrderItemUpdate` atualiza o item de pedido com o método de envio selecionado. Em seguida, o comando `OrderPrepare` é chamado para pré-processar o pedido. Depois, `OrderDisplay` é chamado para exibir a próxima página do processo de compra. Se o parâmetro `statusfor` definido como `P`, **Compra 4: Resumo do Pedido** (`OrderDisplayPending.jsp`) é exibida em seguida.

A página **Compra 3. Selecionar Método de Envio** (`shipping.jsp`) exibe a estrutura de custo e o tempo aproximado de entrega para cada método de envio. Essas informações são armazenadas na tabela `SHPMODEDSC` nos seguintes campos:

- `DESCRIPTION` armazena a descrição do método de envio.
- `SHPMODEDSC.FIELD1` armazena a descrição da estrutura de custo.

- SHPMODEDSC.FIELD2 armazena o tempo aproximado de entrega.

Se a despesa de envio for alterada no banco de dados, não deixe de alterar também a descrição na tabela SHPMODEDSC de forma que os valores atualizados sejam exibidos nessa página.

Nota: SHPMODEDSC.FIELD1 e SHPMODEDSC.FIELD2 foram criados utilizando os campos personalizados na tabela SHPMODEDESC.

Página Selecionar Método de Envio da Loja de Exemplo NewFashion

Na terceira etapa do processo de compra da loja de exemplo, conforme descrito no Caso de utilização Registrar saída do carrinho de compras, o cliente deve selecionar um método de envio (`shipping.jsp`). O cliente pode selecionar um método de envio separado para cada item no pedido.

Comandos

`shipping.jsp` utiliza os seguintes comandos:

- `OrderItemUpdate`
- Exibição do Produto
- `OrderItemDisplay`
- `AllocationCheck`

Beans

`shipping.jsp` utiliza os seguintes beans:

- `OrderDataBean`
- `ShippingDataBean`
- `OrderItemDataBean`
- `ShippingModeDescriptionDataBean`
- `CatalogEntryDataBean`

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando os clientes iniciam o processo de compra clicando em **Registrar Saída** na página Carrinho de Compras, passam por uma série de páginas de compra. O comando `OrderItemDisplay` é utilizado para determinar qual página será carregada em seguida. `OrderItemDisplay` retorna `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` inclui diferentes arquivos JSP baseados no parâmetro `page`. Se um valor para `pagefor shipmethod`, a terceira página, Compra 3. Selecionar Método de Envio (`shipping.jsp`), será carregada.

A página Compra 3. Selecionar Método de Envio (`shipping.jsp`) permite que os clientes selecionem um método de envio para cada item do pedido. Os clientes selecionam um método de envio no campo **Método de Envio** para cada item.

O comando `OrderItemUpdate` atualiza o item do pedido com o método de envio selecionado. O comando `OrderItemDisplay` é chamado para exibir a próxima página do processo de compra quando os clientes clicam em **Avançar**. Se alguns

dos itens não estão disponíveis, a página Compra 3a, Disponibilidade do Produto (ProductAvailability.jsp) é exibida em seguida. A página Disponibilidade do produto é direcionada a AllocationCheck.jsp, que determina para qual página ir, dependendo se há inventário disponível.

A página Compra 3. Selecionar Método de Envio (shipping.jsp) exibe a estrutura de custo para cada método de envio. Essas informações são armazenadas na tabela SHPMODEDSC nos seguintes campos:

- DESCRIPTION armazena a descrição do método de envio.
- SHPMODEDSC.FIELD1 armazena a descrição da estrutura de custo.

Se a despesa de envio for alterada no banco de dados, não deixe de alterar também a descrição na tabela SHPMODEDSC de forma que os valores atualizados sejam exibidos nessa página.

Nota: SHPMODEDSC.FIELD1 é um campo personalizado na tabela SHPMODEDESC.

Página Selecionar Endereço de envio da Loja de Exemplo

Na segunda etapa do processo de compra da loja de exemplo, a página Compra 2. Selecionar o Endereço de Envio (shipaddress.jsp), o cliente pode selecionar um endereço existente como endereço de envio, editar aquele endereço ou criar um novo endereço para ser utilizado como o endereço de envio.

Para obter mais informações, consulte o Caso de utilização registrar saída do Carrinho de compras.

Comandos

shipaddress.jsp utiliza os seguintes comandos:

- OrderItemUpdate
- AddressForm
- OrderItemDisplay

Beans

shipaddress.jsp utiliza os seguintes beans:

- OrderBean
- AddressAccessBean
- OrderItemDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente inicia o processo de compra clicando em **Registrar Saída** na página Carrinho de Compras, ele passa por uma série de páginas de compra. A segunda dessas páginas é a página Compra 2. Selecionar Endereço de Envio (shipaddress.jsp). O shipaddress.jsp, exibe todos os endereços existentes, permitindo que o cliente selecione um como endereço de envio.

O cliente também pode criar um novo endereço, clicando em **Criar novo endereço**. Clicar em **Criar novo endereço** chama o comando AddressForm, que está associado no banco de dados a AddressForm.jsp. AddressForm.jsp chama address.jsp, que carrega a página Incluir Endereço. address.jsp verifica o parâmetro page para determinar a próxima página a ser carregada. Se o valor page estiver definido como shipaddress, o valor da URL no formulário AddressAdd será definido como OrderItemDisplay. OrderItemDisplay chama shipaddress.jsp, que traz o cliente de volta para a página Compra 2. Selecionar Endereço de Envio quando o cliente clica em **Enviar**.

Nota: Durante o registro, o WebSphere Commerce requer que um endereço seja criado. Como a loja de exemplo não pede um endereço durante o registro do cliente, alguns dos campos requeridos, como o address1, são definidos como "-". Ao verificar endereços, **billingaddress.jsp** verifica se o valor de address1 é "-". Se for, o endereço não será exibido.

Página Selecionar Endereço de Envio da Loja de Exemplo NewFashion

Na segunda etapa do processo de compra da loja de exemplo, a página Compra 2. Selecionar o Endereço de Envio (shipaddress.jsp), os clientes podem selecionar um endereço existente como endereço de envio, editar esse endereço ou criar um novo endereço para ser utilizado como o endereço de envio. O endereço existente é selecionado escolhendo-se um pseudônimo sob o campo **Enviar para**. Cada item do pedido pode ter um endereço de envio separado.

Para obter mais informações, consulte o Caso de utilização registrar saída do carrinho de compras.

Comandos

shipaddress.jsp utiliza os seguintes comandos:

- OrderItemUpdate
- AddressForm
- AddressBookForm
- Exibição do Produto
- OrderItemDisplay

Beans

shipaddress.jsp utiliza os seguintes beans:

- OrderBean
- AddressDataBean
- OrderItemDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando os clientes iniciam o processo de compra clicando em **Registrar Saída** na página Carrinho de Compras, passam por uma série de páginas de compras. A segunda dessas páginas é Compra 2. Selecionar Endereço de Envio (shipaddress.jsp). shipaddress.jsp, que exibe todos os itens do pedido e

pseudônimos ligados a endereços existentes, permitindo que o cliente selecione um pseudônimo separado como endereço de envio.

O comando `OrderItemDisplay` é utilizado para determinar qual página será carregada em seguida. `OrderItemDisplay` retorna `OrderItemDisplay.jsp` no banco de dados. `OrderItemDisplay.jsp` inclui diferentes arquivos JSP baseados no parâmetro `page`. Se um valor para `pageforShipMethod`, a terceira página, `Compra 3. Selecionar Método de Envio (shipping.jsp)`, será carregada.

Os clientes podem criar novos endereços, clicando em **Criar novo endereço**. Clicar em **Criar novo endereço** chama o comando `AddressForm`, que está associado no banco de dados a `AddressForm.jsp`. `AddressForm.jsp` chama `address.jsp`, que carrega a página `Incluir Endereço`. A página `Incluir Endereço` leva o cliente de volta à página `Selecionar endereço de envio` quando ele clica em **Enviar**.

Os clientes também podem editar um endereço, clicando em **Editar catálogo de endereços**. Clicar em **Editar catálogo de endereços** chama o comando `AddressBookForm`, que está associado a `AddressBookForm.jsp` no banco de dados. `AddressBookForm.jsp` chama `AddressBook.jsp`, que carrega a página `Catálogo de Endereços`. Todos os endereços associados àquele cliente são exibidos. Em seguida, os clientes podem clicar em **Editar** sob o endereço escolhido. Clicar em **Editar** exibe a página `Atualizar endereço` onde as alterações podem ser feitas. Os clientes são levados de volta para a página `Catálogo de Endereços` onde clicam em **Enviar**. A página `Catálogo de Endereços` leva os clientes de volta para a página `Compra 2. Selecionar Endereço de Envio`, onde eles clicam em **Retornar para compra**.

Nota: Durante o registro, o WebSphere Commerce requer que um endereço seja criado. Como a loja de exemplo não pede um endereço durante o registro do cliente, os campos do endereço criado durante o registro estão vazios. Ao verificar os endereços, `shipaddress.jsp` verifica se o `LogonID` é igual a `NickName`. Se for, isso significa que o endereço é um endereço próprio, ou seja, o endereço criado durante o registro e o endereço não é exibido. Se for, o endereço não será exibido.

Página de resumo de compra rápida da loja de exemplo

Quando um cliente clica em **Compra Rápida** no Carrinho de compras, a página `Efetuar Compra Rápida: Resumo do Pedido (quickcheckoutsummary.jsp)` é exibida, com os detalhes do pedido e informações de envio, cobrança e pagamento preenchidas. O cliente deve então concluir o pedido clicando em **Pedir Agora**.

Para obter mais informações, consulte o Caso de utilização da compra rápida.

Comandos

`quickcheckoutsummary.jsp` utiliza os seguintes comandos:

- `OrderProcess`
- `OrderItemDisplay`
- `PrivacyView`
- `ContactView`

Beans

`quickcheckoutsummary.jsp` utiliza os seguintes beans:

- `OrderBean`

- AddressDataBean
- ErrorDataBean
- ProfileCassetteAccountDataBean
- OrderPaymentInfoAccessBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

A página de resumo de compra rápida comporta-se da mesma maneira que a Página de resumo do pedido da loja de exemplo, com as seguintes diferenças:

- Os campos de informações sobre pagamento são previamente preenchidos com informações do perfil de compra rápida.
- As informações sobre pagamento são recuperadas por meio do OrderPaymentInfoAccessBean.

Páginas de exibição de pedidos das lojas de exemplo WebFashion e NewFashion

Depois que os clientes registrados efetuam um pedido, eles podem exibir o status desse pedido a qualquer hora. Para exibir um pedido, o cliente clica em **Minha Conta** e depois, na página Minha Conta, clica em **Exibir pedidos**. A página Status do Pedido (`trackorderstatus.jsp`) é exibida com a lista de pedidos que o cliente efetuou. Para obter mais informações sobre um pedido específico, os clientes clicam no pedido. A página Detalhes do Pedido (`orderdetail.jsp`) é exibida.

Nota: Pedidos temporários, pendentes, com edição de CSR, com perfil de pedido rápido, com lista de requisições privada, com lista de requisições compartilháveis, sem inventário e cancelados não podem ser exibidos.

Para obter mais informações, consulte o Caso de utilização Exibir Pedidos.

Beans

`trackorderstatus.jsp` utiliza os seguintes beans:

- OrderDataBean
- PayStatusListPMDDataBean
- OrderAccessBean
- PriceDataBean

`orderdetail.jsp` utiliza os seguintes beans:

- OrderDataBean
- OrderItemDataBean

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Quando um cliente clica em **Exibir pedidos**, a página Status do Pedido (trackorderstatus.jsp) é exibida com uma lista de pedidos que o cliente efetuou. OrderDataBean recupera todos os pedidos efetuados pelo cliente, utilizando o seguinte código para fazer um loop por todos os pedidos da lista:

```
<jsp:useBean id="orderABFinder"
class="com.ibm.commerce.order.objects.OrderDataBean" scope="page" />
Enumeration ordersABList = orderABFinder.findByMemberForUpdate(userId);
```

OrderDataBean exibe as seguintes informações do pedido:

- Número do pedido
- Data do pedido
- Valor total
- Status do pagamento

OrderDataBean recupera o status do pagamento chamando o método .getStatus(). O método .getStatus() retorna uma cadeia de status de um caractere que representa o status de pagamento conforme visto na tabela abaixo.

Status	Descrição Breve	Significado
P	Pendente	O cliente pode modificar o Pedido.
I	Enviado	O cliente não pode modificar o Pedido.
W	Aprovação pendente	O comando de tarefa CheckOrderApproval indicou que alguns dos itens do pedido não estão aprovados.
N	Aprovação negada	O comando de tarefa CheckOrderApproval indicou que foi negada aprovação para alguns itens do pedido.
M	Autorização de pagamento pendente	Aguardando autorização de pagamento.
A	A autorização do pagamento requer avaliação	A autorização de pagamento encontrou uma circunstância incomum, como um aviso de verificação de endereço. A autorização de pagamento deve ser examinada e aceita, ou o pedido deve ser cancelado, utilizando a interface do usuário Gerenciamento de Pedido. Se a autorização é aceita, a interface do usuário altera o status do pedido para 'B' ou 'C', conforme apropriado.

B	Item encomendado	Pode ser necessário refazer a autorização de pagamento, pois o valor do Pedido pode mudar em razão da alocação de inventário de item encomendado.
C	Autorização de pagamento concluída	A autorização de pagamento foi concluída. Todos os itens de pedidos são alocados no inventário existente. O valor do pedido não muda.
E	Edição de CSR	Um representante de atendimento ao cliente está trabalhando com o pedido.
R	Liberado	Todos os itens do pedido foram liberados para atendimento.
S	Enviado	Todos os itens do pedido foram enviados.
D	Depositado	O pagamento foi capturado.
L	Sem inventário	Não há inventário disponível para um ou mais itens do pedido.
T	Temporário	Utilizado pela interface do usuário Gerenciamento de Pedido para fazer backup temporário de um pedido.
Q	Perfil de pedido rápido.	O pedido contém informações padrão do pedido que podem ser copiadas rapidamente para criar novos pedidos.
F	Pronto para atendimento remoto	O pedido está pronto para ser enviado para um sistema remoto para atendimento. Esse status é utilizado pelo recurso MQAdapter.
G	Atendimento remoto pendente	O pedido foi enviado para um sistema remoto para atendimento. Esse status é utilizado pelo recurso MQAdapter.
Y	Lista de requisição privada	O pedido é uma lista de requisição privada.
Z	Lista de requisição compartilhável	O pedido é uma lista de requisição compartilhável.
X	Cancelado	O pedido foi cancelado.

Nota: O Payment Manager verifica se o pagamento foi rejeitado. Se o status do pagamento não for rejeitado, uma mensagem será exibida, dependendo do que o método `.getStatus()` retornou.

```

if
(payStatusBean.getPaymentState(sOrderId).equalsIgnoreCase("PAYMENT_VOID")
||
payStatusBean.getPaymentState(sOrderId).equalsIgnoreCase("PAYMENT_DECLINED"))
{

```

Quando os clientes recuperam suas listas de pedidos, podem clicar em um pedido específico para obter mais informações (`orderdetail.jsp`). O `OrderDataBean` recupera todos os itens do pedido para o cliente e o `OrderItemDataBean` recupera os detalhes sobre cada item. O `OrderItemDataBean` recupera os seguintes detalhes:

- Preço do pedido
- Conteúdo do pedido
- Data estimada ou real de envio do pedido
- Número de rastreamento para cada item

Cada item pode consistir em várias peças enviadas em diferentes caixas e pode ter mais de um número de rastreamento. Vários itens podem compartilhar os mesmos números de rastreamento.

O `OrderDataBean` recupera a data de envio estimada ou real do pedido, conforme se vê no seguinte código:

```

orderDate = orderABFinder.findByOrderForUpdate(new
Long(orderId)).getActualShipDate();

```

Páginas da Lista de Itens Desejados das Lojas de Exemplo WebFashion e NewFashion

As páginas da lista de itens de interesse da loja de exemplo permitem que clientes registrados incluam itens em suas listas de itens de interesse (lista de interesses) e exibam e editam a lista (`interestitemdisplay.jsp`), incluindo produtos em seu carrinho de compras conforme desejarem. Os clientes também podem enviar sua lista de itens de interesse para a família e os amigos por e-mail (`sendwishlistmsg.jsp`). Na loja NewFashion, os clientes podem enviar uma mensagem pessoal com suas listas de itens de interesse.

Nota: A NewFashion permite que clientes convidados e clientes registrados incluam itens na lista de itens de interesse. A WebFashion permite apenas que clientes registrados incluam itens na lista de itens de interesse.

Quando os clientes recebem uma lista de itens de interesse por e-mail, eles podem ver a página da lista de itens de interesse (`sharedwishlist.jsp`). Essa página é igual ao `interestitemdisplay.jsp`, sem os botões **Enviar lista de itens de interesse** e **Remover item**. Os clientes podem selecionar um item para comprar e incluí-lo no seu carrinho de compras.

Para obter mais informações, consulte o Caso de utilização Exibir lista de itens de interesse e o Caso de utilização Incluir item na lista de itens de interesse.

Os procedimentos da lista de itens de interesse da loja de exemplo utilizam os seguintes arquivos JSP:

- `interestitemdisplay.jsp` (página Lista de itens de interesse)
- `sharedwishlist.jsp`
- `sendwishlistmsg.jsp`

Comandos

O `interestitemdisplay.jsp` utiliza os seguintes comandos:

- `OrderItemAdd`
- `SendWishListMsg`

O `sharedwishlist.jsp` utiliza os seguintes comandos:

- `OrderItemAdd`

O `sendwishlistmsg.jsp` utiliza os seguintes comandos:

- `SendMsgCmd`

Beans

O `interestitemdisplay.jsp` utiliza os seguintes beans:

- `InterestItemListDataBean`
- `InterestItemDataBean`
- `CatalogEntryAccessBean`
- `UserRegistrationDataBean`

O `sharedwishlist.jsp` utiliza os seguintes beans:

- `InterestItemListDataBean`
- `InterestItemDataBean`
- `CatalogEntryAccessBean`

O `sendwishlistmsg.jsp` utiliza os seguintes beans:

- `UserRegistrationDataBean` (apenas `WebFashion`)

Detalhes de implementação

Nota: Para obter informações sobre técnicas de implementação comuns a todas as páginas de loja de exemplo, incluindo informações multiculturais, consulte Páginas de loja de exemplos: técnicas de implementação comuns.

Incluir na lista de itens de interesse

Quando os clientes clicam em **Incluir na lista de itens de interesse** o comando `InterestItemAdd` é chamado. Para obter mais informações, consulte Páginas de produtos da loja de exemplo.

Antes de incluir o item na lista de itens de interesse, o sistema utiliza o comando `UserRegistrationDataBean` para verificar se o cliente está registrado na loja. Se o cliente não estiver registrado, o JSP inclui o `registerfirst.jsp`, que exibe uma mensagem solicitando que o cliente faça o registro e tente novamente.

Exibir a lista de itens de interesse

Quando os clientes clicam em **Exibir lista de itens de interesse**, a página Lista de itens de interesse (`interestitemdisplay.jsp`) é exibida com o conteúdo da lista de itens de interesse. O `InterestItemListDataBean` e o `InterestItemDataBean` recuperam as informações sobre o conteúdo da lista de itens de interesse.

Incluir no carrinho de Compras e Remover itens

Na página Lista de itens de interesse (`interestitemdisplay.jsp`), os clientes podem optar por incluir itens em seus carrinhos de compras ou

excluir itens da lista de itens de interesse. Quando um cliente clica em **Incluir itens selecionados no carrinho de compras**, o comando `OrderItemAdd` é chamado. Quando um cliente clica em **Remover item**, o comando `InterestItemDelete` é chamado.

Enviar lista de itens de interesse

Quando um cliente preenche os campos **Nome** e **Endereço de e-mail** e clica em **Enviar lista de itens de interesse**, o comando `SendWishListMsg` é chamado. O comando `SendWishListMsg` está registrado na tabela `VIEWREG` para exibir o `sendwishlistmsg.jsp`.

É necessário criar uma instância para que o comando `SendWishListMsg` possa ser executado. A `WebFashion` e a `NewFashion` criam uma instância utilizando o seguinte: `SendMsgCmd sendMsgCmd = (SendMsgCmd) CommandFactory.createCommand(cmdEntry);`

Por exemplo:

```
CommandRegistryEntry cmdEntry = CommandFactory.locateCommandEntry("com.ibm.commerce.messaging.commands.SendMsgCmd", new Integer(storeId)); SendMsgCmd sendMsgCmd = (SendMsgCmd) CommandFactory.createCommand(cmdEntry);
```

Você pode definir os parâmetros do comando de tarefa "sendMsgCmd" para enviar a mensagem imediatamente: `sendMsgCmd.sendImmediate();`

Nota: A `WebFashion` tem apenas um campo **Endereço de e-mail** de destinatário. A `NewFashion` tem um campo **E-mail** de destinatário, um campo **Nome** do remetente e um campo **Mensagem pessoal**. O campo **E-mail** do destinatário e os campos **Nome** do remetente são obrigatórios.

Tratamento de erros


Se o cliente não selecionou nenhum item na lista de itens de interesse e clicar em **Incluir itens selecionados no carrinho de compras**, uma mensagem de erro será exibida. O seguinte código executa essa ação:


```
function checkForm(form) { var hasItem var i, e hasItem = false for (i = 0; i < form.elements.length; i++) { e = form.elements[i] if (e.type == "checkbox") { if (e.checked) { hasItem = true break } } } if (hasItem) form.submit() else alert("<%=infashiontext.getString("SELECTITEMS")%>") } </script>
```

Casos de utilização da loja de exemplo








Os casos de utilização detalham o fluxo de cada interação de usuário na loja, como o registro ou a pesquisa de produtos. Um conjunto de casos de utilização, que detalham as interações das lojas de exemplo `InFashion`, `WebFashion` e `NewFashion` é fornecido na ajuda online. Esses casos de utilização podem ajudar a entender melhor o fluxo das lojas de exemplo e podem ser utilizados como um guia para criar casos de utilização para a sua própria loja.

Notas:

1. `WebFashion` é a versão da `Professional Edition` da loja `InFashion`. Páginas e recursos apenas da `WebFashion` são denotados por  .

2. As páginas listadas abaixo não denotadas por  podem conter alguns recursos da Professional Edition.

Os seguintes casos de utilização são fornecidos:

- Caso de utilização Home page
- Caso de utilização de registro
- Caso de utilização de logon
- Caso de utilização Gerenciar Conta Pessoal
- Caso de utilização Alterar informações pessoais
- Caso de utilização Exibir categoria de produto
- Caso de utilização Exibir página de produtos
-  Caso de utilização da página Exibir Pacote
-  Caso de utilização da página Exibir pacote configurável
- Caso de utilização Exibir carrinho de compras
- Caso de utilização Registrar saída do carrinho de compras
-  Caso de utilização Compra Rápida
- Caso de utilização Editar um endereço
- Caso de utilização Incluir novo endereço
-  Caso de utilização Exibir pedidos
-  Caso de utilização Incluir item na lista de itens de interesse
-  Caso de utilização Exibir lista de itens de interesse
-  Caso de utilização Criar Perfil de Compra Rápida

Para obter mais informações sobre como as páginas funcionam juntas, consulte o fluxograma de compras da loja de exemplo.

Capítulo 5. Casos de Utilização da Loja de Exemplo

Caso de utilização Incluir Novo Endereço (Business Edition)

Business Os clientes podem incluir um novo endereço no catálogo de endereços.

Ator

Cliente

Fluxo principal

O cliente clica em **Incluir um novo endereço**. O sistema exibe uma página com os seguintes campos:

- Pseudônimo (apenas NewFashion)
- Nome
- Sobrenome
- Endereço (composto de duas caixas de texto)
- Cidade
- Estado
- CEP ou Código Postal
- País/Região
- Número do telefone

O cliente digita as informações nos campos e clica em **Enviar**. O sistema inclui um novo endereço no catálogo de endereços (E1).

Fluxo alternativo

Nenhum

Fluxos de exceção

E1: Campo Obrigatório Ausente

Se algum dos campos seguintes estiver ausente, o sistema emite uma mensagem de erro.

- Nome
- Sobrenome
- Endereço
- Cidade
- Estado
- CEP ou Código Postal
- País/Região

Se o pseudônimo digitado já existir no catálogo de endereços do cliente, o sistema relatará uma mensagem de erro.

O caso de utilização resume a partir do início.

Caso de utilização Incluir item na lista de itens de interesse da WebFashion e da NewFashion

Os clientes podem exibir e editar os produtos que selecionaram para comprar através do carrinho de compras.

Na loja WebFashion, uma lista de itens de interesse permite que clientes registrados incluam produtos em uma lista que gostariam de pedir no futuro. Na loja NewFashion, os clientes não precisam ser registrados para incluir um item em uma lista de itens de interesse. Uma lista de itens de interesse pode ser enviada por e-mail para a família ou para os amigos, que podem então adquirir os itens como presente para o cliente. Uma lista de itens de interesse difere de um carrinho de compras por que, neste, os clientes planejam adquirir os produtos durante aquela sessão de compras.

Ator

Cliente registrado

Fluxo principal

O cliente exibe um produto, pacote ou pacote configurável (para obter mais informações, consulte o Caso de utilização da página Exibir produto, Caso de utilização da página Exibir pacote configurável e Caso de utilização da página Exibir pacote) e clica em **Incluir na lista de itens de interesse**. O sistema inclui o produto, pacote ou pacote configurável na lista de itens de interesse (E1) e exibe a página Lista de itens de interesse, conforme descrito no Caso de utilização Exibir lista de itens de interesse.

Fluxos de exceção

E1: Comprador convidado tenta incluir um item na lista de itens de interesse

Se um comprador convidado tentar incluir um item na lista de itens de interesse, o sistema exibirá a seguinte mensagem: Para salvar itens na lista de itens de interesse, registre-se na WebFashion e tente novamente.

Na loja NewFashion, os compradores convidados podem fazer inclusões na lista de itens de interesse sem registrar-se.

Caso de utilização Exibir categoria de produto

As páginas do grupo de catálogos exibem uma lista de subcategorias e produtos. Essas páginas ajudam os clientes a navegarem pelos produtos. As páginas iniciais do grupo de catálogos levam a áreas mais amplas e páginas subseqüentes estreitam a pesquisa.

Geralmente existem três tipos de páginas de grupos de catálogos:

- Páginas que exibem subcategorias em uma categoria pai
- Páginas que exibem produtos em uma subcategoria
- Páginas que exibem subcategorias e produtos

Ator

Cliente

Fluxo principal

As categorias de produtos de nível superior estão listadas na Home page. Por exemplo, Moda Masculina, Moda Feminina e Novidades. Quando o cliente clica em Moda Masculina ou Moda Feminina, o sistema recupera as subcategorias da categoria selecionada do banco de dados e exibe as informações da categoria correspondente. Quando o cliente clica no link Novidades, novos produtos são exibidos. As seguintes informações são exibidas para cada categoria e produto:

- Uma imagem em miniatura (apenas produtos)
- O nome de cada categoria ou produto
- Um descrição breve para identificar o produto.
- O preço do produto.

As páginas de categorias da NewFashion e da WebFashion também incluem um produto especial, no formato de um pacote ou um pacote configurável. O recurso especial inclui uma imagem e uma descrição resumida do pacote ou do pacote configurável. O produto especial da InFashion é somente um único produto.

O cliente clica na imagem em miniatura ou no nome. Em seguida, o sistema exibe a página correspondente (A1, A2, A3, A4).

Fluxo alternativo

A1: O cliente seleciona a categoria

Quando o cliente seleciona a categoria, o sistema exibe outra página de categorias, reiniciando este caso de utilização.

A2: O cliente seleciona o produto

Quando o cliente seleciona um produto, a página de produtos é exibida conforme descrito no Caso de utilização Exibir página de produtos.

A3: O cliente seleciona um pacote configurável (somente WebFashion e NewFashion)

Quando o cliente seleciona um pacote configurável, a página do pacote configurável é exibida conforme descrito no Caso de utilização da página Exibir pacote configurável.

A4 O cliente seleciona um pacote (somente WebFashion e NewFashion)

Quando o cliente seleciona um pacote, a página do pacote é exibida conforme descrito no Caso de utilização da página Exibir pacote.

Fluxos de exceção

Nenhum

Caso de Utilização Registrar Saída do Carrinho de Compras

Os clientes registram a saída e enviam um pedido quando pagam pelos produtos listados em seus carrinhos de compras.

Ator

Cliente

Fluxo principal

O cliente clica em **Registrar Saída** para iniciar o caso de utilização.

O sistema exibe a página Endereço de Cobrança. Se o cliente tiver pelo menos um endereço no catálogo de endereços, o sistema solicita que o cliente selecione um dos endereços no catálogo de endereços como o endereço de cobrança. O cliente pode selecionar um endereço ou clicar em **Criar novo endereço**. Se o cliente clicar em **Criar novo endereço**, um novo endereço é incluído através do Caso de utilização Incluir novo endereço. O cliente, seleciona então um endereço. O sistema configura o endereço selecionado como o endereço de cobrança para o pedido. Se o cliente não tiver nenhum endereço no catálogo de endereços, A2 Digital Endereço de Cobrança será executado.

O sistema exibe a página Endereço de Envio. Uma lista de endereços no catálogo de endereços é exibida. Se o cliente localizar o endereço de envio apropriado na lista, o cliente seleciona o endereço. Caso contrário, o cliente clica em **Criar Novo Endereço** e inclui um novo endereço utilizando o Caso de utilização Incluir novo endereço. O sistema configura o endereço selecionado como o endereço de envio do pedido.

O sistema exibe uma lista de métodos de envio adequados para o endereço de envio. As informações a seguir serão exibidas para cada método de envio:

- Descrição resumida
- Caso de envio:
 - Custo fixo de envio por pedido
 - Custo de envio por item pedido
- Prazo de entrega aproximado (apenas InFashion e WebFashion)

O cliente seleciona o método de envio. O sistema define o método selecionado como o método de envio do pedido.

Nota: Na loja NewFashion, o endereço de envio e o método de envio são especificados para cada item do pedido.

Apenas para NewFashion

Na loja de exemplo NewFashion, o sistema exibe a disponibilidade de cada item no pedido, assim como várias opções para envio dos itens com base na disponibilidade. Para cada produto no pedido, o sistema exibe:

- Quantidade
- Descrição resumida do produto
- Valores de atributos (por exemplo, tamanho: x = grande; cor: azul)
- Data de disponibilidade esperada com base no inventário futuro
- Botão Remover (para remover o item do pedido)

Se a quantidade solicitada para um item específico não estiver em estoque, o sistema divide o pedido em dois pedidos, um contendo os itens de pedido disponíveis e o outro contando as partes não disponíveis do inventário. O cliente escolhe uma das três Preferências de envio:

- Opção 1, aguardar até que o pedido inteiro esteja pronto para envio. O sistema mostra a data de disponibilidade esperada de todo o pedido.
- Opção 2, enviar os itens disponíveis agora e o restante mais tarde.
- Opção 3, enviar os itens disponíveis agora e deixar o restante no carrinho de compras para ser comprado em outra data.

O cliente, então, seleciona a preferência de envio para o pedido.

Nota: Se todos os itens estiverem disponíveis, o cliente não verá a data de disponibilidade de cada item.

Para obter mais informações sobre o subsistema de inventário da NewFashion, consulte Subsistema de inventário.

Para InFashion, WebFashion e NewFashion

Além de exibir as informações do produto, o sistema também exibe os detalhes das informações do pedido. Para cada item no pedido, o sistema exibe:

- Descrição resumida do produto
- Valores de atributos (por exemplo, tamanho: x = grande; cor: azul)
- Quantidade
- Preço unitário
- Preço total
- O sistema também exibe:
 - Endereço para cobrança
 - Endereço de envio
 - Subtotal (custo total de todos os produtos pedidos)
 - Descontos, se houver
 - Imposto total, se houver (vários impostos devem ser exibidos separadamente)
 - Encargos, se houver
 - Custo de envio incluindo o método de envio
 - Total geral (o valor que será cobrado do cliente)
- Data de envio estimada (exibida na parte superior) (apenas NewFashion)

Nota: Na loja NewFashion, o endereço de envio e o método de envio são exibidos para cada item do pedido.

O sistema solicita ao usuário que digite as informações sobre o cartão de crédito. São solicitadas as informações a seguir:

- Tipo de cartão de crédito (por exemplo, Visa(R) ou MasterCard(R))
- Número do cartão
- Mês de expiração
- Ano de expiração

O cliente digita as informações do cartão de crédito e clica em **Pedir Agora**. O sistema armazena as informações sobre pagamento (E1) e exibe uma página de confirmação com as seguintes informações sobre o pedido:

- Número do pedido
- Sub total (custo total de todos os produtos pedidos)
- Imposto total
- Envio
- Descontos, se houver
- Total geral

O cliente imprime a página para referência futura.

Fluxo alternativo

A1: Incluir endereço de cobrança

O cliente é solicitado a incluir um novo endereço. O cliente digita um endereço utilizando o Caso de utilização de novo endereço. O sistema configura o novo endereço como o endereço de cobrança para o pedido. O caso de utilização continua.

Fluxos de exceção

E1: Número de cartão de crédito inválido

O sistema verifica a validade do número do cartão de crédito. Se a verificação falhar, o sistema exibirá uma mensagem de erro informando: Número de cartão de crédito inválido

e solicita ao cliente que digite novamente as informações. O caso de utilização continua.

E2: Cartão de crédito expirado

Se a data de expiração do cartão de crédito for anterior à data atual, o sistema exibirá uma mensagem de erro. O caso de utilização volta ao início.

Nota: Se um item não estiver disponível, a Página Disponibilidade do produto será exibida para o cliente.

Caso de utilização da página Exibir Pacote Configurável da WebFashion e NewFashion

A página pacote configurável exibe detalhes sobre os produtos ou itens que compõem um pacote configurável.

Ator

Cliente

Fluxo principal

O cliente seleciona um pacote configurável na página de categorias. Em seguida, o sistema recupera as informações sobre o pacote configurável do banco de dados e exibe uma página com informações detalhadas sobre o pacote configurável selecionado. A página do pacote configurável é exibida com as seguintes informações:

- Descrição resumida
- Descrição longa
- Imagem em tamanho normal do pacote configurável
- Imagens dos produtos que compõem o pacote configurável
- Preço de cada produto no pacote configurável
- Uma lista de atributos (tamanho e cor), se os produtos que compõem o pacote configurável tiverem variações, e uma lista de valores para cada atributo (vermelho e azul para cor, e Grande e Extra-Grande para tamanho).

Em seguida, o cliente seleciona os valores apropriados de cada atributo e clica em **Incluir no carrinho de compras**. Na loja NewFashion, o cliente pode especificar a quantidade a incluir no carrinho de compras. O sistema inclui o pacote configurável no carrinho de compras (E-1) e exibe a página Carrinho de Compras, conforme descrito no Caso de utilização Exibir carrinho de compras.

Nota: A quantidade padrão é 1. Os clientes podem modificar a quantidade depois do pacote configurável ter sido incluído no carrinho de compras.

O cliente também pode optar por clicar em **Incluir na lista de itens de interesse**. Se o cliente o fizer, o sistema incluirá o pacote configurável na lista de itens de interesse e exibirá a lista, conforme descrito no Caso de utilização Incluir item na lista de itens de interesse.

Fluxo alternativo

Nenhum

Fluxos de exceção

Nenhum

Caso de utilização da página Exibir Pacote da WebFashion e NewFashion

A página pacote exibe detalhes sobre os produtos ou itens que compõem um pacote.

Ator

Cliente

Fluxo principal

O cliente seleciona um pacote na página de categorias. O sistema recupera as informações sobre o pacote do banco de dados e então exibe uma página com informações detalhadas sobre o pacote selecionado. A página do pacote é exibida com as seguintes informações:

- Descrição resumida
- Descrição longa
- Imagens dos produtos que compõem o pacote
- Preço do pacote
- Uma lista de atributos (tamanho e cor), se os produtos que compõem o pacote tiverem variações, e uma lista de valores para cada atributo (vermelho e azul para cor, e Grande e Extra-Grande para tamanho).

Em seguida, o cliente seleciona os valores apropriados de cada atributo e clica em **Incluir no carrinho de compras**. O sistema inclui o pacote configurável no carrinho de compras (E-1) e exibe a página Carrinho de Compras conforme descrito no Caso de utilização Exibir o carrinho de compras. Na loja NewFashion, o cliente pode especificar a quantidade a ser incluída no carrinho de compras. A quantidade padrão é um.

Nota: Como todos os itens de um pacote não podem ser vendidos separadamente, um pacote é contado como um item no carrinho de compras.

O cliente também pode optar por clicar em **Incluir na lista de itens de interesse**. Se o cliente o fizer, o sistema incluirá o pacote na lista de itens de interesse e exibirá a lista, conforme descrito no Caso de utilização Incluir item na lista de itens de interesse.

Fluxo alternativo

Nenhum

Fluxos de exceção

E1: Atributo não está selecionado

Para produtos com diversos atributos, o cliente precisa selecionar um valor para cada atributo. Se o cliente não selecionar um valor, o sistema selecionará um valor padrão e incluirá o pacote no carrinho de compras ou na lista de itens de interesse.

Caso de utilização Editar um Endereço (Business Edition)

 Os clientes podem editar endereços no catálogo de endereços.

Ator

Cliente

Fluxo principal

O cliente seleciona um endereço no catálogo de endereços para editar. O sistema recupera e exibe os detalhes do endereço selecionado:

- Pseudônimo (apenas NewFashion)
- Nome
- Sobrenome
- Endereço
- Cidade
- Estado
- CEP ou Código Postal
- País ou Região
- Número do telefone

Na NewFashion, o pseudônimo digitado pelo cliente é exibido na parte superior da página.

O cliente faz as alterações desejadas e clica em **Submeter**. O sistema atualiza o endereço (E1).

Fluxo alternativo

Nenhum

Fluxos de exceção

E1: O sistema notifica o cliente sobre as informações obrigatórias que estão faltando e solicita as informações que estão faltando que podem incluir o seguinte:

- Nome
- Sobrenome
- Endereço

- Cidade
- Estado
- CEP ou Código Postal
- País ou Região

Se o pseudônimo já existir no catálogo de endereços, uma mensagem de erro será exibida.

O cliente digita informações obrigatórias que estão faltando.

Caso de utilização Home page

A home page funciona como fachada da loja, trazendo os clientes para a loja.

Ator

Cliente

Fluxo principal

Um cliente digita a URL da loja no navegador da Web. O sistema exibe então a home page.

A home page inclui o seguinte:

- Uma barra de navegação com links para as seguintes páginas:
 - Home page
 - Página de informações de contato
 - Página de segurança e privacidade
 - Página do carrinho de compras
 - Página Registro
 - Página Contas
 - Página de ajuda
- Links para categorias principais ou de nível superior. Para cada categoria, as seguintes informações são exibidas:
 - Uma imagem
 - Links para produtos em promoção
 - Se o cliente tiver especificado o sexo ao registrar-se, os produtos destinados ao sexo do cliente serão exibidos. Caso contrário, um conjunto genérico de produtos será exibido.
 - Para cada produto, as seguintes informações são exibidas:
 - Uma imagem
 - Uma descrição resumida

O cliente clica na imagem. Em seguida, o sistema exibe a página correspondente (A1, A2).

Fluxo alternativo

A1: O cliente seleciona a categoria

Quando o cliente seleciona uma categoria, o sistema exibe uma página de categoria, conforme descrito no Caso de utilização Exibir Categoria do Produto.

A2: O cliente seleciona o produto

Quando o cliente seleciona um produto, a página de produtos é exibida conforme descrito no Caso de utilização Exibir página de produtos.

Caso de utilização de Logon

O processo de logon permite que compradores registrados acessem suas contas digitando seu nome de usuário e senha.

Ator

Cliente

Fluxo principal

O cliente seleciona **Minha Conta**. O sistema exibe então uma página com os seguintes campos:

- Endereço de e-mail
- Senha

Os clientes digitam as informações apropriadas nos campos acima e selecionam **Login**. O sistema assegura que os endereços de e-mail e senhas dos clientes estejam corretos e então permite que eles entrem em suas contas. Se os clientes esquecerem suas senhas, devem selecionar **Esqueceu sua Senha** e o fluxo alternativo A1 será executado.

Fluxo alternativo

A1: Esqueceu Sua Senha

Se os clientes esquecerem suas senhas, devem selecionar **Esqueceu sua Senha**. O sistema exibe então uma página solicitando que os clientes digitem seus endereços de e-mail. Os clientes digitam seus endereços de e-mail e clicam em **Enviar minha senha**. O sistema envia então a senha para o endereço de e-mail (E1).

Fluxos de exceção

E1: Não há endereço de e-mail correspondente no sistema.

Se o sistema não consegue localizar um cliente com o endereço de e-mail correspondente, a seguinte mensagem de erro é exibida: Não é possível localizar um cliente com esse e-mail. O caso de utilização é abortado.

Se o cliente tentar efetuar login com o mesmo nome de usuário e falhar 6 vezes, o cliente será bloqueado. Uma mensagem será exibida informando que a conta está bloqueada e que o cliente deve entrar em contato com um representante da loja para reativá-la.

Caso de utilização Gerenciar Conta Pessoal

Os clientes gerenciam sua conta através das páginas de conta.

Ator

Cliente

Fluxo principal

O cliente seleciona **Minha Conta**. O sistema exibe então a página Minha Conta com as seguintes opções:

- Alterar informações pessoais
- Editar meu catálogo de endereços
- Criar ou atualizar o Perfil de Compra Rápida (apenas WebFashion)
- Exibir a lista de itens de interesse
- Exibir pedidos

Se o cliente selecionar **Alterar informações pessoais**, o fluxo alternativo A1: Alterar e-mail e senha será executado.

Se o cliente selecionar **Editar meu catálogo de endereços**, o fluxo alternativo A2: Editar catálogo de endereços será executado.

Se o cliente selecionar **Criar ou atualizar perfil** (o perfil Compra Rápida), a página Perfil de Compra Rápida será exibida, conforme descrito no Caso de utilização Criar perfil de Compra Rápida.

Se o cliente selecionar **Exibir lista de itens de interesse**, a página Lista de itens de interesse será exibida, conforme descrito no Caso de utilização Exibir lista de itens de interesse.

Se o cliente selecionar **Exibir pedidos**, a página Status de Pedidos será exibida, conforme descrito no Caso de utilização Exibir pedidos.

Fluxo alternativo

A1: Editar e-mail e senha

O sistema solicita ao usuário que modifique o endereço de e-mail e a senha usando o procedimento descrito no Caso de utilização Alterar informações pessoais. O caso de utilização resume a partir do início.

A2: Editar catálogo de endereços

O sistema exibe uma página que lista todos os endereços já incluídos no catálogo de endereços. Ao lado de cada endereço há dois botões: **Editar** e **Excluir**. Abaixo da lista de endereços há um botão **Incluir um Novo Endereço**.

Se o cliente clica em **Excluir**, o sistema exclui o endereço correspondente do banco de dados e, em seguida, exibe uma mensagem indicando que o endereço foi excluído corretamente. O fluxo alternativo volta ao início.

Se o cliente clicar em **Editar**, o sistema solicitará que o cliente edite o endereço utilizando o procedimento descrito no Caso de utilização Editar endereço. O sistema exibe então uma mensagem confirmando que o endereço foi atualizado corretamente e o fluxo alternativo volta ao início.

Se o cliente clicar em **Incluir um novo endereço**, o sistema solicita que o cliente digite um novo endereço, utilizando o procedimento descrito em Caso de utilização incluir novo endereço. Este fluxo alternativo volta ao início.

Fluxos de exceção

Nenhum.

Caso de utilização Exibir Pedidos da WebFashion e NewFashion

Os clientes podem rastrear e exibir o status de seu pedido.

Ator

Cliente

Fluxo principal

O cliente clica em **Minha Conta** e em **Exibir Pedidos**. O sistema exibe a página Status do Pedido, que lista todos os pedidos feitos pelo cliente. Se o cliente não tiver feito nenhum pedido, E1: Nenhum pedido encontrado será executado.

As informações a seguir serão exibidas para cada pedido:

- Número do pedido (ligado à página Detalhes do Pedido)
- Data do pedido
- Status do pedido
- Valor total

Para obter mais informações sobre o pedido, o cliente pode clicar no número do pedido. O sistema exibe a página Detalhes do Pedido. Essa página exibe as seguintes informações sobre cada produto no pedido:

- Quantidade
- Descrição resumida do produto
- Os atributos (tamanho e cor) e os valores de cada atributo (vermelho e azul para cor e Grande e Extra-grande para tamanho)
- Preço unitário
- Preço total
- Data de envio de todo o pedido (apenas NewFashion)
- Números de ID de rastreamento (apenas NewFashion)

Fluxos de exceção

E1: Nenhum pedido encontrado

Se o sistema não puder localizar nenhum pedido associado a este cliente, exibirá uma mensagem de erro informando:
Nenhum pedido foi encontrado

O cliente poderá, então, clicar em **Retornar à Home Page** ou em **Minha Conta**.

Caso de utilização Exibir página de produtos

A página de produtos exibe mais detalhes sobre um produto.

Ator

Cliente

Fluxo principal

O cliente seleciona um produto na página de categorias. Em seguida, o sistema recupera as informações sobre o produto do banco de dados e exibe uma página com informações detalhadas sobre o produto selecionado. A página do produto é exibida com as seguintes informações:

- Nome do produto
- Descrição detalhada
- Preço
- Imagem em tamanho normal do produto
- Se o produto tem variações, uma lista de atributos (tamanho e cor) e uma lista de valores para cada atributo (vermelho e azul para cor e Grande e Extra Grande para tamanho).

Em seguida, o cliente seleciona os valores apropriados de cada atributo e clica em **Incluir no carrinho de compras**. O sistema inclui o item selecionado no carrinho de compras (E1) e o exibe, conforme descrito no Caso de utilização Exibir o carrinho de compras. Na loja NewFashion, o cliente pode especificar a quantidade a ser incluída no carrinho de compras. A quantidade padrão é 1.

Na WebFashion e na NewFashion, o cliente tem a opção de clicar em **Incluir na lista de itens de interesse**. Se o cliente o fizer, o sistema incluirá o pacote na lista de itens de interesse e exibirá a lista, conforme descrito no Caso de utilização Incluir item na lista de itens de interesse.

Fluxo alternativo

Nenhum

Fluxos de exceção

Nenhum

Caso de utilização Alterar informações pessoais

Os clientes podem alterar informações pessoais, como endereço de e-mail e senha.

Ator

Cliente

Fluxo principal

O cliente clica em **Alterar Informações Pessoais** (E1). O sistema exibe a página Alterar informações pessoais, que inclui os seguintes campos:

- **Nome**, preenchido previamente com o nome atual.
- **Sobrenome**, preenchido previamente com o sobrenome atual.
- **Endereço de e-mail**, preenchido previamente com o endereço de e-mail atual.
- **Senha**
- **Senha de verificação**
- **Idade**, preenchido previamente com a idade atual (apenas WebFashion)
- **Sexo**, preenchido previamente com o sexo (apenas WebFashion)
- **Idioma preferido** (apenas NewFashion)
- **Moeda preferida** (apenas NewFashion)

O cliente altera os campos conforme necessário. O campo da senha pode ser deixado vazio. Se o cliente decide deixar o campo da senha em branco, o sistema não modifica a senha atual. Em seguida, o cliente seleciona **Submeter**, e o sistema atualiza o endereço de email e a senha (E2). Se o campo da senha estiver vazio, o sistema deixará a senha anterior inalterada.

Fluxo alternativo

Nenhum

Fluxos de exceção

A senha digitada deve ter pelo menos 6 caracteres, conter pelo menos um dígito, conter pelo menos 1 letra e não deve ter o mesmo caractere repetido 5 vezes ou 4 vezes consecutivas ou uma mensagem de erro será exibida. Uma mensagem de erro também será exibida se a senha for a mesma que o endereço de e-mail ou se o cliente decidir alterar a senha, mas a senha for a mesma que a anterior. Para obter mais informações, consulte as tarefas relacionadas abaixo.

E1: Cliente não efetuou logon

Os clientes precisam efetuar logon para alterar seus endereços de e-mail e senhas. Se não efetuarem logon, o sistema solicitará que os clientes façam isso utilizando o procedimento descrito no Caso de utilização de Logon. Este caso de utilização volta então ao início.

E2: E-mail já existe para um comprador diferente

O endereço de e-mail informado não pode corresponder ao endereço de e-mail de qualquer outro cliente existente. Se o sistema encontrar uma correspondência, ele solicita que o cliente selecione um endereço de e-mail diferente. O caso de utilização volta ao início.

E3: Não é possível verificar senha

A senha informada deve corresponder à senha especificada como senha de verificação. Caso contrário, o sistema exibe a seguinte mensagem de erro: A senha não corresponde à senha de verificação, tente novamente. O caso de utilização volta ao início.

Caso de utilização do registro

O processo de registro permite que os clientes digitem informações pessoais no banco de dados.

Ator

Cliente

Fluxo principal

O cliente seleciona **Registrar**. O sistema exibe uma página com os seguintes campos:

- E-mail
- Senha
- Senha de verificação
- Nome
- Sobrenome
- Idade (opcional) (apenas WebFashion)

- Sexo (opcional) (apenas WebFashion)
- Idioma preferido (apenas NewFashion)
- Moeda preferida (apenas NewFashion)

O cliente digita as informações apropriadas nos campos acima e seleciona **Enviar**. O sistema cria um novo cliente e salva as informações desse cliente (E1, E2, E3).

O sistema solicita que os clientes gerenciem suas contas seguindo o processo no Caso de utilização Gerenciar Conta Pessoal.

Fluxo alternativo

Nenhum.

Fluxos de exceção

E1: O endereço de e-mail já existe

Se o endereço de e-mail já existe no sistema, o sistema exibe uma mensagem de erro solicitando que o usuário digite outro endereço de e-mail. O caso de utilização volta ao início.

E2: Estão faltando campos obrigatórios

Se qualquer um dos campos a seguir (E-mail, Senha, Verificação da senha, Nome, Sobrenome) não estiver preenchido, o sistema emitirá uma mensagem de erro. O caso de utilização volta ao início.

E3: Senha inválida

Se a senha não corresponder à senha de verificação, o sistema emite um aviso.

Nota: A senha digitada deve ter pelo menos seis caracteres, conter pelo menos um dígito e uma letra e não deve conter o mesmo caractere repetido cinco vezes ou quatro vezes consecutivas ou uma mensagem de erro será exibida. Uma mensagem de erro também é exibida se a senha for a mesma que o endereço de e-mail.

Caso de utilização Exibir carrinho de compras

Os clientes podem exibir e editar os produtos que selecionaram para comprar através do carrinho de compras.

Ator

Cliente

Fluxo principal

O cliente clica em **Carrinho de compras**. O sistema exibe então uma página mostrando o conteúdo do carrinho de compras. Uma lista dos produtos que estão no carrinho é exibida (E1). Para cada produto, as seguintes informações são exibidas:

- Descrição resumida do produto
- Pares de nome do atributo e valor do atributo do item, como tamanho e grande.
- Quantidade
- Preço unitário

- Preço total

O sistema também exibe o custo total do pedido antes do imposto e do envio.

Se o cliente clicar em **Atualizar totais**, A1: Atualizar totais será executado.

Se o cliente clicar em **Remover item**, A2: Remover um item do carrinho será executado.

Se o cliente clica em **Registrar Saída**, A3: Atualizar carrinho e registrar saída é executado.

Se o comprador clicar em **Retornar às Compras**, o sistema exibirá a mesma página da qual o comprador chegou a página do carrinho de compras. O caso de utilização termina.

Se o cliente clicar em **Compra Rápida**, o sistema exibirá a página **Compra Rápida: Resumo do pedido**, conforme descrito no Caso de utilização da **Compra Rápida**

Sub fluxos

A1: Atualizar totais

Para atualizar a quantidade de qualquer item, o cliente digita a nova quantidade na caixa de texto e clica em **Atualizar totais**. O sistema atualiza a quantidade de itens no carrinho de compras. Se a nova quantidade de um item for 0, o item será removido do carrinho.

A2: Remover um item do carrinho

Para excluir um item do carrinho, o cliente clica em **Remover item** para o item. O sistema remove o item do carrinho.

A3: Atualizar carrinho e registrar saída

O sistema atualiza a quantidade dos itens no carrinho de compras e inicia o processo de compra. Consulte o Caso de utilização **Registrar saída do Carrinho de compras**, Caso de utilização **registrar saída do Carrinho de compras**.

Fluxos de exceção

E1: Carrinho de compras vazio

Se não houver itens no carrinho de compras, o sistema exibe a seguinte mensagem: Seu carrinho está vazio.

Se um número inválido for digitado no campo quantidade, como um caractere, a página carrinho de compras será exibida novamente com uma mensagem de erro.

Caso de utilização da página Exibir Lista de Itens de Interesse da WebFashion e NewFashion

Os clientes podem exibir e editar os produtos que incluíram na lista de itens de interesse (ou lista de interesses).

Ator

Cliente

Fluxo principal

Os clientes clicam em **Minha Conta** e, em seguida em **Exibir Lista de Itens de Interesse**. Em seguida, o sistema exibe uma página que mostra o conteúdo da lista de itens de interesse. Uma lista de produtos na lista de itens de interesse será exibida (E1). Para cada produto, pacote ou pacote configurável na lista de itens de interesse, as seguintes informações serão exibidas.

- Descrição resumida do produto
- Atributo e valor do item, por exemplo, valores de tamanho e cor
- Preço unitário
- Link **Remove**

O sistema também exibe os seguintes itens na página:

- Botão **Retornar às Compras**.
- Botão **Incluir itens selecionados no carrinho de compras**.
- Botão **Enviar lista de itens de interesse**
- Campo **Endereço de e-mail** (Para: / destinatário)
- Campo **Mensagem de e-mail** (apenas NewFashion)
- **De:/nome do submissor**(apenas NewFashion)
- **Endereço de e-mail**(apenas NewFashion)

Para excluir um item da lista de itens desejado, o cliente clica em **Remove**. O sistema completa A1, Remove item.

Para enviar a lista de itens de interesse para os amigos e para a família, os clientes digitam os endereços de e-mail dos destinatários no campo **Endereço de e-mail** e clicam em **Enviar lista de itens de interesse**. O sistema completa A2, Enviar lista de itens de interesse. Na loja NewFashion, os clientes podem enviar mensagem de e-mail utilizando o campo **Mensagem de e-mail**. Os clientes precisam especificar seus nomes e podem especificar, como opcional, seus endereços de e-mail. Os clientes registrados têm o campo **Nome** preenchido anteriormente.

Para incluir itens da lista de itens de interesse no carrinho de compras, os clientes selecionam cada item que gostariam de incluir e clicam em **Incluir itens selecionados no carrinho de compras**. O sistema completa A3, Incluir no carrinho.

Fluxo alternativo

A1: Remove item

Para excluir um item do carrinho, os clientes clicam em **Remove** para o item. O sistema remove o item da lista de itens de interesse. O caso de utilização volta ao início.

A2: Enviar lista de itens de interesse

Para enviar a lista de itens de interesse para os amigos e para a família, os clientes digitam os endereços de e-mail dos destinatários no campo **Endereço de e-mail**. Para enviar a lista de itens de interesse a mais de um amigo, cada endereço de e-mail precisa estar separado por uma vírgula, por exemplo: info@infashion.com, wcs@infashion.com. Em seguida, os clientes clicam em **Enviar lista de itens de interesse**.

O sistema compõe uma mensagem de e-mail contendo as seguintes informações:

- instruções sobre como adquirir os itens da lista de itens de interesse;

- um link para a lista de itens de interesse, que contém as seguintes informações:
 - descrição resumida de cada produto, pacote ou pacote configurável incluído na lista;
 - um link para a página de exibição do produto, pacote ou pacote configurável;
 - preço
 - o botão **Incluir itens selecionados no carrinho de compras**;

Na loja NewFashion, os clientes podem enviar uma mensagem de e-mail personalizada junto com a mensagem do sistema, digitando a mensagem no campo **E-mail**.

A3: Incluir no carrinho

Para incluir um item da lista de itens de interesse no carrinho de compras, os clientes selecionam cada item a ser incluído e clicam em **Incluir itens selecionados no carrinho de compras**. O sistema inclui todos os itens selecionados no carrinho de compras e exibe o carrinho de compras, conforme descrito no Caso de utilização Exibir carrinho de compras.

Fluxos de exceção

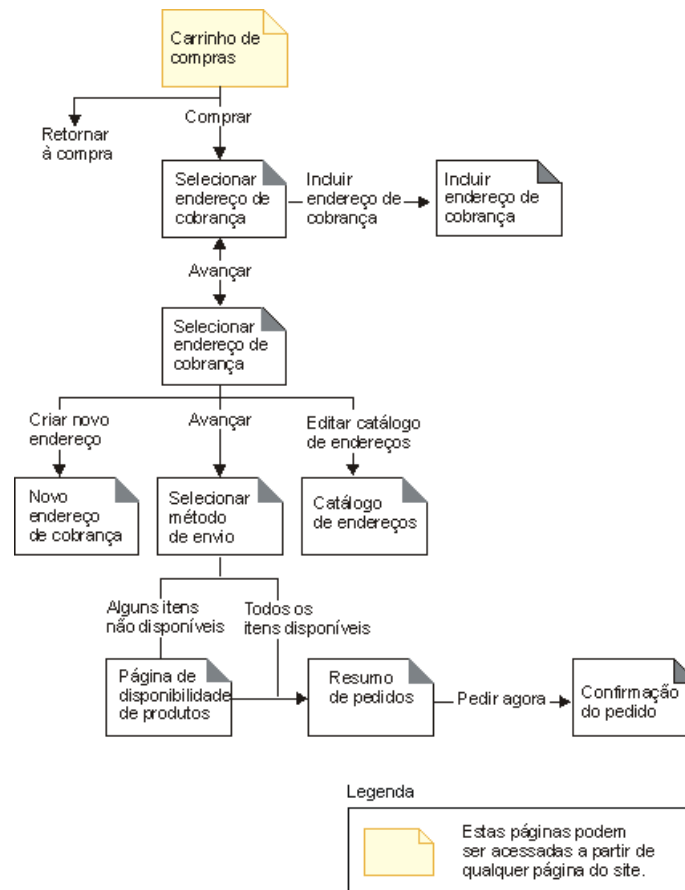
E1: Lista de itens de interesse vazia

Se não houver itens na lista de itens de interesse, o sistema exibirá a seguinte mensagem: Lista de itens de interesse vazia.
O caso de utilização termina.

Capítulo 6. Fluxo de Compra NewFashion

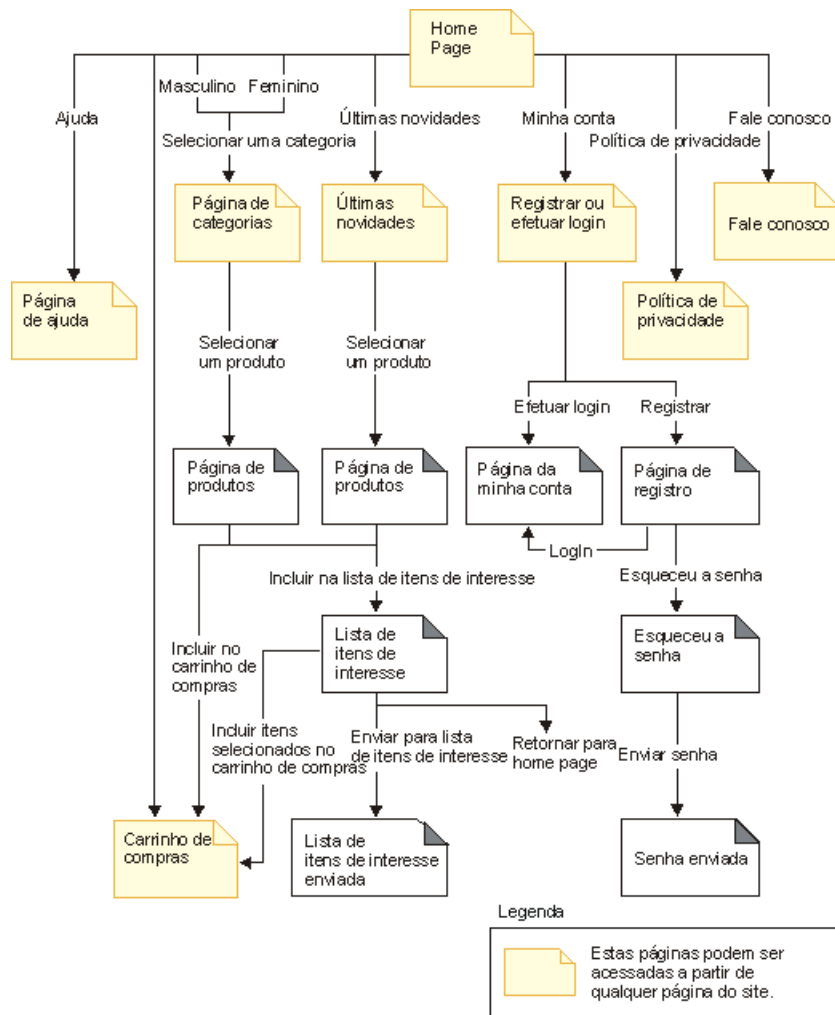
Fluxo do Carrinho de Compras da Loja de Exemplo NewFashion

O diagrama a seguir detalha o fluxo do carrinho de compras da NewFashion. Para obter mais informações sobre o fluxo de compras completo, consulte as referências relacionadas abaixo.



Fluxo do Carrinho de Compras da Loja de Exemplo NewFashion

O diagrama a seguir detalha as páginas iniciais do fluxo de compras da NewFashion. Para obter mais informações sobre os fluxos do carrinho de compras e de minha conta, consulte a seção referências relacionadas abaixo.



Avisos

Qualquer referência a um programa licenciado da IBM neste documento não significa que apenas o programa licenciado da IBM possa ser utilizado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado. A avaliação e verificação da operação em conjunto com outros produtos, com exceção daqueles expressamente indicados pela IBM, é responsabilidade do usuário.

A IBM pode ter patentes ou solicitações de patentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. pedidos de licença devem ser enviados, por escrito, à:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138 - 146 / Botafogo
CEP 22290-240
Rio de Janeiro - RJ

Os possuidores de licença deste programa que pretendam obter mais informações sobre o mesmo com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138 - 146 - Botafogo
CEP: 22290-240
Rio de Janeiro - RJ

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

Esta publicação contém exemplos de dados e relatórios utilizados em operações diárias de negócios. Para que sejam ilustrados do modo mais completo possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa comercial real será mera coincidência.

Esta publicação pode conter informações sobre outros produtos de empresas, incluindo referências de tais sites da Internet de empresas. A IBM não se responsabiliza pela exatidão, totalidade ou uso de tais informações.

Este produto é baseado no protocolo SET.

Nota a Usuários do Governo dos Estados Unidos — Documentação relacionada a direitos restritos — Uso, duplicação ou divulgação sujeitos às restrições definidas no documento GSA ADP Schedule Contract com a IBM Corp.

Marcas registradas

Os termos a seguir são marcas ou marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países:

DB2 Extenders
IBM
SecureWay
400

Encina
iSeries
VisualAge

HotMedia
MQSeries
WebSphere

Blaze Advisor é uma marca registrada da HNC Software, Inc. nos Estados Unidos, em outros países ou em ambos.

Java e todas as marcas e logotipos baseados em Java são marcas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

Lotus e Domino são marcas da Lotus Development Corporation nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT, e o logotipo de Windows são marcas registradas de Microsoft Corporation nos Estados Unidos, em outros países ou em ambos.

Oracle é marca registrada da Oracle Corporation.

SET e o logotipo SET são marcas comerciais de propriedade da SET Secure Electronic Transaction LLC. Para obter informações adicionais, consulte <http://www.setco.org/aboutmark.html>.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Outras companhias, produtos ou nome de serviços podem ser marcas ou marcas de serviços de terceiros.

IBM