

IBM WebSphere Commerce



Tienda de ejemplo NewFashion: Archivos de ayuda en línea

Versión 54

IBM WebSphere Commerce



Tienda de ejemplo NewFashion: Archivos de ayuda en línea

Versión 54

Aviso

Antes de utilizar esta información y el producto al que da soporte, lea la información de carácter general de la sección "Avisos".

Contenido

Capítulo 1. Creación de una tienda

utilizando un ejemplo 1

Crear una tienda utilizando un ejemplo	1
Crear un archivador de tienda de ejemplo	1
Cambiar elementos de base de datos de tienda	3
Cambiar elementos Web	16
Publicar un archivador de tienda de ejemplo desde Servicios de tienda	16
Publicar un archivador de tienda utilizando la línea de mandatos	17
Configurar Payment Manager para la tienda	20
Crear trabajos planificados para una tienda de ejemplo.	23
Configurar notificaciones de correo electrónico para una tienda de ejemplo.	25

Capítulo 2. Elementos de base de datos de la tienda de ejemplo 27

Elementos de base de datos de tienda	27
Elementos de base de datos de Catálogo de NewFashion	28
Elementos de base de datos de contrato y de política de negocio de NewFashion	30
Elementos de base de datos de envío de NewFashion	31
Elementos de base de datos de control de acceso de las tiendas de ejemplo	33

Capítulo 3. Tienda de ejemplo NewFashion 35

Tienda de ejemplo NewFashion.	35
---------------------------------------	----

Capítulo 4. Páginas de la tienda NewFashion 37

Páginas de búsqueda de la tienda de ejemplo NewFashion	37
Páginas de listín de la tienda de ejemplo	39
Páginas de listín de la tienda de ejemplo NewFashion	40
Páginas de visualización de paquete compuesto de las tiendas de ejemplo WebFashion y NewFashion	43
Páginas de grupo de catálogo de la tienda de ejemplo.	44
Páginas de la tienda de ejemplo: técnicas de implementación comunes.	46
Página Contáctenos de la tienda de ejemplo	49
Pie de página de la tienda de ejemplo	49
Cabecera de la tienda de ejemplo	50
Página Ayuda de la tienda de ejemplo	52
Página de presentación de la tienda de ejemplo	52
Páginas de conexión de la tienda de ejemplo	53
Página de privacidad de la tienda de ejemplo	56
Páginas de productos de la tienda de ejemplo	56
Página de registro de la tienda de ejemplo	58

Página Novedades de la tienda de ejemplo	61
Página de confirmación del pedido de la tienda de ejemplo NewFashion	62
Página de notificación de correo electrónico de WebFashion y NewFashion	62
Página de resumen del pedido de la tienda de ejemplo.	63
Página de comprobación de disponibilidad de productos de la tienda de ejemplo NewFashion	64
Páginas de visualización de paquete de las tiendas de ejemplo WebFashion y NewFashion	65
Marco de navegación izquierdo de la tienda de ejemplo.	67
Marco de navegación izquierdo de la tienda de ejemplo NewFashion	68
Página de selección de dirección de facturación de la tienda de ejemplo	70
Página de selección de dirección de facturación de la tienda de ejemplo NewFashion	72
Carro de la compra de la tienda de ejemplo	73
Carro de la compra de la tienda de ejemplo NewFashion	75
Página de método de envío de la tienda de ejemplo	76
Página de selección de método de envío de la tienda de ejemplo NewFashion	77
Página de selección de dirección de envío de la tienda de ejemplo	79
Página de selección de dirección de envío de la tienda de ejemplo NewFashion	80
Página de resumen de caja rápida de la tienda de ejemplo.	81
Páginas para ver pedidos de las tiendas de ejemplo WebFashion y NewFashion	83
Páginas de lista de deseos de las tiendas de ejemplo WebFashion y NewFashion	85
Ejemplos de uso de la tienda de ejemplo	87

Capítulo 5. Ejemplos de uso de la tienda de ejemplo 89

Ejemplo de uso de Añadir nueva dirección (Business Edition)	89
Ejemplo de uso de Añadir artículo a la lista de deseos de WebFashion y NewFashion	90
Ejemplo de uso de Ver categoría de producto	90
Ejemplo de uso de Pasar por caja el carro de la compra.	91
Ejemplo de uso de Visualizar página de paquete compuesto de WebFashion y NewFashion	94
Ejemplo de uso de Visualizar página de paquete de WebFashion y NewFashion	95
Ejemplo de uso de Editar una dirección (Business Edition)	96
Ejemplo de uso de Página de presentación	97
Ejemplo de uso de Conexión	98
Ejemplo de uso de Gestionar cuenta personal	99

Ejemplo de uso de Ver pedidos de WebFashion y NewFashion	100
Ejemplo de uso de Visualizar página de producto	101
Ejemplo de uso de Modificar información personal	102
Ejemplo de uso de Registro.	103
Ejemplo de uso de Visualizar carro de la compra	104
Ejemplo de uso de Ver lista de deseos de WebFashion y NewFashion	105

Flujo de carro de la compra de la tienda de ejemplo NewFashion	109
Flujo de compra de la tienda de ejemplo NewFashion	110

Avisos 112

Capítulo 6. Flujo de compra de la tienda NewFashion 109

Capítulo 1. Creación de una tienda utilizando un ejemplo

Crear una tienda utilizando un ejemplo

Para crear un archivador de tienda utilizando una de las tiendas de ejemplo, haga lo siguiente:

1. (Opcional)  Cree una organización que actúe como vendedor. Para determinar si desea crear una nueva organización que actúe como vendedor, consulte Elementos de datos compartidos.
2. Cree usuarios para los roles siguientes:
 - Administrador de sitio (si no está utilizando el administrador de sitio por omisión)
 - Administrador de vendedores
 - Administrador de tienda
 - Desarrollador de tiendas

Importante: Para publicar un archivador de tienda debe tener acceso de Administrador de sitio, Administrador de tienda o Desarrollador de tiendas de todos los accesos de tienda. Si va a crear un usuario con autorización de Administrador de tienda, asegúrese de que el grupo de acceso se aplica a todas las tiendas.
3. Cree un archivador de tienda utilizando Servicios de tienda
4. Cambie elementos de base de datos de tienda.
5. Cambie los elementos Web.
6. Publique un archivador de tienda.
7. (Opcional) Configure tiendas.
8. Configure Payment Manager para la tienda.

Importante:

1. En algunos casos, es necesario realizar determinadas tareas de configuración en las tiendas de ejemplo para que todas las características funcionen correctamente. Si está basando su tienda en una tienda de ejemplo, es posible que necesite llevar a cabo la configuración. Para obtener más información, consulte Configurar una tienda de ejemplo.
2. Ciertos elementos de tienda como, por ejemplo, los catálogos y los centros de despacho de pedidos, los comparten varias tiendas. Por este motivo, si publica diversas tiendas en la misma tienda de ejemplo, el catálogo y el centro de despacho de pedidos serán los mismos para cada una de las tiendas. Si efectúa cambios en un catálogo, al publicar otra tienda en la misma tienda de ejemplo, los cambios se perderán. Para obtener más información e indicaciones para no perder estos cambios, consulte Elementos de datos compartidos.

Crear un archivador de tienda de ejemplo

Para crear un archivador de tienda que puede utilizarse como ejemplo con los Servicios de tienda, consulte la publicación *IBM WebSphere Commerce, Guía del desarrollador de tiendas*.

Cambiar elementos de base de datos de tienda

Si crea el archivador de tienda utilizando las herramientas de los Servicios de tienda, el nuevo archivador de tienda contendrá inicialmente los mismos elementos de base de datos de tienda que el archivador de tienda de ejemplo en el que está basado, por ejemplo `infashion.sar`. En un archivador de tienda, los elementos de base de datos de tienda adoptan el formato de archivos XML.

En la mayoría de casos, para cambiar los elementos de base de datos de la tienda, debe editar los archivos XML directamente. En algunos casos, puede utilizar las herramientas de los Servicios de tienda para editar los elementos de la tienda.

También tiene la opción de editar los elementos de base de datos directamente, es decir, cuando haya publicado el archivador de tienda en Commerce Server, podrá editar la base de datos utilizando WebSphere Commerce Accelerator, Loader package o directamente mediante inserciones SQL. Si decide editar la base de datos en lugar de los elementos del archivador de tienda, debe actualizar el archivador de tienda para que los cambios coincidan en la base de datos o dejar de utilizar el archivador de tienda.

Las opciones figuran listadas en la tabla siguiente.

Importante:

1. Las herramientas de los Servicios de tienda buscan los nombres de elementos que figuran listados en la tabla siguiente. Por ello, si desea editar el archivador de tienda utilizando las herramientas de Servicios de tienda, deberá utilizar los mismos nombres de elementos del archivador de tienda.
2. Cuando se publica un archivador de tienda en WebSphere Commerce Server, la información de base de datos se carga en el orden especificado en la columna Elementos siguiente. Como resultado, el orden de los elementos, tal y como se ha especificado en el archivo `sarinfo.xml`, deberá coincidir con el orden de los elementos especificado a continuación.
3. El orden de la información de base de datos de cada archivo XML no necesariamente tiene que coincidir con el orden especificado en la columna Tablas de base de datos siguiente. Sin embargo, la información de una tabla padre debe preceder a la información de la tabla hijo.
4. La información que está marcada como opcional no es necesaria para crear una tienda que funcione.
5. En la columna Opciones de edición de base de datos, observe que a menos que se especifique lo contrario todos los elementos de base de datos se pueden editar utilizando las inserciones SQL o Loader package. Como resultado, la columna refleja los elementos que WebSphere Commerce Accelerator puede editar.

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
fulfillment	FFMCENTER (0...1)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Páginas Nuevo centro de despacho de pedidos y Cambiar el centro de despacho de pedidos en WebSphere Commerce Accelerator. • Consulte Crear un centro de despacho de pedidos y Cambiar un centro de despacho de pedidos
store	STOREENT	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – IDENTIFIER – DIRECTORY – SETCURR 	
	STADDRESS	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. 	<ul style="list-style-type: none"> • Páginas Nuevo centro de despacho de pedidos y Cambiar el centro de despacho de pedidos en WebSphere Commerce Accelerator. • Consulte Crear un centro de despacho de pedidos y Cambiar un centro de despacho de pedidos
	STOREENTDS	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – DESCRIPTION – DISPLAYNAME 	
	STORELANG	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – LANGUAGE_ID 	
	STORELANGDS	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
store	STORE	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - PHONE1 - CITY - STORE_ID - COUNTRY - STATE - EMAIL1 - ADDRESS1 - ADDRESS2 - FAX - ZIPCODE 	
	DISPENTREL (plantilla CATENTRY por omisión, CATENTRY_ID = -1)	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita la columna de base de datos PAGENAME. 	
	DISPCGPREL (plantilla CATEGORY por omisión, CATGROUP_ID = -1)	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita la columna de base de datos PAGENAME. 	
	VENDOR	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
catalog	CATGROUP	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATGRPATTR	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	CATGRPDESC	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATALOG	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	CATALOGDSC	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	CATTOGRP	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	CATGRPREL	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATENTRY	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATENTDESC	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	ATTRIBUTE	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
catalog	ATTRVALUE	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATGPENREL	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATENTREL	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	BASEITEM	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	ITEMSPC	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	VERSIONSPC	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	DISTARRANG	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	RECEIPT	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
	RCPTAVAIL	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	
STOREITEM	<ul style="list-style-type: none"> • Cambiar información del catálogo • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
tax	JURSTGROUP (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - JURSTGROUP_ID - DESCRIPTION - SUBCLASS - STOREENT_ID - CODE 	
	JURST (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - JURST_ID - COUNTRY - STOREENT_ID - CODE - SUBCLASS - STATE 	
	JURSTGPREL (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - JURST_ID - JURSTGROUP_ID - SUBCLASS 	
	CALMETHOD (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	TXCDCLASS (opcional, solamente para categorías de CALCODE, utilizadas por WebSphere Commerce Accelerator)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
tax	TAXCGRY	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - TAXCGRY_ID - STOREENT_ID - NAME 	
	CALCODE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - CALCODE_ID - CODE - CALUSAGE_ID - STOREENT_ID - GROUPBY - CALMETHOD_id - CALMETHOD_id_app - CALMETHOD_id_qfy 	
	CALCODEDSC (común entre impuestos y envío, opcional)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CALRULE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - CALRULE_ID - CALCODE_ID - TAXCGRY_ID - CALMETHOD_ID 	
	CALSCALE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> - CALSCALE_ID - CODE - STOREENT_ID - CALUSAGE_ID - SETCURR - CALMETHOD_ID 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
tax	CALSCALEDS (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CALRANGE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> – CALRANGE_ID – CALSCALE_ID – CALMETHOD_ID – RANGESTART 	
	CALRLOOKUP (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> – CALSCALE_ID – CALRULE_ID 	
	CRULESCALE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> – CALSCALE_ID – CALRULE_ID 	
	CALRULEMGP (opcional)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CALCOTXEX (exención de impuestos, utilizado en descuentos)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	STENCALUSG (común, CALCODE por omisión para tienda, 2 entradas: 1 para impuestos sobre venta, 1 para impuestos sobre envío)	<ul style="list-style-type: none"> • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. 	
taxfulfillment	TAXJCRULE	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. • Cambiar valores de impuestos utilizando el cuaderno de Impuestos. El cuaderno de Impuestos edita las siguientes columnas de base de datos: <ul style="list-style-type: none"> – CALRULE_ID – FFMCENTER_ID – JURSTGROUP_ID 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
store-catalog-tax	CATENCALCD (opcional)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATGPCALCD (opcional)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
shipping	JURSTGROUP (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – JURSTGROUP_ID – DESCRIPTION – SUBCLASS – STOREENT_ID – CODE 	
	JURST (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – JURST_ID – COUNTRY – STOREENT_ID – CODE – SUBCLASS – STATE 	
	JURSTGPREL (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – JURST_ID – JURSTGROUP_ID – SUBCLASS 	
	SHIPMODE	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> – CODE – CARRIER – SHIPMODE_ID 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
shipping	SHPMODEDSC	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - SHIPMODE_ID - LANGUAGE_ID 	
	CALMETHOD (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CALCODE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALCODE_ID - CODE - CALUSAGE_ID - STOREENT_ID - GROUPBY - CALMETHOD_ID 	
	CALCODEDSC (opcional)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. 	
	CALRULE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALRULE_ID - CALCODE_ID - TAXCGRY_ID - CALMETHOD_ID 	
	CALSCALE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <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 archivos XML del archivador de tienda. 	
	CALSCALEDS (común entre impuestos y envío)	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
shipping	CALRANGE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALRANGE_ID - CALSCALE_ID - CALMETHOD_ID 	
	CALRLOOKUP (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALRLOOKUP_ID - SETCURR - CALRANGE_ID - VALUE 	
	CRULESCALE (común entre impuestos y envío)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALSCALE_ID - CALRULE_ID 	
	STENCALUSG (común, CALCODE por omisión para la tienda. 1 entrada para cálculos de envío, 1.)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. 	
ship fulfillment	SHPJCRULE (como mínimo una norma por omisión para la tienda)	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - CALRULE_ID - SHPARRANGE_ID - JURSTGROUP_ID 	
	SHPARRANGE	<ul style="list-style-type: none"> • Cambiar valores de envío utilizando el cuaderno Envío. El cuaderno Envío edita las columnas de base de datos siguientes: <ul style="list-style-type: none"> - SHARRAND_ID - STORE_ID - FFMCENTER_ID - SHIPMODE_ID 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
store-catalog	STORECAT	• Editar archivos XML del archivador de tienda.	
	STORECENT	• Editar archivos XML del archivador de tienda.	
	STORECGRP	• Editar archivos XML del archivador de tienda.	
	DISPENTREL	• Editar archivos XML del archivador de tienda.	
	DISPCGPREL	• Editar archivos XML del archivador de tienda.	
store fulfillment	INVENTORY	• Editar archivos XML del archivador de tienda.	
offering	TRADEPOSCN (1)	• Editar archivos XML del archivador de tienda.	
	MGTRDPSCN (opcional, para grupo de clientes)	• Editar archivos XML del archivador de tienda.	
	OFFER	• Editar archivos XML del archivador de tienda.	• Herramientas de gestión de productos en WebSphere Commerce Accelerator
	OFFERDESC	• Editar archivos XML del archivador de tienda.	
	OFFERPRICE	• Editar archivos XML del archivador de tienda.	• Herramientas de gestión de productos en WebSphere Commerce Accelerator
	LISTPRICE	• Editar archivos XML del archivador de tienda.	
command	URLREG	• Editar archivos XML del archivador de tienda.	
	CMDREG	• Editar archivos XML del archivador de tienda.	
	VIEWREG	• Editar archivos XML del archivador de tienda.	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
currency	CURCONVERT	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CURFORMAT	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CURFMTDESC	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CURCVLIST	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	CURLIST	<ul style="list-style-type: none"> • Cambiar información de la tienda utilizando el cuaderno Perfil de tienda. El cuaderno Perfil de tienda edita CURRSTR. 	
campaign	EMSPOT	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de zonas de e-Marketing en WebSphere Commerce Accelerator
	CAMPAIGN	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de campañas en WebSphere Commerce Accelerator
store-catalog-shipping	CATENCALCD	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	<ul style="list-style-type: none"> • Herramientas de gestión de productos en WebSphere Commerce Accelerator
	CATGPCALCD	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
store-defaults	STOREDEF	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
consistency check		<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
payment	CMDREG	<ul style="list-style-type: none"> • Cambiar los valores de pago. • Editar archivos XML del archivador de tienda. 	
	VIEWREG	<ul style="list-style-type: none"> • Cambiar los valores de pago. • Editar archivos XML del archivador de tienda. 	
policy	POLICY	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	POLICYCMD	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	

Elementos de base de datos de tienda	Tablas de base de datos	Opciones de edición de archivador de tienda	Opciones de edición de base de datos
 organization	ORGENTITY	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	MBRREL	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	ADDRBOOK	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
	ADDRESS	<ul style="list-style-type: none"> • Editar archivos XML del archivador de tienda. 	
 business account	TERMCOND	Los elementos de cuenta no se pueden actualizar ni volver a publicar mediante el archivador de tienda. Si necesita cambiar los elementos de cuenta deberá editar los elementos en la base de datos utilizando las herramientas de gestión de relaciones comerciales de WebSphere Commerce Accelerator.	Herramientas de gestión de relaciones comerciales en WebSphere Commerce Accelerator Importante: Las cuentas no se pueden cargar utilizando Loader package.
	ACCOUNT		
	TRADING		
	TCDESC		
	PATTRVALUE		
	CREDITLINE		
	TRDDESC		
	POLICYTC		
	BUYERPO		
	PARTICIPNT		
	ATTACHMENT		
TRDATTACH			
 contract	CONTRACT	<ul style="list-style-type: none"> • Editar archivos XML de contrato en el archivador de tienda 	Herramientas de gestión de relaciones comerciales en WebSphere Commerce Accelerator Importante: Los contratos no se pueden cargar utilizando Loader package.
	TERMCOND		
	PRODUCTSET		
	TRADING		
	TCDESC		
	PATTRVALUE		
	TRDDESC		
	POLICYTC		
	PARTICIPNT		
	TRADEPOSCN		
	ATTACHMENT		
	OFFER		
	TRDATTACH		
	OFFERPRICE		
	STORECNTR		
	PURCHASELT		
	STOREDEF (utilizado por lo servicios de tienda)		
APRVSTATUS			
FLINSTANCE			

Cambiar elementos Web

Para cambiar los elementos Web del archivador de tienda, tiene las opciones siguientes:

- Importe el archivador de tienda a WebSphere Studio y realice los cambios necesarios en los elementos Web utilizando Page Designer o una herramienta de su elección. Si es necesario, cree páginas de tienda nuevas. A continuación, vuelva a exportar los elementos Web al archivador de tienda o a la tienda en funcionamiento.
- Baje los elementos Web del archivador de tienda a una ubicación de su elección utilizando el diálogo Elementos Web de Servicios de tienda y, a continuación, cámbielos utilizando una herramienta de su elección. O bien sustituya los elementos Web del archivador de tienda por los elementos Web existentes utilizando el diálogo Elementos Web de Servicios de tienda. Si es necesario, cree páginas de tienda nuevas.
- Abra manualmente el archivo archivado comprimido de elementos Web del archivador de tienda y realice cambios en los archivos existentes o añada archivos nuevos.

Publicar un archivador de tienda de ejemplo desde Servicios de tienda

Publicar un archivador de tienda de ejemplo en WebSphere Commerce Server le permite crear una tienda funcional. Para publicar un archivador de tienda, complete los pasos siguientes:

1. Asegúrese de que tiene acceso de Administrador de sitio o de Administrador de tienda. Si tiene acceso de Administrador de tienda, asegúrese de que el acceso es para todas las tiendas.
2. Asegúrese de haber completado las tareas de Comprobar que los servicios y servidores adecuados están en ejecución.
3. Abra Servicios de tienda.
4. En la lista **Archivador de tienda**, seleccione el archivador de tienda que desea publicar.
5. Pulse **Publicar**.
Se visualizará la página Publicar archivador de tienda.
6. Seleccione las opciones de publicación que desee. Para obtener más información sobre las opciones de publicación, pulse **Ayuda**.
Sugerencia: Para crear una tienda totalmente funcional, la primera vez que publique un archivador de tienda seleccione todas las opciones de publicación, incluida la opción de datos de productos.
7. Seleccione **Aceptar**.
Mientras la tienda se publica, volverá a la lista Archivador de tienda. La columna Estado de la publicación mostrará el estado de la publicación. Pulse **Renovar** para actualizar el estado.
8. Seleccione el archivador de tienda de la lista y pulse **Resumen de publicación** para ver el resultado de la publicación.
9. Una vez finalizada la publicación, pulse **Iniciar tienda** para ver y comprobar la tienda. Cuando haya finalizado, cree un marcador del sitio y cierre el navegador.

Importante:

1. Si modifica la vía de acceso Web o el directorio raíz de documentos de la aplicación Web, debe asegurarse de que coincidan con la vías de acceso definidas en WebSphere Commerce Server.

2. Sólo puede publicarse un archivador de tienda a la vez. La publicación simultánea de tiendas no está soportada, por lo que si intenta publicar dos tiendas no obtendrá resultados satisfactorios.
3. Durante el proceso de publicación, la comprobación de coherencia confirma que existan los archivos a los que hace referencia el archivador de tienda. Si la comprobación de coherencia encuentra un error, éste se grabará en un archivo de anotaciones cronológicas. El proceso de publicación seguirá normalmente.
4. Antes de volver a publicar una tienda, suprima los archivos del directorio siguiente:

```

▶ NT      unidad:\WebSphere\CommerceServer\instances\nombreinstancia\cache
▶ 2000    unidad:\Archivos de
programa\WebSphere\CommerceServer\nombreinstancia\cache
▶ AIX     /usr/WebSphere/CommerceServer/instances/nombreinstancia/cache
▶ Solaris /opt/WebSphere/CommerceServer/instances/nombreinstancia/cache
▶ 400     /QIBM/UserData/WebCommerce/instances/nombreinstancia/cache

```

Mientras esté en la fase de desarrollo de la tienda, tal vez desee inhabilitar el almacenamiento en antememoria. Para obtener más información, consulte Configurar almacenamiento en antememoria.

5. Cuando inicia la tienda desde los Servicios de tienda, se conecta a la tienda con el mismo nombre de usuario y contraseña que ha utilizado para conectarse a los Servicios de tienda. Si cambia la contraseña en la tienda, también la cambia para los Servicios de tienda. En lugar de ello, para probar las características en la tienda, incluido el cambio de contraseña, guarde la dirección de sitio, cierre todas las ventanas de navegador y, a continuación, conéctese a la tienda otra vez.
6. Si se ha conectado como el administrador por omisión, no puede navegar en una tienda basada en la tienda de ejemplo de empresa a empresa (business to business), sino que debe crear un usuario nuevo que pertenezca a la organización por omisión y luego navegar por la tienda.
7. Ciertos elementos de tienda como, por ejemplo, los catálogos y los centros de despacho de pedidos, los comparten varias tiendas. Por este motivo, si publica diversas tiendas en la misma tienda de ejemplo, el catálogo y el centro de despacho de pedidos serán los mismos para cada una de las tiendas. Si efectúa cambios en un catálogo, al publicar otra tienda en la misma tienda de ejemplo, los cambios se perderán. Para obtener más información e indicaciones para no perder estos cambios, consulte Elementos de datos compartidos.

Publicar un archivador de tienda utilizando la línea de mandatos

Aunque el método principal para publicar un archivador de tienda es mediante los Servicios de archivador de tienda, también puede publicarlo desde la línea de mandatos. Para ello, haga lo siguiente:

1. Asegúrese de que tiene acceso de Administrador de sitio o de Administrador de la tienda. Si tiene acceso de Administrador de tienda, asegúrese de que el acceso es para todas las tiendas.
2. Utilizando los parámetros válidos para el archivador de tienda, escriba el mandato siguiente:

```
publishstore sarName hostname logonId logonPwd {insert|update}
  destination1=webapp.zip,destination2=properties.zip
```

 donde:

- `sarName` es el nombre del archivador de tienda. `sarName` es sensible a las mayúsculas y minúsculas. Compruebe que utiliza las mayúsculas/minúsculas correctas.
- `hostname`, donde *hostname* es el nombre TCP/IP totalmente calificado de WebSphere Commerce Server y el número de puerto de las herramientas de su instancia. Puede encontrar el número de puerto de las herramientas en el Gestor de configuración, bajo **Propiedades de instancia** > **WebSphere**. Por omisión es `nombresistpral:8000`.
- `logonId` es el ID de usuario de WebSphere Commerce.
- `logonPwd` es la contraseña de conexión del usuario para WebSphere Commerce
- `insert|update` determina si la tienda se está creando (`insert`) o actualizando (`update`).
- `{ALL|NOCATLG}` determina que los archivos XML de SAR se deben publicar. Para publicar todos los archivos, especifique `ALL`. Para publicar todo excepto el catálogo, especifique `NOCATLG`.
- `destination1=webapp.zip,destination2=properties.zip`, es la lista de los archivos de elementos de archivo del SAR, por ejemplo `webapp.zip`, y las vías de acceso en las que se publicarán, por ejemplo, `destination1` es


```
> NT unidad:\Websphere\AppServer\installedApps\
WC_Enterprise_App_nombreinstancia.ear\wcstores.war
> 2000 unidad:\Archivos de
programa\Websphere\AppServer\installedApps\
WC_Enterprise_App_nombreinstancia.ear\wcstores.war
```

`destination2` es

```
> NT
unidad:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
wcstores.war\WEB-INF\classes
> 2000 unidad:\Archivos de
programa\WebSphere\AppServer\installedApps\
WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes
```

A continuación, se muestra un ejemplo del mandato:

```
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 Internet Explorer. Vaya a la dirección Web siguiente:
`http://nombresistpral/webapp/wcs/stores/directorio tienda/index.jsp`, donde *directorio tienda* es el directorio de la tienda que acaba de publicar.
Se visualizará la tienda.

Nota: Si está publicando un archivador de tienda creado con WebSphere Commerce Suite versión 5.1, necesitará realizar los pasos siguientes antes de iniciar la tienda a través del URL:

1. Si utiliza DB2, haga lo siguiente. Si utiliza Oracle, vaya al paso 2.
 - a. Cuando el mandato se haya ejecutado, en el menú **Inicio**, seleccione **Programas, DB2 para Windows NT y Ventana de mandatos**.
 - b. En la ventana **DB2 CLP**, escriba `db2 connect to nombrebd`, donde `nombrebd` es la base de datos en la que publica la tienda. Pulse **Intro**.

- c. En la línea de mandatos, escriba `db2 select * from store`. Pulse **Intro**. Se visualizará una lista de tiendas. Apunte el número de la tienda que ha creado.
 - d. En la línea de mandatos, escriba `db2 select * from catalog`. Pulse **Intro**. Se visualizará una lista de catálogos. Apunte el número del catálogo de tienda de ejemplo.
 - e. Vaya al paso 3.
2. Si utiliza Oracle, haga lo siguiente:
 - a. Cuando el mandato se haya ejecutado, en el menú **Inicio**, seleccione **Programas, Oracle - HomeOra81, Application Development** y, finalmente, **SQL Plus**.
 - b. En la ventana, escriba su nombre de usuario, contraseña y serie de sistema principal.
 - c. En la ventana de SQL Plus, escriba `select * from store;`. Pulse **Intro**. Se visualizará una lista de tiendas. Apunte el número de la tienda que ha creado.
 - d. En la ventana de SQL Plus, escriba `select * from catalog;`. Pulse **Intro**. Se visualizará una lista de catálogos. Apunte el número del catálogo de tienda de ejemplo.
 - e. Vaya al paso 3.
 3. Abra Internet Explorer. Vaya al siguiente URL:
<http://nombresistpral/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=ID tienda del paso 1c o 2c&langId=-1&catalogId=ID catálogo del paso 1d o 2d>
 Se visualizará la tienda.
 Si tiene problemas a la hora de visualizar la tienda, consulte Resolución de problemas de publicación.

Importante:

1. Durante el proceso de publicación, la comprobación de coherencia confirma que existan los archivos a los que hace referencia el archivador de tienda. Si la comprobación de coherencia encuentra un error, éste se grabará en un archivo de anotaciones cronológicas. El proceso de publicación seguirá normalmente.
2. Antes de volver a publicar una tienda, suprima los archivos del directorio siguiente:
 - ▶ NT
 - unidad:\WebSphere\CommerceServer\instances*nombre instancia*\cache
 - ▶ 2000 *unidad*:\Archivos de programa\WebSphere\CommerceServer\instances*nombre instancia*\cache
3. Mientras esté en la fase de desarrollo de la tienda, inhabilite los desencadenantes de almacenamiento en antememoria y la antememoria. Si deja activada la antememoria puede ocurrir lo siguiente:
 - Los cambios realizados en los archivos JSP pueden no visualizarse en el navegador.
 - Cuando la base de datos se actualice, se invocarán los desencadenantes de almacenamiento en antememoria durante la publicación. Los desencadenantes de almacenamiento en antememoria pueden generar una actividad innecesaria en la base de datos que puede provocar un desbordamiento de las anotaciones cronológicas de transacciones de base de datos. Para obtener más información, consulte Configurar almacenamiento en antememoria.

4. Si se ha conectado como el administrador por omisión, no puede navegar en una tienda basada en la tienda de ejemplo de empresa a empresa (business to business), sino que debe crear un usuario nuevo que pertenezca a la organización por omisión y luego navegar por la tienda.

Configurar Payment Manager para la tienda

Puede completar la configuración de Payment Manager para su tienda utilizando la Consola de administración o la interfaz de usuario de Payment Manager. Si utiliza la Consola de administración, los elementos de menú aparecerán en el menú **Payment Manager**. Si utiliza la interfaz de usuario de Payment Manager, los elementos de menú aparecerán en el marco de navegación **Administración**.

Si crea la tienda mediante el archivador de tienda de ejemplo (que es lo que se recomienda), Payment Manager se configurará parcialmente.

Para completar la configuración de Payment Manager para su tienda, haga lo siguiente:

1. Abra la Consola de administración o la interfaz de usuario de Payment Manager.
2. Asigne roles de usuario de Payment Manager a los usuarios de WebSphere Commerce, según sea necesario. Para asignar roles de usuario de Payment Manager, seleccione **Usuarios**.

Al Administrador de sitio de WebSphere Commerce por omisión, wcsadmin, se le asigna por omisión el rol de administrador de Payment Manager. Es posible que desee asignar varios roles de Payment Manager a otros usuarios de WebSphere Commerce.

3. Autorice casetes para la tienda; para ello:
 - a. Seleccione **Valores de comerciante**.
 - b. Pulse el nombre de la tienda en la columna **Nombre de comerciante**.
 - c. Seleccione los casetes que desea autorizar a la tienda.
 - d. Pulse **Actualizar**.

Importante: Si creó la tienda manualmente, debe añadir un nuevo comerciante (la tienda) para autorizar casetes a la tienda. Al crear un nuevo comerciante, el número de comerciante especificado debe coincidir con el ID de tienda de WebSphere Commerce. Puede crear un nuevo comerciante seleccionando **Valores de comerciante** y pulsando **Añadir un comerciante**.

4. Configure casetes para la tienda; para ello:
 - a. Seleccione **Valores de comerciante**.
 - b. Seleccione el casete que desea configurar pulsando el icono que aparece en la fila de la tienda y la columna del casete que quiere configurar.
 - c. Pulse **Cuentas** en la página del casete para la tienda y lleve a cabo una de las acciones siguientes:
 - Para cambiar cuentas existentes, pulse el nombre de la cuenta.
 - Para crear una nueva cuenta, pulse **Añadir una cuenta**.

Para obtener más información sobre la configuración del Casete para BankServACH, consulte la publicación *IBM WebSphere Payment Manager for Multiplatforms, Cassette for BankServACH Supplement, Version 3.1*.

Para obtener información sobre cómo configurar los casetes OfflineCard y CustomOffline, consulte la publicación *IBM WebSphere Payment Manager para varias plataformas, Guía del administrador, Versión 3.1*.

Para obtener información sobre cómo configurar el casete para SET, consulte la publicación *IBM WebSphere Payment Manager for Multiplatforms, Cassette for SET Supplement, Version 3.1*.

Para obtener información sobre cómo configurar el casete para CyberCash, consulte la publicación *IBM WebSphere Payment Manager for Multiplatforms, Cassette for CyberCash Supplement, Version 3.1*.

Para obtener información sobre cómo configurar el casete para VisaNet, consulte la publicación *IBM WebSphere Payment Manager for Multiplatforms, Cassette for VisaNet Supplement, Version 3.1*.

Para obtener ayuda cuando utilice Payment Manager desde la Consola de administración o la interfaz de usuario de Payment Manager, pulse  en la esquina superior derecha de la página de Payment Manager en la que esté trabajando.

Para obtener más información sobre las tareas anteriores u otras tareas de administración de Payment Manager, consulte la publicación *IBM WebSphere Payment Manager para varias plataformas, Guía del administrador, Versión 3.1*.

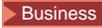
Para obtener información detallada sobre cómo instalar WebSphere Payment Manager, consulte las siguientes publicaciones:

- *IBM WebSphere Commerce Business Edition, Guía de instalación, Versión 5.4*
- *IBM WebSphere Payment Manager para varias plataformas, Guía de instalación, Versión 3.1*.

Crear trabajos planificados para una tienda de ejemplo

Después de publicar una tienda de ejemplo, es necesario crear trabajos planificados para la tienda. La tabla siguiente muestra los trabajos planificados que deberá crear para cada tienda de ejemplo:

Tienda de ejemplo	Trabajos planificados necesarios
InFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ReturnCreditAndCloseScan
NewFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ProcessBackorders• RAReallocate• ReleaseExpiredAllocations• ReleaseToFulfillment• ReturnCreditAndCloseScan
WebFashion	<ul style="list-style-type: none">• BalancePayment• PayCleanup• ReturnCreditAndCloseScan

Tienda de ejemplo	Trabajos planificados necesarios
 ToolTech	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ProcessBackorders • RAreallocate • ReleaseExpiredAllocations • ReleaseToFulfillment • ReturnCreditAndCloseScan
WebAuction	<ul style="list-style-type: none"> • BalancePayment • PayCleanup • ReturnCreditAndCloseScan

A continuación se proporciona una breve descripción de los trabajos:

BalancePayment

Este trabajo llama al mandato DoDepositCmd para que capture el pago una vez se ha enviado el pedido. Este mandato implementa la función de captura de pago automática para WebSphere Commerce.

PayCleanup

Este trabajo cancela los pedidos WebSphere Commerce cuyas peticiones de autorización de pago han sido rechazadas por las entidades financieras respectivas durante un período de tiempo superior al período de tiempo que hay configurado para una tienda.

ProcessBackorders

Este trabajo asigna el inventario a pedidos en espera que se han creado cuando el inventario no estaba disponible.

RAreallocate

(Redistribuir asignaciones en inventario esperado) Este trabajo redistribuye los EIR (Expected Inventory Records) entre los pedidos en espera existentes. Es necesario para pronosticar con exactitud cuándo estarán disponibles los artículos de pedidos en estado de espera a medida que se añade o modifica la información de los EIR y a medida que los artículos de pedidos en estado de espera se van suprimiendo o asignando.

ReleaseExpiredAllocations

Este trabajo devuelve el inventario asignado a la tabla de recepciones de un artículo de línea de pedidos que se había asignado anteriormente y que ya ha sobrepasado el límite de su tiempo de caducidad.

ReleaseToFulfillment

Este trabajo envía los artículos asignados a un pedido para su despacho.

ReturnCreditAndCloseScan

Este trabajo escanea las autorizaciones de artículos devueltos que pueden estar acreditados y marcadas como cerrados.

Para crear un trabajo planificado, siga las instrucciones que aparecen en el apartado Planificar un trabajo de nivel de tienda para cada trabajo. La tabla siguiente lista los parámetros recomendados para cada trabajo:

Nombre del trabajo planificado	Hora de inicio recomendada	Intervalo recomendado (segundos)	Prioridad recomendada
BalancePayment	00:00	86400	1
PayCleanup	00:00	86400	1
ProcessBackorders	00:00	43200	8
RReallocate	00:00	86400	1
ReleaseExpiredAllocations	00:00	3600	8
ReleaseToFulfillment	00:00	600	10
ReturnCreditAndCloseScan	00:00	86400	1

Si se establece la hora de inicio de un trabajo planificado en 00:00, éste se inicia inmediatamente.

Nota: No es necesario cumplimentar el campo **Parámetros de trabajo** de la ventana Planificar trabajo para estos trabajos.

Configurar notificaciones de correo electrónico para una tienda de ejemplo

El procedimiento siguiente permite notificar por correo electrónico al cliente cuando se autoriza un pago, cuando autoriza un pedido o cuando se cancela un pedido. Tenga en cuenta que debe tener establecido el servidor de correo de modo que pueda enviar correo electrónico a los clientes.

Nota: Si no tiene establecido un servidor de correo electrónico, no podrá enviar notificaciones de correo electrónico desde la tienda, pero el resto de las características de la tienda de ejemplo continuarán funcionando.

Las diferentes tiendas de ejemplo dan soporte a diferentes notificaciones de correo electrónico. La tabla siguiente muestra las notificaciones de correo electrónico a las que se da soporte en cada tienda:

Tienda de ejemplo	Notificaciones de correo electrónico soportadas	Tipo de mensaje
InFashion	Restablecimiento de contraseña	Mensaje de notificación para el restablecimiento de contraseña
NewFashion	Pedido autorizado	Mensaje de notificación de un pedido autorizado
	Restablecimiento de contraseña	Mensaje de notificación para el restablecimiento de contraseña
	Pedido sometido	Mensaje de notificación para un pedido recibido
	Pedido cancelado	Mensaje de notificación para un pedido cancelado
	Notificación de envío	Mensaje para notificar al cliente un manifiesto de envío de pedido.
	Lista de deseos	Mensaje de difusión

Tienda de ejemplo	Notificaciones de correo electrónico soportadas	Tipo de mensaje
WebFashion	Pedido autorizado	Mensaje de notificación de un pedido autorizado
	Restablecimiento de contraseña	Mensaje de notificación para el restablecimiento de contraseña
	Lista de deseos	Mensaje de difusión

Nota: La tienda de ejemplo WebAuction se basa en WebFashion. Para establecer la notificación por correo electrónico debe realizar todos los pasos para WebFashion y los pasos adicionales relacionados con la subasta. Para obtener más información sobre los pasos para la notificación por correo electrónico para WebAuction, consulte las Tareas relacionadas que se indican más abajo.

Para habilitar las notificaciones de correo electrónico, haga lo siguiente:

1. Compruebe que el servidor de administración de IBM WebSphere esté iniciado.
2. Abra la Consola de administración utilizando un ID de Administrador de sitio.
3. En la página Selección de sitio/tienda para Consola de administración, seleccione **Tienda**. Se muestra la sección Seleccionar tienda e idioma.
4. En el menú desplegable **Nombre**, seleccione la tienda.
5. En la lista desplegable **Idioma**, seleccione el idioma. Pulse **Aceptar**. Se visualizará la página de presentación de la Consola de administración de la tienda.
6. En el menú **Configuración**, pulse **Transportes**. Aparecerá la página Configuración del transporte.
 - a. Compruebe que el transporte **Correo electrónico** tiene el estado Activo. Si el correo electrónico está inactivo, selecciónelo, pulsando **Cambiar estado**.
 - b. Seleccione **Correo electrónico** y pulse **Configurar**. Aparecerá la página Parámetros de configuración del transporte.
 - c. En el campo **Sistema principal**, escriba el nombre totalmente calificado del servidor de correo, por ejemplo, `miservidor.ibm.com`.
 - d. En el campo **Protocolo**, escriba `smtp` en minúsculas o el protocolo que elija. Pulse **Aceptar**.
7. En el menú **Configuración**, pulse **Tipos de mensajes**. Se visualiza la página Configuración del tipo de mensaje.
8. Cree la notificación que se ha de enviar cuando se autorice el pago, como se indica a continuación:
 - a. Pulse **Nuevo**. Se visualizará la página Asignación de transporte de mensajes.
 - b. Seleccione el tipo de mensaje en la lista desplegable **Tipo de mensaje**. Consulte la tabla anterior para ver los tipos de mensajes que puede utilizar en la tienda.
 - c. En el campo **Gravedad del mensaje**, escriba 0 a 0
 - d. En la lista desplegable **Transporte**, seleccione **Correo electrónico**.
 - e. En la lista desplegable **Formato de dispositivo**, seleccione **Formato de dispositivo estándar**.
 - f. Pulse **Siguiente**. Se muestra la página Parámetros de asignación de transporte de mensajes.

g. Rellene los campos como se indica a continuación:

Sistema principal	El nombre totalmente calificado del servidor de correo, por ejemplo, <code>example.ibm.com</code>
Protocolo	Escriba <code>smtp</code> (debe escribirlo en minúsculas) o el protocolo que esté utilizando.
Destinatario	Escriba una dirección de correo electrónico válida. Esta dirección se sustituirá por la dirección de correo electrónico del cliente durante la ejecución.
Remitente	Escriba la dirección de correo electrónico que desea utilizar como remitente del mensaje, por ejemplo, <code>orders@example.ibm.com</code> . La dirección debe ser una dirección de correo electrónico para un usuario válido del servidor de correo.
Asunto	Escriba el texto que desea visualizar como la línea de asunto del mensaje, por ejemplo, Su pedido ha sido aceptado.

h. Pulse **Finalizar**. Se visualiza la página Configuración del tipo de mensaje.

9. Para cada tipo de mensaje de su tienda de ejemplo, repita el paso 8.

Notas:

- Generalmente, se tarda mucho tiempo en enviar una notificación de correo electrónico por un pedido cancelado. El tiempo puede reducirse si:
 1. Establece en un valor más pequeño el valor de la columna `REJECTEDORDEREXPIRY` en la tabla de base de datos `STORE`.
 2. Modifica el trabajo planificado `PayCleanup` para que el intervalo de planificación sea más pequeño.
- Para obtener más información sobre las notificaciones de envío, consulte el mensaje `ReleaseShipNotify`

Capítulo 2. Elementos de base de datos de la tienda de ejemplo

Elementos de base de datos de tienda

Los datos de tienda es la información que se carga en la base de datos del servidor de WebSphere Commerce, que permite que la tienda funcione. Para funcionar correctamente, un tienda debe tener los datos necesarios para dar soporte a todas las actividades de los clientes. Por ejemplo, para que un cliente haga una compra, la tienda debe contener un catálogo de la mercancía en venta (datos de catálogo), los datos asociados con el proceso de pedidos (datos de impuestos y de envío), y el inventario para despachar la petición (datos de inventario y de despacho de pedidos).

Los datos pueden ser exclusivos de una tienda o los pueden compartir varias tiendas. Para obtener más información, consulte Elementos de datos compartidos.

Los elementos de base de datos de tienda que se encuentran en los archivadores de las tiendas de ejemplo que se proporcionan con WebSphere Commerce, son archivos XML bien contruidos y válidos para Loader package, con las siguientes excepciones: los archivos XML de archivador de tienda están pensados para ser portables y no deben contener claves primarias generadas que sean específicas de una instancia determinada de la base de datos. En su lugar, utilizan alias internos (que se describen en Convenios de Loader de archivador de tienda) que IDResolver resuelve en el momento de la publicación. Los archivadores de tienda también utilizan un conjunto de macros DTD (que en XML se denominan entidades). Las macros actúan como contenedores de los valores que selecciona en los Servicios de tienda durante la creación de la tienda. La utilización de estos dos convenios permite copiar y publicar los archivadores de tienda de ejemplo varias veces.

El archivador de tienda de ejemplo incluye todos los elementos de base de datos necesarios para crear una tienda funcional. Puede modificar estos archivos para utilizarlos en su propio archivador de tienda o utilizarlos como guía para crear sus propios archivos XML. WebSphere Commerce necesita que se carguen determinados datos en la base de datos de WebSphere Commerce para crear una tienda funcional y que estos datos se carguen en el orden determinado por el esquema. Por ejemplo, la tabla FFMCENTER debe llenarse de datos antes que la tabla STOREENT. Dado que las tiendas de ejemplo incluyen todos los datos obligatorios en el orden y la estructura que WebSphere Commerce necesita, si utiliza como base o guía para su propia tienda los elementos de base de datos, se ahorrará una cantidad de tiempo importante durante el periodo de creación inicial.

Para obtener una lista de los archivos de elementos de base de datos utilizados en el archivador de tienda de ejemplo, consulte Elementos de base de datos de archivador de tienda de ejemplo. Para obtener información más detallada sobre los datos de tienda, consulte la publicación *IBM WebSphere Commerce, Guía del desarrollador de tiendas*.

Nota: Las DTD para los archivos XML de elementos de base de datos de tienda de ejemplo no están en los archivos de archivador de tienda. Están ubicados en el directorio siguiente:

▶ NT	<code>unidad:\WebSphere\CommerceServer\xml\sar</code>
▶ 2000	<code>unidad:\Archivos de programa\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>

Elementos de base de datos de Catálogo de NewFashion

El archivo `catalog.xml` almacena la información de catálogo para las tiendas de ejemplo de WebSphere Commerce. Cada tienda de ejemplo tiene su propio archivo `catalog.xml`. Para más información, vea el archivo `catalog.xml`.

Los elementos de base de datos de catálogo de NewFashion se dividen en las secciones siguientes:

- Grupos de catálogo
- Entidades de catálogo
- Característica ATP

Grupos de catálogo

Los grupos de catálogo son grupos de categorías y productos. Una categoría es de por sí un grupo de catálogo. Por ejemplo, Moda para caballero es una agrupación de las categorías que componen Moda para caballero, por ejemplo Pantalones y Camisas, mientras que la categoría Pantalones es una agrupación de productos.

En la tienda de ejemplo NewFashion hay varios grupos de catálogo:

- Moda para caballero
- Moda para señora
- Novedades
- Promoción de la página de presentación

Los grupos de catálogo se crean en la tabla `CATGROUP` y se asocian con un catálogo específico en la tabla `CATTOGRP`. Los grupos de catálogo pueden tener categorías superiores y subcategorías. Por ejemplo, Moda para caballero sería una categoría superior, y Pantalones una subcategoría. Si tiene una subcategoría, debe ponerla bajo la categoría superior en la tabla `CATGRPREL`.

Nota: Los productos de la categoría Promoción de la página de presentación o la categoría Novedades aparecen sólo en las categorías especiales. Cuando las promociones terminen, estos productos deben trasladarse a su categoría habitual.

Entidades de catálogo

El catálogo de la tienda de ejemplo NewFashion está formado por entidades de catálogo. Estas entidades se componen de lo siguiente:

- productos
- artículos
- paquetes
- paquetes compuestos

La información sobre las entidades de catálogo se almacena en la tabla CATENTRY. La tabla CATENTREL almacena las relaciones entre entidades de catálogo, como relaciones entre productos-artículos, paquetes y paquetes compuestos. Para obtener más información sobre entidades, consulte Paquetes y paquetes compuestos.

Productos

Para cada producto de la tienda NewFashion hay una entrada en la tabla BASEITEM. Un artículo base es un producto. Cada producto sólo puede tener una versión, que está determinada por una fecha de caducidad. Por consiguiente, puede tener solamente una camisa roja de señora, talla 42, que caduque el 1 de enero de 2010. La información de versión por producto se almacena en la tabla ITEMVERSN.

En la tienda NewFashion, el inventario no se comparte con otros comerciantes. Como resultado, cada producto del catálogo tiene una sola entrada en la tabla DISTARRANG. Cada fila de la tabla DISTARRANG representa un plan de distribución, lo que permite a una tienda vender su propio inventario.

Artículos

Un artículo es una instancia específica de un producto, definida por atributos. La información sobre un artículo específico se almacena en la tabla ITEMSPC. La información de inventario de cada artículo se guarda en la tabla RECEIPT. La fecha en que está previsto recibir un artículo en espera se encuentra en la tabla RADETAIL.

Nota: En la tienda NewFashion, el catálogo utiliza los precios de artículos de la tabla OFFERPRICE. Para más información, vea el archivo `offering.xml`.

Paquetes

Los paquetes son productos que se venden juntos, como un paquete de golf. Un paquete aparece como una línea en un carro de la compra. Los productos de un paquete no se pueden comprar por separado. Los productos se definen en el archivo `product.xml`, y se agrupan en paquetes en la tabla CATENTRY. Cada producto del paquete tiene una entrada separada.

Paquetes compuestos

Los paquetes compuestos son productos que están agrupados juntos pero que se pueden vender por separado. Cada producto de un paquete compuesto aparece como una línea en el carro de la compra. Los paquetes compuestos se almacenan en la tabla CATENTRY, y los productos asociados a un paquete compuesto se almacenan en la tabla CATGPENREL. Las relaciones entre productos y paquetes compuestos se definen en la tabla CATENTREL.

Los atributos para las entidades de catálogo, paquetes y paquetes compuestos se definen en las tablas PKGATTR y PKGATTRVAL. La tabla PKGATTR hace referencia a una tabla de atributos existente que pertenece al producto del paquete. La tabla PKGATTRVAL hace referencia al valor de atributo del producto en el paquete.

Característica ATP

La tienda NewFashion tiene algunos artículos especiales que presentan la característica ATP, como los siguientes:

- Caballeros, Ropa deportiva, Camiseta de deporte tiene un inventario disponible de cero.
- Señoras, Complementos, Bolso de noche está en pedido en espera forzado.
- Caballeros, Ropa deportiva, Camiseta de deporte tiene RADetails.

Si un artículo tiene cero existencias disponibles o está en pedido en espera forzado, no importa cuántos de estos artículos pida el cliente, éstos se pondrán automáticamente en pedido en espera. Si no hay ningún RADetail significa que la fecha prevista para el pedido en espera será la fecha actual más el margen de tiempo por omisión de la tienda para los pedidos en espera, (columna DEFAULTBOOFFSET en la tabla de la tienda), que está definido en 30 días.

Elementos de base de datos de contrato y de política de negocio de NewFashion

En WebSphere Commerce, todas las tiendas deben tener un contrato por omisión. El contrato por omisión está compuesto de términos y condiciones que se almacenan en el archivo `contract.xml`. Si los términos y condiciones hacen referencia a una política de negocio, la información de la política de negocio se almacena en el archivo `businesspolicy.xml`.

La información de elementos de base de datos de contrato y de política de negocio para NewFashion está dividida en las secciones siguientes:

- Contrato
- Política de negocio

Contrato

Términos y condiciones

Todos los contratos de las tiendas de ejemplo deben tener términos y condiciones. Los términos y condiciones son el conjunto de normas que se han acordado en un contrato y controlan el proceso de compra entre un comprador y un vendedor. Para cada contrato debe tener un conjunto de términos y condiciones para el precio.

También puede tener términos y condiciones para otros tipos de cargos, por ejemplo costes de envío. Cada contrato sólo debe tener un término y condición de costes de envío.

En el archivo `contract.xml`, se debe hacer referencia al comprador y al vendedor.

Política de negocio

Términos y condiciones

Una política describe las normas seguidas por una empresa para un proceso de negocio determinado. Si los términos y condiciones del contrato hacen referencia a una política de negocio, deberá llenar con datos `businesspolicy.xml` antes de importar el archivo `contract.xml`.

Nota: Algunas políticas a las que se hace referencia en `contract.xml` no están contenidas en `businesspolicy.xml`. Estas políticas forman parte de los datos de rutina de carga. Para obtener más información sobre los datos de rutina de carga, consulte Archivos de rutina de carga.

Elementos de base de datos de envío de NewFashion

Los elementos de base de datos de envío de NewFashion se almacenan en los archivos XML siguientes:

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

Los elementos de base de datos de envío de NewFashion se dividen en las secciones siguientes:

- Jurisdicciones
- Modalidades de envío
- Códigos de cálculo
- Normas de cálculo
- Escala de cálculo
- Rango de cálculo
- Búsqueda de cálculo
- Combinaciones de cálculo
- Envío-despacho de pedidos

Jurisdicciones

El archivo `shipping.xml` identifica las jurisdicciones para el envío. Las jurisdicciones se definen en la tabla `JURST`, `JURSTGROUP` asigna la jurisdicción a un grupo y una subclase, y `JURSTPREL` asigna la jurisdicción y el grupo de jurisdicciones a la misma subclase.

Modalidades de envío

Una modalidad de envío es la combinación de una empresa de transportes y su servicio de envío. Por ejemplo, Transportes XYZ, Servicio urgente es una modalidad de envío. La información sobre las modalidades de envío se almacena en la tabla `SHIPMODE`.

Códigos de cálculo

Los códigos de cálculo se utilizan para calcular descuentos, costes de envío, el impuesto sobre la venta y el impuesto sobre el envío. El archivo `shipping.xml` contiene todos los códigos de cálculo para el envío. La tabla `CALCODE` define los códigos de cálculo para el envío. El campo `displaylevel` muestra un número que visualiza la cantidad que se ha calculado.

- 0 = Artículo de pedido
- 1 = Pedido
- 2 = Producto
- 3 = Artículo
- 4 = Contrato

Normas de cálculo

Cada código de cálculo tiene un conjunto de normas de cálculo que definen cómo se realizará el cálculo. Por ejemplo, si va a enviar mercancías a una región, puede aplicar ciertas normas al cálculo, y si va a enviar mercancías a otra región, puede aplicar normas diferentes al cálculo. La tabla CALRULE almacena las normas de cálculo para el envío. El campo flag especifica si se debe invocar el método CalculationCodeQualifyMethod del Código de cálculo específico.

0 = El método no se invocará.

1 = El método se invocará.

Escala de cálculo

Una escala de cálculo es el conjunto de rangos que se aplican al cálculo. Por ejemplo, para los costes de envío puede tener un conjunto de rangos de peso que correspondan cada uno a un coste determinado. Es decir, un producto que pese entre 0 y 5 kg puede tener un coste de envío de 10 euros y un producto que pese entre 5 y 10 kg puede tener un coste de envío de 15 euros. La tabla CALSCALE almacena el código de escala para el envío, uno para cada pedido y uno para cada artículo.

Nota: La tabla CALSCALE almacena el código de escala para las monedas si éstas se aplican al rango correspondiente.

Rango de cálculo

El rango para los códigos de escala se almacena en la tabla CALRANGE. calmethod_id_10 se utiliza para el envío por pedido y calmethod_id_11 para envío por artículo.

Búsqueda de cálculo

Los valores de búsqueda de cálculo son los valores asociados a la escala de cálculo. Los valores de búsqueda de cálculo para un producto que pese entre 0 y 5 kg y tenga un coste de envío de 10 euros y un producto que pese entre 5 y 10 kg y tenga un coste de envío de 15 euros serían 10 y 15 euros respectivamente. Hay un valor de búsqueda por moneda para cada ID de CALRANGE específico. La tabla CARLOOKUP define el ID de búsqueda y el valor.

Combinaciones de cálculo

Las normas de cálculo y los rangos de escala se combinan en la tabla CRULESCALE, tal como se muestra en el código siguiente: Las normas y los métodos de cálculo se combinan en la tabla STENCALUSG. El valor por omisión de la tienda para los cálculos también se almacena en esta tabla. El campo usageflag controla cómo utiliza el cálculo el mandato OrderPrepare.

1 = use - utilizar este uso de cálculo.

2 = check - generar una excepción ECAApplicationException si este cálculo no produce un valor para un artículo de pedido.

Envío-despacho de pedidos

Los elementos de envío-despacho de pedidos asocian un grupo de jurisdicciones de envío a las normas de cálculo, y un centro de despacho de pedidos a la modalidad de envío (shipmode) de la tienda. La información de envío-despacho de pedidos se almacena en las tablas SHPJCRULE y SHPARRANGE y puede verse a continuación tal como aparece en shipfulfill.xml.

Elementos de base de datos de control de acceso de las tiendas de ejemplo

En WebSphere Commerce, todas las tiendas tienen políticas de control de acceso. Una política de control de acceso autoriza a los usuarios, o a un grupo de usuarios, a realizar acciones determinadas. Hay dos archivos de políticas de control de acceso para cada tienda:

- *nombretiendaejemploAccessPolicies.xml*
- *nombretiendaejemploAccessPolicies_locale.xml*

Tanto *AccessPolicies.xml* como *AccessPolicies_locale.xml*, son archivos de control de acceso nativo de nivel general. *AccessPolicies.xml* es independiente del idioma nacional, mientras que *AccessPolicies_locale.xml* es dependiente del idioma nacional. Cada archivo incluye las posibles acciones, recursos, grupos de acciones y definiciones de política utilizados en las tiendas de ejemplo. Estos archivos se transforman en los archivos *AccessPoliciesOut.xml* y *AccessPoliciesOut_locale.xml*, respectivamente. Cada archivo transformado se inserta en la base de datos. Para obtener más información sobre cómo transformar los archivos de políticas de acceso, consulte los enlaces relacionados que aparecen al final de la página.

Nota: Sólo los archivos transformados pueden cargarse masivamente o utilizarse directamente en el archivo SAR, no así los archivos transformados previamente.

La información sobre los elementos de base de datos para las tiendas de ejemplo puede dividirse en las siguientes secciones:

- Acciones
- Categorías de recursos
- Grupos de recursos
- Grupos de acciones
- Definiciones de políticas

Acciones

Las acciones que se pueden realizar bajo la política de control de acceso se definen en el archivo *AccessPolicies.xml* de cada tienda de ejemplo.

Categorías de recursos

Las categorías de recursos definen recursos que se pueden proteger.

Grupos de recursos

Los grupos de recursos contienen los recursos que controla la política de control de acceso. Un grupo de recursos puede incluir objetos de negocio, como un “contrato” o una “propuesta de intercambio”, o un conjunto de mandatos relacionados. El archivo *AccessPolicies.xml* de cada tienda de ejemplo define los grupos de recursos de la política.

Grupos de acciones

Los grupos de acciones definen qué acciones se pueden realizar en los grupos de recursos de la política de control de acceso. Estos grupos se definen en el archivo `AccessPolicies.xml` de cada tienda.

Definiciones de políticas

La política de cada tienda de ejemplo se define en el archivo `AccessPolicies.xml` de cada tienda individual. La tienda de ejemplo ToolTech tiene dos políticas.

Capítulo 3. Tienda de ejemplo NewFashion

Tienda de ejemplo NewFashion

NewFashion es una de las tiendas de moda en línea de empresa a consumidor que se proporciona con WebSphere Commerce. Implementa muchas de las características utilizadas más comúnmente en los sitios minoristas de primera fila de la actualidad. Algunas de las características incluidas en la tienda de ejemplo NewFashion son:

- Soporte multicultural
- Fechas de disponibilidad para artículos de pedido
- Pedidos en espera de artículos que no están actualmente en almacén
- División de pedidos basándose en la preferencia del cliente
- Seguimiento de estado del pedido
- Notificación de correo electrónico del estado del pedido
- Posibilidad de búsqueda
- Colaboración mediante soporte al cliente en tiempo real

La tienda NewFashion proporciona todas las páginas y características necesarias para una tienda en línea en funcionamiento. NewFashion está empaquetado con WebSphere Commerce como un archivador de tienda y, por lo tanto, no es necesario realizar ninguna instalación adicional. Lo único que necesita realizar para ver la tienda de ejemplo es crear un archivador de tienda nuevo basado en NewFashion utilizando las herramientas de Servicios de tienda y, a continuación, publicarlo en WebSphere Commerce Server. Para obtener más información, consulte [Crear un archivador de tienda utilizando los Servicios de tienda](#).

Dado que la creación de tienda en WebSphere Commerce se basa en la selección de un archivador de tienda de ejemplo y en su modificación, NewFashion está diseñado para actuar como base para la tienda. Se basa en un flujo de compra simple, aunque comprobado, y todas las páginas de la tienda de ejemplo pueden personalizarse fácilmente.

Para obtener más información sobre el flujo de compra de NewFashion, consulte el diagrama de flujo de compra de la tienda de ejemplo y los ejemplos de uso. Los ejemplos de uso detallan el flujo de cada interacción de usuario en la tienda, por ejemplo el registro o la visualización de un producto. Para obtener más detalles técnicos sobre cómo funciona cada página, vea la información de consulta correspondiente de cada página.

Capítulo 4. Páginas de la tienda NewFashion

Páginas de búsqueda de la tienda de ejemplo NewFashion

La página de resultados de la búsqueda, `resultlist.jsp`, se visualiza después de que un cliente ha entrado una petición de búsqueda desde la página de presentación, `sidebar.jsp`, o la página de búsqueda avanzada, `advancedsearch.jsp`.

Nota: La página `subcategory.jsp` no incluye `sidebar.jsp` como marco lateral de la página. Como resultado, la función de búsqueda se ha incorporado directamente en esta página y las características de búsqueda son las mismas que las de `sidebar.jsp`.

Beans

`resultlist.jsp` utiliza los beans siguientes:

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

`advancedsearch.jsp` utiliza los beans siguientes:

- `CatalogDataBean`
- `CategoryDataBean`

Mandatos

`resultlist.jsp` utiliza los mandatos siguientes:

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

`advancedsearch.jsp` utiliza los mandatos siguientes:

- `CatalogSearchResultView`

`sidebar.jsp` utiliza los mandatos siguientes:

- `CatalogSearchResultView`
- `AdvancedSearchView`
- `LogonForm` (sólo Búsqueda)
- `HelpView` (sólo Búsqueda)
- `StoreCatalogDisplay` (sólo Búsqueda)
- `SetCurrencyPreference` (sólo Búsqueda)

`subcategory.jsp` utiliza los mandatos siguientes:

- `CatalogSearchResultView`
- `AdvancedSearchView`

Nota: La página `subcategory.jsp` no incluye `sidebar.jsp` como marco lateral de la página. Como resultado, la función de búsqueda se ha incorporado directamente en esta página y las características de búsqueda son las mismas que las de `sidebar.jsp`.

Implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando el cliente pulsa **IR** desde el marco de navegación izquierdo, `sidebar.jsp`, o `subcategory.jsp`, o pulsa **Someter** desde la página de búsqueda avanzada `advancedsearch.jsp`, se visualiza `resultlist.jsp`.

Cuando `CatEntrySearchListDataBean` se activa, va a la base de datos y recupera todos los resultados de la búsqueda, tal como se muestra en el código siguiente.

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

El parámetro `request` es la serie URL que se pasa a la página `resultlist.jsp`. En el bean `CatEntrySearchListDataBean`, el método `getResultList()` devuelve una matriz de `CatalogEntryDataBean`. Cada `CatalogEntryDataBean` es un resultado que se lista en la página de resultados de la búsqueda. El método `getResultList()` devuelve un número específico de resultados por página. Este número se especifica en la dirección de la petición URL con la variable `pageSize` cuando `CatEntrySearchListBean` se activa. Cuando `result.jsp` se visualiza y los clientes pulsán los botones **Anterior** o **Siguiente**, `CatalogSearchResultView` lista los resultados y el parámetro URL `beginindex` se pasa a `CatalogSearchResultView`. `CatalogSearchResultView` indica a `CatEntrySearchListDataBean` en qué número de resultado de la lista de resultados de la búsqueda debe empezar la visualización.

En la tienda de ejemplo `NewFashion` sólo se muestran productos en los resultados de la búsqueda. Cuando se llama al mandato `CatalogSearchResultView`, la variable `resulttype` se pasa al mandato como un argumento de URL. `resulttype` indica a `CatEntrySearchListBean` que liste productos, artículos, o ambas cosas. Los valores de `resulttype` son:

- 1 = Listar sólo artículos.
- 2 = Listar sólo productos.
- 3 = Listar artículos y productos.

El código siguiente lista sólo productos.

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

Limitaciones

El catálogo de la tienda sigue una estructura jerárquica con categorías superiores, como Caballeros, Señoras y Novedades, y subcategorías por debajo de ellas, como Camisas y Pantalones.

Cuando los clientes seleccionan **Búsqueda avanzada**, sólo pueden buscar bajo la categoría superior o subcategoría específica elegida. Si los clientes buscan bajo la categoría Caballeros, sólo se visualizarán los productos que estén directamente bajo

esa categoría. Los productos bajo la subcategoría de Caballeros, Camisas o Pantalones, no se visualizarán. Para buscar en la subcategoría Pantalones o Camisas, los clientes deben seleccionar esa subcategoría específica para que se puedan visualizar resultados de búsqueda.

Páginas de listín de la tienda de ejemplo

Las páginas de listín de la tienda de ejemplo permiten a los clientes registrados añadir direcciones, incluidas las direcciones de envío y facturación, a un listín.

El cliente registrado se conecta y pulsa **Editar mi listín** en la página Mi cuenta (`account.jsp`). La página Listín (`addressbookform.jsp`) se visualiza y en ella el cliente puede añadir una nueva dirección o editar una existente. Para obtener más información, consulte el Ejemplo de uso de Añadir nueva dirección y el Ejemplo de uso de Editar una dirección.

Los procedimientos de listín de la tienda de ejemplo utilizan los siguientes archivos JSP:

- `account.jsp` (página Mi cuenta)
- `addressbookform.jsp` (página Listín)
- `addressform.jsp` (incluye los parámetros para el mandato `AddressForm`. No se muestra al cliente.)
- `address.jsp` (página Añadir dirección y página Actualizar dirección)
Nota: Se utiliza `address.jsp` para la página Añadir dirección y para la página Actualizar dirección. Si se proporciona el `addressId`, `address.jsp` se carga como una página Actualizar dirección. De lo contrario, se carga como una página Añadir dirección. Si el `addressId` se proporciona como un parámetro para el mandato `AddressAdd`, el mandato actualiza la dirección del `addressId` especificado. De lo contrario, se crea una nueva dirección.

Mandatos

`account.jsp` utiliza los mandatos siguientes:

- `UserRegistrationForm`
- `AddressBookForm`

`addressbook.jsp` utiliza los mandatos siguientes:

- `AddressForm`
- `AddressDelete`

`address.jsp` utiliza los mandatos siguientes:

- `AddressAdd`
- `PrivacyView`

Beans

`addressbook.jsp` utiliza los beans siguientes:

- `AddressAccessBean`

`address.jsp` utiliza los beans siguientes:

- `ErrorDataBean`
- `AddressDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas InFashion: técnicas de implementación comunes.

Cuando un cliente pulsa **Editar mi listín** en la página Mi cuenta, se llama al mandato AddressBookForm. A continuación, AddressBookForm carga la página Listín (addressbook.jsp). Si un cliente pulsa **Añadir una nueva dirección** se llama al mandato AddressForm. AddressForm está registrado con AddressForm.jsp en la base de datos y comprueba el parámetro page. Si page está establecido en newshipaddress, se carga la página Pasar por caja 1: Añadir dirección de facturación billingaddress.jsp, de lo contrario se carga la página Añadir dirección (address.jsp).

Nota: AddressForm cargará la página billingaddress.jsp si está establecido en newshipaddress debido al manejo de errores correspondiente a añadir dirección de facturación. Si page es igual a newshipaddress, significa que los clientes estaban creando una nueva dirección desde la página de dirección de facturación e intentaron ir a la página de dirección de envío pero hubo un error. Como resultado, se devuelve a los clientes a la página Dirección de facturación.

Una vez que el cliente ha cumplimentado los campos de la página Añadir dirección (address.jsp), comprueba si existe el addressId. Si el addressId existe, se actualiza el listín; de lo contrario, se crea una nueva dirección.

Si se proporciona el addressId, address.jsp se carga como una página Actualizar dirección. De lo contrario, se carga como una página Añadir dirección. En la página Actualizar dirección, los campos de entrada están cumplimentados de antemano con los valores entrados anteriormente, tal como se muestra a continuación:

Antes de crear una nueva dirección, se crea un apodo (un identificador exclusivo de la dirección, que incluya la hora y la fecha) para la dirección, utilizando el siguiente script Java:

Cuando un cliente cumplimenta la dirección y pulsa **Someter** tanto en la página Añadir dirección (address.jsp) como en la página Actualizar dirección (address.jsp), se llama al mandato AddressAdd. La página de listín (addressbook.jsp) muestra las direcciones existentes.

Manejo de errores

Si el cliente no cumplimenta los campos obligatorios en la página Añadir dirección (address.jsp) o en la página Actualizar dirección (address.jsp), el sistema le solicita que vuelva a escribir los campos.

Páginas de listín de la tienda de ejemplo NewFashion

Las páginas de listín de la tienda de ejemplo permiten a los clientes registrados añadir direcciones de envío y de facturación a un listín.

Un cliente registrado se conecta y luego pulsa **Editar mi listín** en la página Mi cuenta (myaccount.jsp). Se visualiza la página Listín (addressbookform.jsp). En ella, un cliente puede añadir una nueva dirección o editar una existente. Para

obtener más información, consulte el Ejemplo de uso de Añadir nueva dirección y el Ejemplo de uso de Editar una dirección.

El listín de la tienda de ejemplo utiliza los siguientes archivos JSP:

- `account.jsp` (página Mi cuenta)
- `addressbookform.jsp` (página Listín)
- `addressform.jsp` (incluye los parámetros para el mandato `AddressForm`. No se muestra al cliente.)
- `address.jsp`(página Añadir dirección y página Actualizar dirección)
Nota: Se utiliza `address.jsp` para la página Añadir dirección y para la página Actualizar dirección. Si se proporciona `addressId`, `address.jsp` se carga como una página Actualizar dirección. De lo contrario, se carga como una página Añadir dirección. Si se proporciona `addressId` como un parámetro para el mandato `AddressAdd`, el mandato actualiza la dirección del `addressId` especificado. De lo contrario, se crea una nueva dirección.

Mandatos

`account.jsp` utiliza los mandatos siguientes:

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

`addressbookform.jsp` utiliza los mandatos siguientes:

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

`address.jsp` utiliza los mandatos siguientes:

- `AddressAdd`
- `PrivacyView`

Beans

`addressbook.jsp` utiliza los beans siguientes:

- `AddressAccessBean`
- `AddressBookDataBean`

`address.jsp` utiliza los beans siguientes:

- `ErrorDataBean`
- `AddressDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente pulsa **Editar mi listín** en la página Mi cuenta, se llama al mandato `AddressBookForm`. A continuación, `AddressBookForm` carga la página Listín (`addressbookform.jsp`). Si un cliente pulsa **Añadir una nueva dirección**, se llama al mandato `AddressForm`. `AddressForm` está registrado con `AddressForm.jsp`

en la base de datos y comprueba el parámetro page. Si page se establece en newshipaddress, se carga la página Pasar por caja 1: Añadir dirección de facturación (billingaddress.jsp); de lo contrario, se carga la página Añadir dirección (address.jsp).

Nota: AddressForm cargará la página billingaddress.jsp si está establecido en newshipaddress debido al manejo de errores correspondiente a añadir dirección de facturación. Si page corresponde a newshipaddress, significa que los clientes estaban creando una nueva dirección desde la página de dirección de facturación e intentaron ir a la página de dirección de envío pero hubo un error. Como resultado, se devuelve a los clientes a la página Dirección de facturación.

Una vez que el cliente ha cumplimentado los campos de la página Añadir dirección (address.jsp), el sistema comprueba si el addressId existe. Si el addressId existe, se actualiza el listín. Si no, se crea una nueva dirección. Si se proporciona addressId, address.jsp se carga como una página Actualizar dirección. De lo contrario, se carga como una página Añadir dirección. En la página Actualizar dirección, los campos de entrada están cumplimentados de antemano con los valores entrados anteriormente, tal como se muestra a continuación:

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

Cuando un cliente cumplimenta la dirección y pulsa **Someter** tanto en la página Añadir dirección (address.jsp) como en la página Actualizar dirección (address.jsp), se llama al mandato AddressAdd. La página de listín (addressbookform.jsp) muestra las direcciones existentes.

Los clientes pueden pulsar **Editar listín** desde la página Seleccionar dirección de envío para ir al Listín. Si los clientes van a la página Listín desde la página Seleccionar dirección de envío, la página Listín muestra un enlace **Volver a Pasar por caja**. A estos clientes no se les permite eliminar una dirección desde la página Listín. Como resultado, se añade nuevo código para comprobar si los clientes vienen de la página Seleccionar dirección de envío, tal como se muestra a continuación:

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

Manejo de errores

Si el cliente no cumplimenta campos obligatorios en la página Añadir dirección (address.jsp) o la página Actualizar dirección (address.jsp), el sistema solicita al cliente que vuelva a escribir los campos. El código siguiente maneja el error:

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

if (hshErrorProperties != null)
{
//Existe un error en la dirección sometida.
strErrorCode = hshErrorProperties.getString(ECConstants.EC_ERROR_CODE, "");
if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_NICKNAME))
strErrorMessage = infashiontext.getString("ERROR_MESSAGE1");
```

```
...  
}
```

Páginas de visualización de paquete compuesto de las tiendas de ejemplo WebFashion y NewFashion

Una página de visualización de paquete compuesto contiene grupos promocionales de productos de una tienda en línea. Normalmente, incluye una descripción, una lista de los componentes que constituyen el paquete compuesto, el precio de cada componente, una imagen, una lista de atributos (talla y color) si los productos que componen el paquete compuesto tienen variaciones y una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla).

Los clientes pueden añadir todos los productos del paquete compuesto al carro de la compra o a la lista de deseos con una sola pulsación del ratón. Cada producto del paquete compuesto se visualiza como una línea de detalle separada en el carro de la compra, a diferencia del paquete, que se visualiza como una sola línea de detalle para el paquete entero.

Para obtener más información sobre las páginas de visualización de paquetes compuestos de la tienda de ejemplo y cómo funcionan, consulte el Ejemplo de uso de Visualizar página de paquete compuesto.

Mandatos

`bundledisplay.jsp` utiliza los mandatos siguientes:

- `OrderItemAdd`
- `InterestItemAdd`

Beans

`bundledisplay.jsp` utiliza los beans siguientes:

- `BundleDataBean`
- `CompositeProductDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

`Bundledisplay.jsp` está registrado en la base de datos (en la tabla `DISPENTREL`) para mostrar todos los productos de la tienda. `Bundledisplay.jsp` muestra lo siguiente:

- Descripción (incluyendo una lista de los componentes que forman el paquete compuesto) e imagen del paquete compuesto
- Una imagen, precio y una lista de atributos (talla y color) para cada producto del paquete compuesto. Además, una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla).
- El enlace Añadir al carro de la compra y el enlace Añadir a la lista de deseos
- El campo Cantidad (sólo NewFashion)

Descripción (incluyendo una lista de los componentes que forman el paquete compuesto) e imagen del paquete compuesto

El bean BundleDataBean recupera la descripción e imagen del paquete compuesto.

Imagen del producto, precio, atributos y valores de atributo

BundleDataBean llama a ProductDataBean para que recupere la información sobre cada uno de los productos que componen el paquete compuesto. Para más información acerca de cómo funciona ProductDataBean, consulte Páginas de productos de la tienda de ejemplo.

Añadir al carro de la compra y Añadir a la lista de deseos

Para obtener más información, consulte Páginas de productos de la tienda de ejemplo.

El campo Cantidad (sólo NewFashion)

Hay un campo de cantidad para cada producto del paquete compuesto que permite al cliente especificar el número a añadir al carro de la compra para cada producto.

Páginas de grupo de catálogo de la tienda de ejemplo

Las páginas de grupo de catálogo ayudan a los clientes a navegar por los diversos departamentos o grupos de productos o servicios disponibles en una tienda. Las primeras páginas de grupo de catálogo dirigen a los clientes a las áreas en las que desean comprar y muestran normalmente el nombre de cada grupo de catálogo, una descripción breve y una imagen. Las páginas de grupo de catálogo subsiguientes reducen todavía más el tipo de producto seleccionado que el cliente desea examinar. La última página de grupo de catálogo de un recorrido de compras contiene enlaces con las páginas de productos.

Generalmente, existen tres tipos de páginas de grupo de catálogo:

- páginas que enlazan con páginas de grupo de subcatálogo (páginas de grupo de catálogo)
- páginas que enlazan con páginas de productos (páginas de lista de productos)
- páginas que enlazan tanto con páginas de grupo de subcatálogo como con páginas de productos

Para obtener más información sobre las páginas de grupo de catálogo de la tienda de ejemplo y cómo funcionan, consulte el Ejemplo de uso de Ver categoría de producto.

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Visualización de páginas de grupo de catálogo y visualización de páginas de lista de productos

Las tiendas de ejemplo contienen páginas de grupo de catálogo, que muestran las subcategorías de cada una de las categorías de nivel superior, (Caballeros y Señoras) así como un producto de oferta estelar, y páginas de lista de productos, que muestran una lista de los productos de cada una de las subcategorías (Pantalones y camisas).

Todos los hiperenlaces con las categorías de nivel superior contienen un

parámetro adicional denominado top que está establecido en Y. El archivo de página JavaServer `categorydisplay.jsp` está registrado en la base de datos (en la tabla DISPCGREL) como la página que muestra todas las categorías de la tienda. En esta página se comprueba la existencia del parámetro top. Si existe, la página incluye `topcategory.jsp`. De lo contrario, se incluye `subcategory.jsp`. `topcategory.jsp` es la página de grupo de catálogo y `subcategory.jsp` es la página de lista de productos.

Página de grupo de catálogo

`topcategory.jsp` recupera una lista de subcategorías utilizando el método `getSubCategories()` de `CategoryDataBean`. Las ofertas estelares que contiene `InFashion` se crearon en el catálogo y no como parte de una campaña con `WebSphere Commerce Accelerator`. Los productos de oferta estelar se añadieron al grupo de catálogo y se recuperaron posteriormente mediante `topcategory.jsp` utilizando el método `getProducts()` de `CategoryDataBean`. En `WebFashion` y `NewFashion`, la página `topcategory.jsp` de caballeros muestra un paquete compuesto, y la página `topcategory.jsp` de señoras muestra un paquete. El paquete compuesto se recupera utilizando el método `getBundles()` de `CategoryDataBean`, y el paquete se recupera utilizando el método `getPackages()` de `CategoryDataBean`.

Página de lista de productos

`subcategory.jsp` muestra una lista de todos los productos del grupo de catálogo y una lista de todas las categorías hermano (todas las demás categorías bajo la misma categoría de nivel superior) en la barra de navegación izquierda. Para cada producto, `subcategory.jsp` muestra la descripción breve, la imagen de tamaño normal y el precio del producto utilizando los métodos siguientes:

`getDescription().getShortDescription()`, `getCalculatedContractPrice()` y `getDescription().getFullImage()`. El ID de categoría padre es necesario para visualizar la lista de categorías hermano. Para recuperar el ID de categoría padre, se proporciona el parámetro `parent_category_rn` en el hipervínculo, el cual crea un `CategoryDataBean` para la categoría padre. Por omisión, `CategoryDataBean` recupera el ID de categoría del parámetro `CategoryId`. Sin embargo, en este caso, el ID se almacena en `parent_category_rn`. Como resultado, el ID de categoría debe establecerse explícitamente del modo siguiente:

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

A continuación se recupera una lista de productos que pertenecen a este grupo de catálogo utilizando el método `getProducts()` de `CategoryDataBean`.

Nota: En la tienda `NewFashion`, la página `subcategory.jsp` también contiene un enlace con la característica de colaboración `Atención al cliente` si ésta está habilitada en los Servicios de tienda. La página `subcategory.jsp` tiene este enlace añadido porque tiene su propio marco lateral y no incluye `sidebar.jsp` como las demás páginas.

Si los clientes pulsan **Conversación en directo con Asistencia al cliente**, se visualiza una ventana emergente en la pantalla, y los clientes pueden conversar en línea en tiempo real con un representante de servicio al

cliente. Este enlace sólo se visualizará si esta característica está habilitada en los Servicios de tienda. El enlace **Conversación en directo con Asistencia al cliente** está encerrado dentro del cuerpo de un par de códigos personalizados (el código `<flow:ifEnabled feature="customerCare">` y el código `</flow:ifEnabled>`), y se puede habilitar o inhabilitar en función de la opción que se seleccione en los Servicios de tienda. Siempre y cuando el código personalizado permanezca en su sitio, los Servicios de tienda se pueden utilizar para conmutar automáticamente entre un sitio que tenga soporte de colaboración y otro que no lo tenga, sin tener que modificar la página JavaServer. Para habilitar o inhabilitar permanentemente el soporte de colaboración en la página, los códigos personalizados y el enlace **Conversación en directo con Asistencia al cliente** pueden eliminarse de la página JavaServer pulsando **Aplicar permanentemente**, en la GUI de los Servicios de tienda. No es aconsejable eliminar ni alterar manualmente los códigos personalizados ni el código adyacente. En lugar de eso, utilice el botón **Aplicar permanentemente** de los Servicios de tienda.

Nota: El código personalizado adyacente al enlace **Conversación en directo con Asistencia al cliente** no se puede copiar en las páginas JavaServer de otras tiendas. Estos códigos están diseñados para que funcionen únicamente en la tienda que los contenía originariamente. Para obtener más información acerca de la colaboración, consulte los enlaces relacionados que se muestran a continuación.

Páginas de la tienda de ejemplo: técnicas de implementación comunes

La mayoría de las páginas de la tienda de ejemplo utilizan las siguientes técnicas de implementación. Esta página utiliza la tienda InFashion como ejemplo pero puede aplicarse a todas las tiendas de ejemplo. Para obtener más información sobre las técnicas específicas de una página individual, consulte el archivo de referencia de dicha página.

Contenido multicultural

Las páginas de la tienda de ejemplo también se utilizan para mostrar contenido multicultural. Es decir, el mismo conjunto de páginas puede utilizarse para diferentes entornos nacionales. La mayor parte del código que permite una visualización multicultural está en `getResource.jsp`. El archivo `getResource.jsp` efectúa lo siguiente:

- Recupera el entorno nacional actual del contexto del mandato y lo almacena como la variable `locale`.
- Recupera el directorio de la tienda y lo almacena como la variable `storeDir`.
- Recupera el nombre de la tienda y lo almacena como la variable `storeName`.
- Carga el archivo de propiedades específico del idioma utilizando la API `ResourceBundle` y lo almacena como la variable `infashiontext`.

Para que un archivo JavaServer Page pueda acceder a las variables anteriores, `getResource.jsp` debe incluirse en el archivo JavaServer Page, utilizando la acción `include` en tiempo de compilación.

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

Dado que `getResource.jsp` se incluye prácticamente en todos los archivos JavaServer Page de ejemplo, puede ejecutarse varias veces en una sola petición.

Para evitar la duplicación del trabajo, la mayor parte de la información que recupera esta página se almacena dentro del contexto de la petición. Por ejemplo:

```
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); }
```

Mensajes específicos de cada idioma

Los mensajes específicos de cada idioma como, por ejemplo, “Gracias por hacer su pedido” se almacenan en los archivos de propiedades del paquete de recursos. Estos archivos se encuentran en el directorio siguiente:

```
▶ NT unidad:\WebSphere\AppServer\installedApps\WC_Enterprise_App_
nombre_instancia.ear\wcstores.war\WEB-INF\classes\dirtienda
▶ 2000 unidad:\Archivos de programa\WebSphere\AppServer\installedApps\
WC_Enterprise_App_nombre_instancia.ear\wcstores.war\WEB-INF\
classes\dirtienda
▶ AIX /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_
nombre_instancia.ear/wcstores.war/WEB-INF/classes/dirtienda
▶ Solaris /opt/WebSphere/Appserver/installedApps/WC_Enterprise_App_
nombre_instancia.ear/wcstores.war/WEB-INF/classes/dirtienda
▶ 400 /QIBM/UserData/WebASAdv4/nombre_instancia_WAS/installedApps
/WC_Enterprise_App_nombre_instancia.ear/wcstores.war/WEB-INF/
classes/dirtienda
```

Por ejemplo, si el directorio de su tienda es “dirtienda”, el archivo de propiedades en español será:

```
▶ NT unidad:\WebSphere\AppServer\installedApps\WC_Enterprise_App_
nombre_instancia.ear\wcstores.war\WEB-INF\classes\dirtienda
\infashiontext_es_ES.properties.
▶ 2000 unidad:\Archivos de programa\WebSphere\AppServer\installedApps\
WC_Enterprise_App_nombre_instancia.ear\wcstores.war\WEB-INF\
classes\dirtienda\infashiontext_es_ES.properties.
▶ AIX /usr/WebSphere/AppServer/installedApps/
WC_Enterprise_App_nombre_instancia.ear/wcstores.war/WEB-INF/
classes/dirtienda/infashiontext_es_ES.properties.
▶ Solaris /opt/WebSphere/AppServer/installedApps/
WC_Enterprise_App_nombre_instancia.ear/wcstores.war/WEB-INF/
classes/dirtienda/infashiontext_es_ES.properties.
```

```
400 /QIBM/UserData/WebASAdv4/nombre_instancia_WAS/installedApps
/WC_Enterprise_App_nombre_instancia.ear/wcstores.war/WEB-INF/
classes/dirtienda/infashiontext_es_ES.properties
```

El contenido de este archivo se carga utilizando la API de Java `java.util.ResourceBundle` del archivo JSP `getResource.jsp`. El paquete se almacena como la variable `infashiontext`. Un mensaje específico del idioma se visualiza de la siguiente manera:

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

Establecer la codificación del contenido

La mayor parte de los navegadores modernos comprenden los datos HTML codificados en codificación UTF-8. Sin embargo, algunos de los navegadores antiguos sólo comprenden la codificación nativa. Por ejemplo, es posible que un navegador japonés antiguo sólo comprenda los datos HTML codificados en codificación "Shift_JIS". Para solucionar este problema, la tienda de ejemplo especifica la codificación específica del idioma en los archivos de propiedades del paquete de recursos como el nombre de propiedad `ENCODESTATEMENT`. Por ejemplo, el archivo de propiedades en inglés `infashiontext_en_US.properties` contiene la entrada siguiente:

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

Para los archivos JSP individuales, la codificación se establece utilizando el objeto de petición JSP como se muestra a continuación:

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

El tipo de codificación se ha de establecer en el inicio del archivo JSP, tanto como sea posible, ya que la cabecera HTTP se envía antes que cualquier contenido HTML. Por lo tanto, si establece el tipo de contenido (que se envía como cabecera HTTP) después de cualquier contenido HTML, es posible que el efecto no sea el deseado. Y es posible que el navegador no pueda mostrar los datos correctamente.

Inclusión de la cabecera, el pie de página y el marco de navegación izquierdo

Prácticamente todas las páginas de la tienda de ejemplo muestran las páginas de cabecera (`header.jsp`), pie de página (`footer.jsp`) y marco de navegación izquierdo (`sidebar.jsp`). Estas páginas se incluyen en los demás archivos JSP utilizando la siguiente directiva `include` en tiempo de ejecución:

```
<% String incfile;
```

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

Si conoce la ubicación exacta de los archivos JSP, puede simplificar el proceso de inclusión utilizando lo siguiente:

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

donde `header.jsp` está situado en el directorio `dirtienda` bajo el directorio raíz de documentos de la aplicación web.

Página Contáctenos de la tienda de ejemplo

Cuando un cliente pulsa el enlace **Contáctenos**, se visualiza la página Contáctenos (`contact.jsp`). Esta página muestra los diversos tipos de información de contacto de la tienda, incluyendo los números de teléfono y las ubicaciones de las tiendas “reales”.

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Se crea una nueva vista denominada `ContactView` en la tabla `VIEWREG` y se asocia con `contact.jsp`. El URL de la página de privacidad es:
`http://nombre_máquina/webapp/wcs/stores/servlet/ContactView?lista_parámetros`

Pie de página de la tienda de ejemplo

Las tiendas de ejemplo tienen un pie de página en la parte inferior de cada página (`footer.jsp`) que incluye enlaces con lo siguiente:

- Inicio
- Carro de la compra
- Mi cuenta
- Contáctenos
- Política de privacidad
- Ayuda

Mandatos

`footer.jsp` utiliza los mandatos siguientes:

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

Beans

`footer.jsp` utiliza los beans siguientes:

- `UserRegistrationDataBean`
- `ErrorDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Carro de la compra

El enlace **CARRO DE LA COMPRA** del pie de página enlaza con el mandato de controlador `OrderItemDisplay`, que devuelve el mandato de

vista `OrderItemDisplayViewShiptoAssoc`. El mandato de vista `OrderItemDisplayViewShiptoAssoc` está registrado con `OrderItemDisplay.jsp` en la base de datos. `OrderItemDisplay.jsp` carga `shoppingcart.jsp` para mostrar la página Carro de la compra.

Para obtener más información sobre mandatos, consulte Mandatos y la “Guía del programador de WebSphere Commerce”.

Mi cuenta

Si usted es un cliente registrado, al pulsar en **Mi cuenta** irá a la página Mi cuenta (`account.jsp`). Si no es un cliente registrado, al pulsar en **Mi cuenta** irá a la página Regístrese o conéctese (`myaccount.jsp`). Esto se lleva a cabo utilizando el código siguiente:

```
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>
```

Contáctenos

Al pulsar en Contáctenos se llama al mandato `ContactView`, que carga la página Contáctenos (`contact.jsp`).

Ayuda Al pulsar en Ayuda se llama al mandato `HelpView`, que carga la la página de Ayuda (`help.jsp`).

Política de privacidad

Al pulsar en Política de privacidad se llama al mandato `PrivacyView`, que carga la página de Política de privacidad (`privacy.jsp`).

Cabecera de la tienda de ejemplo

Las tiendas de ejemplo tienen una cabecera en la parte superior de cada página (`header.jsp`) que incluye enlaces con lo siguiente:

- Carro de la compra
- Mi cuenta
- Contáctenos
- Ayuda
- Caballeros
- Señoras
- Novedades

Mandatos

`header.jsp` utiliza los mandatos siguientes:

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

- CategoryDisplay

Beans

header.jsp utiliza los beans siguientes:

- UserRegistrationDataBean
- CatalogDataBean
- CategoryDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

CARRO DE LA COMPRA

El enlace **CARRO DE LA COMPRA** de la cabecera enlaza con el mandato OrderItemDisplay, que devuelve el mandato de vista OrderItemDisplayViewShiptoAssoc. El mandato de vista OrderItemDisplayViewShiptoAssoc está registrado con OrderItemDisplay.jsp en la base de datos. OrderItemDisplay.jsp carga shoppingcart.jsp para mostrar la página Carro de la compra.

MI CUENTA

Si usted es un cliente registrado, al pulsar **MI CUENTA** irá a la página Mi cuenta (account.jsp). Si usted no es un cliente registrado, al pulsar **MI CUENTA** irá a la página Regístrese o conéctese (myaccount.jsp). Esto se lleva a cabo utilizando el código siguiente:

```
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>
```

CONTÁCTENOS

Al pulsar **CONTÁCTENOS** se llama al mandato ContactView, que carga la página Contáctenos (contact.jsp).

AYUDA

Al pulsar **AYUDA** se llama al mandato HelpView, que carga la página de Ayuda (help.jsp).

Categorías de nivel superior (Caballeros, Señoras, Novedades)

Cuando un cliente pulsa una de las categorías de nivel superior de la cabecera, se llama al mandato CategoryDisplay. CategoryDisplay está registrado en la base de datos con CategoryDisplay.jsp. Cuando el parámetro top se establece en Y, como en el ejemplo siguiente, se carga topcategory.jsp, que muestra la página de categoría adecuada, del modo siguiente:

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

Página Ayuda de la tienda de ejemplo

La página Ayuda (`help.jsp`) se muestra cuando un cliente pulsa **Ayuda**.

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Se crea una nueva vista denominada `HelpView` en la tabla `VIEWREG` y se asocia con `help.jsp`. El URL de la página de privacidad es:
`http://nombre_máquina/webapp/wcs/v5/stores/HelpView?lista_parámetros`.

Página de presentación de la tienda de ejemplo

La página de presentación (`storecatalogdisplay.jsp`) actúa como el escaparate que atrae clientes a su tienda. La página de presentación de la tienda de ejemplo muestra todas las categorías de nivel superior de la tienda, tales como Señoras y Caballeros, y promociona algunas ofertas especiales. Para obtener más información sobre la página de presentación de la tienda de ejemplo, consulte el Ejemplo de uso de Página de presentación.

Mandatos

`storecatalogdisplay.jsp` utiliza los mandatos siguientes:

- `CategoryDisplay`
- `ProductDisplay`

Beans

`storecatalogdisplay.jsp` utiliza los beans siguientes:

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

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

`storecatalogdisplay.jsp` lo inicia la página `index.jsp`, que proporciona un URL para invocar la página de presentación de la tienda de ejemplo. `index.jsp` llama al archivo `parameters.jsp`, que contiene los parámetros necesarios para iniciar la tienda. `storecatalogdisplay.jsp` muestra lo siguiente:

Categorías superiores

Las categorías superiores están registradas en la tabla `CATTOGRP`.

`storecatalogdisplay.jsp` recupera las categorías superiores con el método `getTopCategories()` de `CatalogDataBean`.

Ofertas especiales

Las ofertas estelares que contiene `InFashion` y `NewFashion` se crearon en el

catálogo en vez de con WebSphere Commerce Accelerator. Para crear las promociones, se ha añadido al catálogo una categoría de nivel superior especial, con el identificador `HOMEPAGE_PROMO`. Los productos pertenecientes a esta categoría se visualizarán como ofertas especiales.

En WebFashion, los productos objetivo de la página de presentación forman parte de una campaña. La campaña se ha creado utilizando zonas de e-Marketing y WebSphere Commerce Accelerator. La zona de e-Marketing de la página de presentación se denomina `StoreHomePage` y se utiliza para crear promociones basadas en el sexo al que pertenecen los clientes. Para poder activar la zona de e-Marketing, debe crear una campaña utilizando WebSphere Commerce Accelerator.

Los clientes invitados y los clientes registrados que no proporcionaron información sobre el sexo al que pertenecen verán los productos por omisión (`HOMEPAGE_PROMO`).

Nota: Los productos de la categoría Promoción de la página de presentación o la categoría Novedades aparecen sólo en las categorías especiales. Cuando las promociones terminen, estos productos deben trasladarse a su categoría habitual.

Páginas de conexión de la tienda de ejemplo

Las páginas de conexión de la tienda de ejemplo permiten que los clientes registrados se conecten.

Cuando los clientes registrados pulsán el enlace **Regístrese ahora y reciba avisos anticipados de nuestras promociones** en la barra lateral, se visualiza la página `Regístrese o conéctese (account.jsp)`. Después de que los clientes proporcionan sus direcciones de correo electrónico y contraseñas y pulsán **Conexión**, se visualiza la página `Mi cuenta (myaccount.jsp)`. Para obtener más información, consulte el Ejemplo de uso de Conexión.

El procedimiento de conexión de la tienda de ejemplo utiliza los siguientes archivos JSP:

- `account.jsp` (La página **Regístrese o conéctese**)
- `myaccount.jsp` (La página **Mi cuenta**)
- `LoginForm.jsp` (Incluye los parámetros para el mandato `Logon`. No se muestra al cliente.)
- `Logoff.jsp` (Incluye los parámetros para el mandato `Logoff`, No se muestra al cliente.)
- `forgetpassword.jsp` (La página **Contraseña olvidada**)
- `ResetPasswordForm.jsp` (Incluye los parámetros del mandato `ResetPassword`. No se muestra al cliente.)
- `password.jsp` (muestra un mensaje a los clientes indicándoles que se les ha enviado la contraseña)
- `ChangePasswordForm.jsp` (La página **Cambiar contraseña**)
- `ResetPasswordError.jsp` (Se invoca si hay un problema al restablecer la contraseña. No se muestra al cliente.)

Mandatos

`account.jsp` utiliza los mandatos siguientes:

- `Logon`
- `Logoff`

myaccount.jsp utiliza los mandatos siguientes:

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

forgetpassword.jsp utiliza los mandatos siguientes:

- Logoff
- ResetPassword

ChangePasswordForm.jsp utiliza los mandatos siguientes:

- ResetPassword

forgetpassword_err.jsp utiliza los mandatos siguientes:

- ResetPassword
- Logoff

password.jsp utiliza los mandatos siguientes:

- LogonForm

Beans

forgetpassword.jsp utiliza el bean siguiente:

- ErrorDataBean

forgetpassword_err.jsp utiliza los beans siguientes:

- ErrorDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Después de que los clientes se han registrado en la página de registro, o han escrito sus direcciones de correo electrónico y contraseñas en la página de conexión (account.jsp), los valores se convierten a minúsculas utilizando el código siguiente:

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

account.jsp también establece los campos que espera el mandato Logon, por ejemplo:

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

Cuando los clientes pulsán **Conexión** desde la página Regístrese o conéctese, se llama al mandato Logon. Logon está registrado en la base de datos con LoginForm.jsp. LoginForm.jsp utiliza el parámetro page para determinar qué página (myaccount.jsp oaccount.jsp) ha de cargar.

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

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

}
```

Si se ha especificado una combinación de dirección de correo electrónico y contraseña correcta, LoginForm.jsp carga la página Mi Cuenta (myaccount.jsp). Si se ha especificado una combinación de dirección de correo electrónico y contraseña incorrecta, LoginForm.jsp vuelve a cargar la página Regístrese o conéctese (account.jsp).

Si los clientes olvidan la contraseña y pulsán **¿Ha olvidado su contraseña?**, se llama al mandato Logoff, con el parámetro state=forgetpassword. El mandato Logoff está registrado con Logoff.jsp en la base de datos. Logoff.jsp comprueba el estado del parámetro, como se describe a continuación:

```
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";

}
```

Se carga la página Contraseña olvidada (forgetpassword.jsp) si el estado es igual a forgetpassword. Cuando los clientes cumplimentan los campos de la página y pulsán **Envíenme mi contraseña**, se llama al mandato ResetPassword. La contraseña antigua se establece como caducada en la base de datos y la contraseña nueva se envía por correo electrónico a los clientes. Cuando los clientes se conectan utilizando la contraseña nueva, se les obliga a cambiar la contraseña y se les envía a la página Cambiar contraseña (ChangePasswordForm.jsp).

Nota: Si las contraseñas de los clientes están establecidas para que que caduquen, se les dirigirá automáticamente a la página Cambiar contraseña la próxima vez que se conecten.

Manejo de errores

Si los clientes proporcionan una dirección de correo electrónico o una contraseña incorrecta, o no cumplimentan los campos, se visualiza un mensaje de error y el mandato Logon vuelve a cargar la página Regístrese o conéctese sin establecer el parámetro page. Si se ha entrado una contraseña incorrecta, los clientes tienen que esperar unos segundos antes de conectarse o se visualizará el error siguiente:

Espera unos segundos antes de intentar conectarse otra vez.

Página de privacidad de la tienda de ejemplo

La página de privacidad (Privacy.jsp) se visualiza cuando un cliente pulsa **Política de privacidad**.

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Se crea una nueva vista denominada PrivacyView en la tabla VIEWREG y se asocia con privacy.jsp. El URL para ver la página de privacidad es http://nombre_máquina/webapp/wcs/v5/stores/PrivacyView?lista_parámetros.

Nota: Debe crear su propia política de privacidad e incluirla en su tienda en línea. Para más información, consulte el sitio Web de privacidad de IBM <http://www.ibm.com/privacy/>.

Páginas de productos de la tienda de ejemplo

Una página de producto presenta un producto determinado de una tienda en línea. Normalmente, incluye una descripción, el precio y una imagen y, si el producto tiene atributos (por ejemplo, tallas y colores diferentes), permite a los clientes elegir un atributo.

Para obtener más información sobre las páginas de productos y cómo funcionan, consulte el Ejemplo de uso de Visualizar página de producto.

Mandatos

productdisplay.jsp utiliza los mandatos siguientes:

- OrderItemAdd
- InterestItemAdd (sólo WebFashion y NewFashion)

Beans

productdisplay.jsp utiliza los beans siguientes:

- CategoryDataBean
- ProductDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

ProductDisplay.jsp está registrado en la base de datos (en la tabla DISPENTREL) para mostrar todos los productos de la tienda. ProductDisplay.jsp muestra lo siguiente:

- Descripción, imagen, atributos y valores de atributos del producto.
- Descripción breve de la categoría padre
- Enlace Añadir al carro de la compra
- Cuadro de texto Cantidad (sólo NewFashion)
- Enlace Añadir a la lista de deseos (sólo WebFashion y NewFashion)

Descripción, imagen, atributos y valores de atributos del producto.

La descripción y la imagen del producto se visualizan utilizando las propiedades ProductDataBean.

Los atributos del producto se recuperan utilizando el método getAttributes() para ProductDataBean. Los valores de cada atributo se recuperan utilizando el método getDistinctAttributeValues() de AttributeAccessBean. Si ProductDataBean encuentra información en esta columna, la página de visualización del producto muestra la imagen Hotmedia en lugar de la imagen de tamaño normal.

Descripción breve de la categoría padre

El ID de la categoría padre se proporciona a las páginas de productos mediante el parámetro parent_category_rn. Mediante CategoryDataBean se recupera una descripción breve de la categoría padre. Por omisión, CategoryDataBean obtiene el ID de categoría del parámetro categoryId. En el ejemplo siguiente, el nombre del parámetro es parent_category_rn y el ID de categoría se establece explícitamente:

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

Añadir al carro de la compra

El enlace **Añadir al carro de la compra** se implementa creando un formulario que llama al mandato OrderItemAdd. En InFashion y WebFashion, la cantidad de productos solicitados se establece por omisión en 1 utilizando un campo oculto, como se muestra a continuación:

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

Puede sustituir el campo oculto por un cuadro de texto, de modo que los clientes puedan especificar una cantidad diferente.

En la tienda NewFashion, la cantidad de productos solicitados se establece por omisión en 1 utilizando un campo de texto, como se muestra a continuación:

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

El campo de texto permite a los clientes especificar una cantidad diferente.

Añadir al carro de la compra y Añadir a la lista de deseos

Cuando un cliente selecciona **Añadir al carro de la compra** o **Añadir a la lista de deseos**, se llama al siguiente javascript.

```
<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>
```

Cuando el cliente añade algo al carro de la compra, se llama al mandato `OrderItemAdd`. Cuando el cliente añade algo a la lista de deseos, se llama al mandato `InterestItemAdd`.

Nota: Puede utilizar WebSphere Commerce Accelerator para crear productos. Cuando cree productos, debe crear un precio para el producto para verlo en las páginas de productos de la tienda de ejemplo. Si no crea un precio, debe quitar el método `getCalculatedContractPrice` para el bean `ProductDataBean`.

Página de registro de la tienda de ejemplo

La página de registro de la tienda de ejemplo permite a los usuarios registrarse en la tienda de ejemplo. Cuando los clientes se registran, deben proporcionar su nombre y apellido y una dirección de correo electrónico y, a continuación, crear una contraseña.

Cuando los clientes pulsan **Regístrese ahora y compre**, se muestra la página **Regístrese** o **conéctese**. A continuación, los clientes pulsan **Regístrese** y se muestra la página de Registro (`register.jsp`).

Cuando los clientes desean actualizar su información de registro, pulsan **Modificar información personal** en la página **Mi cuenta**. Se visualiza la página **Modificar información Personal** (`edit_registration.jsp`).

Mandatos

`register.jsp` utiliza los mandatos siguientes:

- `UserRegistrationAdd`
- `PrivacyView`

Beans

`register.jsp` utiliza los beans siguientes:

- `ErrorDataBean`

`edit_registration.jsp` utiliza los beans siguientes:

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

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Registrarse

El enlace **Regístrese** de la página Regístrese o conéctese (`account.jsp`) muestra el formulario de registro. WebSphere Commerce utiliza el mandato `UserRegistrationAdd` para crear un nuevo registro. Si el cliente está conectado, `UserRegistrationAdd` se comporta como `UserRegistrationUpdate`. Es decir, un cliente conectado no puede volver a registrarse y crear una cuenta nueva. Dado que la tienda de ejemplo permite al usuario registrarse varias veces y crear varias cuentas, un cliente conectado debe desconectarse primero antes de volver a registrarse. Para conseguir este resultado, el enlace Regístrese se establece automáticamente en el mandato `Logoff` para los clientes conectados y en la vista `UserRegistrationForm` para los clientes invitados. El mandato `Logoff` llama automáticamente a la tarea `LogoffView`. La tarea `LogoffView` está registrada con `Logoff.jsp` en la base de datos. `Logoff.jsp` comprueba el parámetro URL `state`. Si está establecido en `forgetpassword`, se carga `forgetpassword.jsp`; si no, se carga `UserRegistrationForm.jsp`. La vista `UserRegistrationForm` está asociada al archivo `UserRegistrationForm.jsp` en la tabla `VIEWREG`. `UserRegistrationForm.jsp` comprueba el parámetro URL `new`. Si está establecido en `Y`, se incluye `register.jsp` para visualizar el formulario de nuevo registro. De lo contrario, se incluye `edit_registration.jsp` para visualizar el formulario de actualización de registro.

`register.jsp` contiene un formulario de nuevo registro. El formulario de nuevo registro se somete al mandato `UserRegistrationAdd`. Por omisión, el mandato `UserRegistrationAdd` espera varios campos obligatorios que no son necesarios para la tienda de ejemplo. Como resultado, estos campos se configuran como campos HTML ocultos y los valores se establecen en `-`, como en el ejemplo siguiente:

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

Nota: En la tienda `NewFashion` no se somete la pareja nombre-valor. Si la pareja nombre-valor no se somete, el mandato `UserRegistrationAdd` no comprobará el parámetro URL.

Cuando los clientes se registran y rellenan su perfil, éste puede coincidir con uno de los perfiles de WebSphere Commerce Accelerator. Si el perfil de WebSphere Commerce Accelerator tiene asociado un descuento, los clientes recibirán ese descuento.

Dirección de correo electrónico como conexión

Las tiendas de ejemplo requieren que los clientes escriban su dirección de correo electrónico como su ID de conexión, pero WebSphere Commerce requiere un ID de usuario como ID de conexión. Como solución, la tienda de ejemplo solicita a los clientes que escriban su dirección de correo electrónico en el formulario de registro. A continuación, antes de someter el formulario, el campo de dirección de correo electrónico y el campo de ID de usuario se establecen ambos en el valor de la dirección de correo electrónico, utilizando el siguiente 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()
}

```

Modificar información personal

Al pulsar **Modificar información personal** se llama al mandato `UserRegistrationForm`, que está asociado en la base de datos con `UserRegistrationForm.jsp`. Si el parámetro `new` no es nulo, cargará `register.jsp`. De lo contrario, cargará `edit_registration.jsp`. El mandato `UserRegistrationUpdate` se utiliza en `edit_registration.jsp` para actualizar la información de registro del usuario. Si el cliente no rellena los campos de contraseña o de verificar contraseña, el sistema proporcionará la contraseña de registro, utilizando el código siguiente:

```

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()
}

```

Si un cliente ha proporcionado previamente información sobre el sexo y la edad, los campos de sexo y edad estarán cumplimentados de antemano. `DemographicsAccessBean` extrae la información sobre sexo y edad de la base de datos.

Manejo de errores

Ante cualquier error, el mandato `UserRegistrationAdd` llama a `UserRegistrationErrorView`, el cual está registrado en la base de datos con `UserRegistrationForm.jsp`. Sin embargo, `UserRegistrationUpdate` también invoca la misma vista de error. Para diferenciar un formulario de nuevo registro de un formulario de edición, el formulario de nuevo registro contiene un parámetro oculto denominado `new`. Si dicho parámetro está presente, `UserRegistrationForm.jsp` incluye `register.jsp`. De lo contrario, carga `edit_registration.jsp`.

register.jsp se utiliza tanto en condiciones de error como en condiciones normales. ErrorDataBean y la comprobación de errores se utilizan para determinar bajo qué condición se ejecuta register.jsp. Si existe un error, register.jsp muestra el mensaje de error.

Nota: Aunque los errores de contraseñas los detectan los mandatos UserRegistrationAdd y UserRegistrationUpdate, estos llaman a AuthenticationPolicyErrorView cuando se produce un error de contraseña.

Página Novedades de la tienda de ejemplo

La página Novedades (newarrivals.jsp) se visualiza cuando el cliente pulsa **Novedades**.

Mandatos

newarrivals.jsp utiliza los mandatos siguientes:

- ProductDisplay

Beans

newarrivals.jsp utiliza los beans siguientes:

- CategoryDataBean
- ProductDataBean
- CatalogDataBean
- EMarketingSpotBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

La función de la página Novedades (newarrivals.jsp) es mostrar una lista de artículos en promoción. Las Novedades que contiene InFashion se crearon en el catálogo. Como resultado, se creó una categoría de nivel superior denominada Novedades y todos los productos en promoción se añadieron a esta categoría. Dado que InFashion muestra el contenido de esta categoría de forma diferente al resto de las demás categorías de nivel superior, el archivo newarrivals.jsp está registrado en la base de datos (en la tabla DISPCGPREL) para que muestre esta categoría. Cuando se llama al mandato CategoryDisplay con este ID de categoría, se carga newarrivals.jsp para visualizar los productos que se han designado como novedades.

La página newarrivals.jsp utiliza CategoryDataBean para mostrar información sobre las categorías. Para NewFashion e InFashion, los productos de esta página proceden de una categoría especial que tiene el identificador “What’s Hot”. La lista de productos en promoción se recupera utilizando el método getProducts() de CategoryDataBean.

En WebFashion, la página Novedades muestra la campaña Superofertas. El objetivo de esta campaña es vender los artículos más de moda de la temporada. La campaña Superofertas utiliza una iniciativa de venta por sugerencia, que está dirigida a los clientes registrados y que sugiere artículos en base a la edad del cliente. Los clientes registrados que tienen más de 29 años ven un conjunto de tres

productos, y los clientes menores de 29 años ven otro conjunto. Los clientes invitados y los clientes registrados que no proporcionaron información sobre la edad verán los productos por omisión (“What’s Hot”).

La campaña Superofertas se ha creado utilizando WebSphere Commerce Accelerator y una zona de e-Marketing. La zona de e-Marketing se llama NewArrivalsPage y está situada en la página Novedades. Para activar la zona de e-marketing, debe configurar una campaña en WebSphere Commerce Accelerator.

Página de confirmación del pedido de la tienda de ejemplo NewFashion

Una vez que el cliente completa el pedido en el cuarto paso del proceso de pasar por caja de la tienda de ejemplo, el cliente pulsa **Realizar pedido ahora**. Se visualiza la página Confirmación del pedido (`confirmation.jsp`), que permite al cliente ver el número de pedido, subtotal, total de impuestos, gastos de envío, total general y precio para todo el pedido. Si se procesan dos pedidos, la información de ambos pedidos se visualizará en la página de confirmación.

Para obtener más información, consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Beans

`confirmation.jsp` utiliza los beans siguientes:

- `PayStatusPMDDataBean`
- `OrderDataBean`

Implementación

El bean `PayStatusPMDDataBean` obtiene del sistema la información sobre el pedido. El bean `OrderDataBean` toma la información, le da formato y la presenta al sistema.

Página de notificación de correo electrónico de WebFashion y NewFashion

Después de que un cliente registrado haya completado un pedido pulsando **Realizar pedido ahora** (para más detalles vea el Ejemplo de uso de Pasar por caja el carro de la compra), y que Payment Manager haya autorizado el pedido, el sistema envía un correo electrónico al cliente, `OrderAuthorized.jsp`, indicándole que se ha aceptado el pedido y que se autoriza el pago.

En la tienda NewFashion se envían los siguientes avisos de correo electrónico:

- Correo electrónico de pedido cancelado (`OrderCanceledNotification.jsp`). El pedido no puede completarse porque la tarjeta de crédito ha sido rechazada.
- Correo electrónico de notificación de envío (`ReleaseShipNotify.jsp`). El pedido se ha enviado al comprador.
- Correo electrónico de sometimiento de pedido (`OrderReceived.jsp`). Después de que un cliente registrado ha completado un pedido pulsando **Realizar pedido ahora**, se envía el correo electrónico de sometimiento de pedido.

Beans

OrderAuthorized.jsp utiliza los beans siguientes:

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

Nota: Estos beans se utilizan solamente en la tienda NewFashion.

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Para recuperar la información sobre el pedido del cliente, el sistema utiliza OrderDataBean. La información de la tienda se recupera utilizando StoreDataBean. StoreEntityDescriptionAccessBean proporciona al sistema el nombre de la tienda. AddressDataBean proporciona al sistema la dirección de envío.

Página de resumen del pedido de la tienda de ejemplo

En el cuarto paso del proceso de pasar por caja de la tienda de ejemplo, la página Pasar por caja 4. Resumen del pedido (orderdisplaypending.jsp), los clientes pueden revisar la información detallada sobre su pedido, que incluye una descripción de los artículos adquiridos, así como la cantidad, precio unitario y precio total, dirección de envío y gastos de envío. A continuación, los clientes deben completar el pedido proporcionando la información de pago y pulsando **Realizar pedido ahora**. En la tienda NewFashion, se muestra la fecha de envío estimada.

Para obtener más información, consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Mandatos

orderdisplaypending.jsp utiliza los mandatos siguientes:

- OrderProcess
- PrivacyView
- ContactView
- MultiOrderProcess (sólo NewFashion) (MultiOrderProcess está asociado con MultiOrderProcess.jsp)

Beans

orderdisplaypending.jsp utiliza los beans siguientes:

- OrderDataBean
- OrderItemDataBean
- AddressDataBean
- ErrorDataBean
- ShippingModeDescriptionDataBean

- UsablePaymentTCListDataBean

Detalles sobre implementación

Sólo NewFashion

Cuando los clientes pulsán **Realizar pedido ahora**, se llama al mandato `MultiOrderProcess`. `MultiOrderProcess` es un mandato de vista que está registrado en la tabla `VIEWREG` y está asociado con `MultiOrderProcess.jsp`. `MultiOrderProcess.jsp` ejecuta `OrderProcess` varias veces dependiendo del número de pedidos en la página de resumen de pedidos. En `NewFashion`, el flujo de pasar por caja sólo permite un máximo de dos pedidos en la página de resumen de pedidos.

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Para todas las tiendas de ejemplo

Cuando los clientes inician el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplazan por una serie de páginas de pasar por caja, la cuarta de las cuales es la página Pasar por caja 4. Resumen del pedido (`orderdisplaypending.jsp`). Esta página incluye un formulario que permite a los clientes someter la información de su tarjeta de crédito. Se utiliza el bean de datos `UsablePaymentTCListDataBean` para obtener de `Payment Manager` los nombres de las tarjetas de crédito disponibles, y la acción para el formulario se establece en `OrderProcess`. Una vez sometido el formulario, y si el proceso del pedido se realiza satisfactoriamente, se llama a `OrderOKView`. El mandato `OrderOKView` está registrado en la tabla `VIEWREG` de la base de datos y está asociado con `confirmation.jsp`, que muestra información para confirmar el pedido. En la tienda `NewFashion`, si se procesan dos pedidos, la información de ambos pedidos se mostrarán en la página de confirmación.

Si se produce un error, se llama a `DoPaymentErrorView`. `DoPaymentErrorView` está asociado en la base de datos con `OrderDisplayPending.jsp`. Como resultado, cuando se produce un error, la página Pasar por caja 4. Resumen del pedido (`orderdisplaypending.jsp`) se vuelve a visualizar con un mensaje de error.

Página de comprobación de disponibilidad de productos de la tienda de ejemplo NewFashion

En el tercer paso, parte 3a, del proceso de pasar por caja de la tienda de ejemplo, como se describe en el Ejemplo de uso de Pasar por caja el carro de la compra, el cliente puede comprobar las disponibilidad de los productos y ver la fecha de envío estimada de cada artículo del pedido (`ProductAvailability.jsp`).

Nota: Esta página sólo aparece si alguno de los artículos del pedido no está disponible.

Si actualmente no hay existencias disponibles para algunos de los artículos, los clientes pueden elegir una preferencia de envío, que les permite dividir el pedido o poner los artículos en espera. Los clientes también tienen la opción de eliminar un artículo del pedido.

Mandatos

ProductAvailability.jsp utiliza los mandatos siguientes:

- ProductDisplay
- OrderItemDelete
- OrderPrepare
- OrderItemMove
- OrderItemDisplay

Beans

ProductAvailability.jsp utiliza los beans siguientes:

- OrderDataBean
- OrderItemDataBean
- CatalogEntryDataBean

Implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

La página Pasar por caja 3a, Comprobar disponibilidad de productos (ProductAvailability.jsp) sólo se muestra si hay algunos artículos del pedido que no están disponibles. AllocationCheck.jsp comprueba si hay algún artículo que no está disponible. Si todos los artículos están no disponibles o todos los artículos están disponibles, redirige a OrderDisplayPending.jsp; de lo contrario, incluye ProductAvailability.jsp. ProductAvailability.jsp permite a los clientes ver la fecha de disponibilidad para cada artículo del pedido. Si no hay existencias disponibles para algunos artículos del pedido, el cliente selecciona una opción de envío para el pedido en el campo **Elegir preferencia de envío**.

Los clientes pueden elegir entre tres opciones de envío:

1. Esperar hasta que el pedido entero esté listo para el envío.
2. Enviar los artículos disponibles ahora y enviar el resto cuando estén disponibles.
3. Enviar los artículos disponibles ahora y dejar el resto en el carro de la compra.

Si el cliente elige la opción 2 o 3, el mandato OrderItemMove divide el pedido en dos pedidos separados.

Páginas de visualización de paquete de las tiendas de ejemplo WebFashion y NewFashion

Una página de visualización de paquete contiene grupos promocionales de productos de una tienda en línea. Normalmente, incluye una descripción, una imagen del paquete, una lista de los componentes que constituyen el paquete y una imagen de cada uno, el precio del paquete, una lista de atributos (talla y color) si los productos que componen el paquete tienen variaciones, y una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla).

Los clientes pueden añadir el paquete al carro de la compra o a la lista de deseos con una sola pulsación del ratón. El paquete se visualiza como una sola línea de

detalle en el carro de la compra, a diferencia del paquete compuesto, que se visualiza como una línea de detalle separada para cada artículo del paquete compuesto. Los productos de un paquete no se pueden comprar por separado.

Para obtener más información sobre las páginas de visualización de paquetes de la tienda de ejemplo y cómo funcionan, consulte el Ejemplo de uso de Visualizar página de paquete.

Mandatos

PackageDisplay.jsp utiliza los mandatos siguientes:

- OrderItemAdd
- InterestItemAdd

Beans

PackageDisplay.jsp utiliza los beans siguientes:

- PackageDataBean
- CompositeProductDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

packagedisplay.jsp está registrado en la base de datos (en la tabla DISPENTREL) para mostrar todos los productos de la tienda. packagedisplay.jsp muestra lo siguiente:

- Descripción (incluyendo la lista de componentes que constituyen el paquete), imagen y precio del paquete
- Atributos y valores de atributo de los productos incluidos en el paquete
- Los enlaces Añadir al carro de la compra y Añadir a la lista de deseos
- Cuadro de texto Cantidad (sólo NewFashion)

Descripción (incluyendo la lista de componentes que constituyen el paquete), imagen y precio del paquete

La descripción, imagen y precio del paquete se recuperan utilizando PackageDataBean.

Atributos y valores de atributo de los productos incluidos en el paquete

PackageDataBean recupera ProductDataBeans para mostrar información sobre cada producto del paquete. Para obtener más información, consulte Páginas de productos de la tienda de ejemplo.

Para que el paquete se visualice correctamente, debe utilizar el mismo nombre de parámetro para el valor de atributo para cada producto del paquete. Por ejemplo:

```
<SELECT NAME="attrValue" >
<!-- Visualizar valores de atributo del producto !-->
<%
Object values[] = attribute.getDistinctAttributeValues();
for (int j = 0; j < values.length; j++)
```

```

{
%>
<option><%=values[j]%></option>
<% } %>
</select>

```

Añadir al carro de la compra y Añadir a la lista de deseos

Para obtener más información, consulte Páginas de productos de la tienda de ejemplo.

Cuadro de texto Cantidad

Permite a los clientes especificar el número de paquetes a añadir al carro de la compra o a la lista de deseos.

Marco de navegación izquierdo de la tienda de ejemplo

El marco de navegación izquierdo (`sidebar.jsp`) de las tiendas de ejemplo permite a los clientes seleccionar el idioma en el que se visualizará la tienda. El marco de navegación también incluye enlaces con las páginas de registro y de ayuda.

Para obtener más información, consulte el Ejemplo de uso de Página de presentación.

Mandatos

`sidebar.jsp` utiliza los mandatos siguientes:

- `StoreCatalogDisplay`
- `LogonForm`
- `HelpView`

Beans

`sidebar.jsp` utiliza los beans siguientes:

- `StoreLanguageAccessBean`
- `LanguageDescriptionAccessBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente selecciona un país/región e idioma en la lista desplegable ELIJA UN PAÍS/REGIÓN y pulsa **IR**, las páginas de catálogo de la tienda se vuelven a cargar de acuerdo con el `languageId`, utilizando el código siguiente:

```

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

```

Se crea dinámicamente código JavaScript a fin de encadenar juntos los mandatos `StoreCatalogDisplay` y `SetCurrencyPreference`. Esto se hace para enlazar un idioma

a una moneda por omisión. Cuando el cliente selecciona un idioma, está seleccionando indirectamente la moneda por omisión para ese idioma y, como resultado, no hace falta llamar al mandato OrderPrepare en la página shoppingcart.jsp. Esto se muestra en el código siguiente:

```
<select NAME="currency"> ... <option value="<%=  
(String)currencyId.elementAt(iElementNum) %>" SELECTED>  
<%= (string)currencyid.elementAt(iElementnum+1)%> </option>
```

El siguiente es un ejemplo de javascript generado dinámicamente que enlaza a un idioma, basándose en la moneda seleccionada:

```
<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>
```

Marco de navegación izquierdo de la tienda de ejemplo NewFashion

El marco de navegación izquierdo (sidebar.jsp) de la tienda NewFashion permite a los clientes seleccionar el idioma y la moneda que se visualizarán en la tienda y buscar artículos en el catálogo. El marco de navegación también incluye enlaces con las páginas de Registro, Ayuda, Búsqueda avanzada y Conversación en directo con Asistencia al cliente.

Nota: El enlace **Conversación en directo con Asistencia al cliente** sólo se puede ver si se ha habilitado en la tienda mediante los Servicios de tienda.

Para obtener más información, consulte el Ejemplo de uso de Página de presentación.

Mandatos

sidebar.jsp utiliza los mandatos siguientes:

- StoreCatalogDisplay
- SetCurrencyPreference
- LogonForm
- HelpView
- CatalogSearchResultView
- AdvancedSearchView

Beans

sidebar.jsp utiliza los beans siguientes:

- SupportedLanguageAccessBean
- LanguageDescriptionAccessBean
- CurrencyDescriptionAccessBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando los clientes seleccionan un país/región e idioma de la lista desplegable ELIJA UN IDIOMA y pulsán **IR**, las páginas de catálogo de la tienda se vuelven a cargar de acuerdo con el valor de languageId. El código siguiente obtiene los idiomas soportados por la tienda y los muestra en el recuadro desplegable:

```
<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();
//Obtener el nombre de visualización del idioma en el idioma
//seleccionado actualmente por el comprador.
LanguageDescriptionDataBean langDesc =
new LanguageDescriptionDataBean();
langDesc.setInitKey_languageId(languageId);
langDesc.setInitKey_descriptionLanguageId(storelangId);

//Si este idioma está seleccionado actualmente, seleccionarlo
//en la lista desplegable.
if (languageId.equals(storelangId))
{
%>
<option value="<%= storelangId %>"
SELECTED><%=langDesc.getDescription()%></option>
```

Cuando los clientes seleccionan una moneda de la lista desplegable ELIJA UNA MONEDA, y pulsán **IR**, las páginas de catálogo de la tienda se vuelven a cargar de acuerdo con el valor de supportedCurrencies. El código siguiente obtiene las monedas soportadas y las muestra en el recuadro desplegable:

```
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();
// preseleccionar el valor adecuado en la lista desplegable.
if (currency.equals(supportedCurrencies[i]))
{
%>
<OPTION Value="<%=supportedCurrencies[i]%>"
SELECTED><%=currDesc.getDescription()%></OPTION>
<%
```

Cuando los clientes entran una palabra clave en el campo Búsqueda y pulsán **IR**, el mandato `CatalogSearchResultView` somete los criterios de búsqueda y se visualiza la página `resultlist.jsp` con los resultados de la búsqueda.

Si los clientes pulsán **Conversación en directo con Asistencia al cliente**, se visualiza una ventana emergente en la pantalla, y los clientes pueden conversar en línea en tiempo real con un representante de servicio al cliente. Este enlace sólo se visualizará si esta característica está habilitada en los Servicios de tienda. El enlace **Conversación en directo con Asistencia al cliente** está encerrado dentro del cuerpo de un par de códigos personalizados (el código `<flow:ifEnabled feature="customerCare">` y el código `</flow:ifEnabled>`), y se puede habilitar o inhabilitar en función de la opción que se seleccione en los Servicios de tienda. Siempre y cuando el código personalizado permanezca en su sitio, los Servicios de tienda se pueden utilizar para conmutar automáticamente entre un sitio que tenga soporte de colaboración y otro que no lo tenga, sin tener que modificar la página `JavaServer`. Para habilitar o inhabilitar permanentemente el soporte de colaboración en la página, los códigos personalizados y el enlace **Conversación en directo con Asistencia al cliente** pueden eliminarse de la página `JavaServer` pulsando **Aplicar permanentemente**, en la GUI de los Servicios de tienda. No es aconsejable eliminar ni alterar manualmente los códigos personalizados ni el código adyacente. En lugar de eso, utilice el botón **Aplicar permanentemente** de los Servicios de tienda.

Nota: El código personalizado adyacente al enlace **Conversación en directo con Asistencia al cliente** no se puede copiar en las páginas `JavaServer` de otras tiendas. Estos códigos están diseñados para que funcionen únicamente en la tienda que los contenía originariamente. Para obtener más información acerca de la colaboración, consulte los enlaces relacionados que se muestran a continuación.

Página de selección de dirección de facturación de la tienda de ejemplo

En el primer paso del proceso de pasar por caja de la tienda de ejemplo, la página `Pasar por caja 1. Seleccionar dirección de facturación (billingaddress.jsp)`, el cliente puede seleccionar una dirección existente como la dirección de facturación o crear una nueva dirección para utilizarla como la dirección de facturación.

Para obtener más información, consulte el Ejemplo de uso de `Pasar por caja` el carro de la compra.

Mandatos

`billingaddress.jsp` utiliza los mandatos siguientes:

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

Beans

`billingaddress.jsp` utiliza los beans siguientes:

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

- ErrorDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente inicia el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplaza por una serie de páginas de pasar por caja, la primera de las cuales es la página Pasar por caja 1. Seleccionar dirección de facturación (`billingaddress.jsp`). `billingaddress.jsp` comprueba si el cliente ya tiene direcciones en el listín. Si existen direcciones en el listín, se mostrarán, lo que permitirá al cliente seleccionar una como la dirección de facturación.

El cliente también puede crear una nueva dirección, pulsando **Crear nueva dirección**. Al pulsar **Crear nueva dirección** se llama al mandato `AddressForm`, que está asociado con `AddressForm.jsp` en la base de datos. `AddressForm.jsp` llama a `address.jsp`, que carga la página Añadir dirección. `address.jsp` busca el parámetro `page` para determinar cuál es la página siguiente que debe cargar. Si el valor de `page` se ha establecido en `billingaddress`, el valor de URL del formulario `AddressAdd` se establecerá en `OrderItemDisplay`. `OrderItemDisplay` llama a `billingaddress.jsp`, que devuelve al cliente a la página Pasar por caja 1. Seleccionar dirección de facturación cuando el cliente pulsa **Someter**. Y comprueba el parámetro `page` para determinar qué formulario de dirección ha de cargar.

Si el listín no contiene ninguna dirección, se visualizará el formulario Añadir dirección de facturación, que solicita al cliente que especifique una nueva dirección. El formulario Añadir dirección de facturación de este ejemplo se genera también mediante `billingaddress.jsp`. La acción para este formulario HTML se establece en `AddBillAddressView`, el cual está registrado en la tabla `VIEWREG`. `AddBillAddressView` está asociado con `AddBillAddress.jsp`. Cuando se somete el formulario con `AddBillAddressView`, se llama a `AddBillAddress.jsp`.

`AddBillAddress.jsp` ejecuta los siguientes mandatos:

- `AddressAdd`
- `OrderCopy`

Una vez ejecutado `AddressAdd`, `AddBillAddress.jsp` utiliza el ID de dirección que ha devuelto `AddressAdd` como parte de la entrada del mandato `OrderCopy`. A continuación, `OrderCopy` asigna el ID de dirección a la dirección de facturación del pedido actual y se llama a `OrderItemDisplay.jsp`. El valor del parámetro `page` se establece en `newshipaddress`, por lo tanto, `OrderItemDisplay.jsp` llama a `shipaddress.jsp`.

Nota: Durante el registro, WebSphere Commerce requiere que se cree una dirección. Dado que la tienda de ejemplo no solicita una dirección durante el registro del cliente, algunos de los campos necesarios, como por ejemplo `address1`, se establecen en `unused`. Al comprobar las direcciones, **`billingaddress.jsp`** comprueba si el valor de `address1` es `unused`. Si lo es, la dirección no se visualiza.

Página de selección de dirección de facturación de la tienda de ejemplo NewFashion

En el primer paso del proceso de pasar por caja de la tienda de ejemplo, la página Pasar por caja 1. Seleccionar dirección de facturación (`billingaddress.jsp`), los clientes pueden seleccionar una dirección existente como la dirección de facturación, o crear una nueva dirección para utilizarla como la dirección de facturación.

Para obtener más información, consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Mandatos

`billingaddress.jsp` utiliza los mandatos siguientes:

- `AddressForm`
- `OrderItemDisplay`
- `OrderCopy`
- `AddressAdd`

Beans

`billingaddress.jsp` utiliza los beans siguientes:

- `OrderDataBean`
- `AddressDataBean`
- `OrderItemDataBean`
- `ErrorDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando los clientes inician el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplazan por una serie de páginas de pasar por caja, la primera de las cuales es la página Pasar por caja 1. Añadir dirección de facturación (`billingaddress.jsp`). `billingaddress.jsp` comprueba si los clientes ya tienen direcciones en el listín. Si existen direcciones en el listín, se mostrarán, lo que permitirá a los clientes seleccionar una como la dirección de facturación.

El mandato `OrderItemDisplay` se utiliza para determinar qué página se cargará a continuación. `OrderItemDisplay` devuelve el `OrderItemDisplay.jsp` de la base de datos. `OrderItemDisplay.jsp` incluye archivos JSP diferentes en función del parámetro `page`. Si el valor de `page` es `shipmethod`, se carga la tercera página, Pasar por caja 3. Seleccionar método de envío (`shipping.jsp`).

Los clientes también pueden crear una nueva dirección, pulsando **Crear nueva dirección**. Al pulsar **Crear nueva dirección** se llama al mandato `AddressForm`, que está asociado en la base de datos con `AddressForm.jsp`. `AddressForm.jsp` llama a

address.jsp, que carga la página Añadir dirección. Cuando el cliente pulse **Someter** en la página Añadir dirección, regresará a la página Pasar por caja 1. Añadir dirección de facturación.

Si no hay ninguna dirección existente en el listín, se visualizará el formulario Añadir dirección de facturación, que solicita al cliente que especifique una nueva dirección. El formulario Añadir dirección de facturación de este ejemplo se genera también mediante billingaddress.jsp.

Después de que AddressAdd se ejecuta y llama al mandato OrderCopy, OrderCopy asigna el ID de dirección a la dirección de facturación del pedido actual y llama al mandato OrderItemDisplay, que llama a OrderItemDisplay.jsp. El valor del parámetro page se establece en newshipaddress, por lo tanto, OrderItemDisplay.jsp llama a shipaddress.jsp.

Nota: En las tiendas InFashion y WebFashion, AddBillAddress.jsp se utiliza para llamar a los mandatos AddAddress y OrderCopy. En la tienda NewFashion, se ha eliminado AddBillAddress.jsp encadenando los mandatos AddAddress y OrderCopy mediante el parámetro *URL*. Este método es más fácil de implementar ya que no requiere un JSP adicional, sin embargo tiene una vía de acceso de ejecución en el sistema más larga debido a la redirección adicional.

Carro de la compra de la tienda de ejemplo

Los clientes pueden ver y editar los artículos que han seleccionado en el carro de la compra (shoppingcart.jsp), como se describe en el Ejemplo de uso de Visualizar carro de la compra.

Mandatos

shoppingcart.jsp utiliza los mandatos siguientes:

- OrderItemUpdate
- OrderItemDelete
- StoreCatalogDisplay
- QuickCheckoutView

Beans

shoppingcart.jsp utiliza los beans siguientes:

- OrderDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente pulsa **CARRO DE LA COMPRA** en la cabecera o en el pie de página, se llama al mandato OrderItemDisplay, el cual devuelve el mandato de vista OrderItemDisplayViewShiptoAssoc. El mandato de vista OrderItemDisplayViewShiptoAssoc está registrado con OrderItemDisplay.jsp en la base de datos. OrderItemDisplay.jsp incluye diferentes archivos JSP según el parámetro page. Si no se proporciona un valor para page, se carga la página Carro de la compra, (shoppingcart.jsp).

Nota: OrderItemDisplay.jsp ejecuta OrderPrepare, que vuelve a calcular el total del pedido y convierte la moneda por omisión en la moneda que está utilizando el cliente. Las tiendas con una sola moneda no necesitan ejecutar OrderPrepare.

La página Carro de la compra incluye un botón **Actualizar totales** y un botón **Pasar por caja**. Al pulsar **Actualizar totales**, se actualiza la cantidad de artículos solicitados y luego se vuelve a visualizar la página Carro de la compra. Al pulsar **Pasar por caja**, se actualiza la cantidad de artículos solicitados y, a continuación, se visualiza la página Pasar por caja 1: Seleccionar dirección de facturación.

Los botones **Pasar por caja** y **Actualizar totales** utilizan el mismo formulario HTML. Sin embargo, cuando un cliente pulsa **Pasar por caja** antes de que se someta el formulario, se utiliza JavaScript para añadir un parámetro page adicional con el valor establecido en billingaddress.

El mandato que se utiliza para completar la opción de caja rápida es QuickCheckoutView, que es un mandato de vista registrado en VIEWREG y asociado con QuickCheckout.jsp. QuickCheckout.jsp ejecuta los siguientes mandatos de servidor:

- OrderItemUpdate (actualizar dirección de envío)
- OrderItemUpdate (actualizar modalidad de envío)
- OrderCopy (actualizar dirección de facturación e información sobre el pago)
- OrderPrepare

QuickCheckout.jsp recupera las direcciones de envío y facturación, el método de envío y la información sobre el pago del perfil de caja rápida del cliente utilizando OrderAccessBean. A continuación, asigna esta información al pedido especificado en el parámetro orderId y ejecuta el mandato OrderPrepare.

Una vez completado el proceso de caja rápida, el mandato se redirige a una vista especificada en el URL. WebFashion especifica QuickCheckoutSummaryView como el URL para el mandato QuickCheckout. De este modo, cuando se completa el proceso de caja rápida, se visualiza la página Resumen del pedido de caja rápida.

Manejo de errores

Si la tienda no tiene un centro de despacho de pedidos asociado, o si no hay existencias del producto en el almacén, el mandato OrderItemAdd/OrderItemUpdate llama a la vista de error ResolveFulfillmentCenterErrorView, que está registrada con shoppingcart.jsp en la base de datos. Si los clientes entran un carácter no válido en el campo de cantidad, se llama a la vista de error InvalidInputErrorView. InvalidInputErrorView también está registrada con shoppingcart.jsp en la base de datos.

Puesto que los errores los puede ocasionar tanto el mandato OrderItemUpdate como el mandato OrderItemAdd, shoppingcart.jsp comprueba qué mandato ha originado el problema y muestra un error según el caso. Si el último mandato es OrderItemUpdate, se vuelve a visualizar la página Carro de la compra con un mensaje de error. De lo contrario, se visualiza un página de error separada con un mensaje de error. Esto se muestra en el código siguiente:

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

shoppingcart.jsp se utiliza tanto en condiciones normales como en condiciones de error. El bean `ErrorDataBean` y la comprobación de errores determinan bajo qué condición se muestra `shoppingcart.jsp`. Si existe un error, `shoppingcart.jsp` muestra un mensaje de error adecuado.

Si `QuickCheckout.jsp` (`QuickCheckoutView`) no puede encontrar el perfil de caja rápida, llama a `QuickCheckoutError.jsp` para que muestre un mensaje de error. Como resultado, se visualiza `quickcheckouterrorview.jsp`.

`quickcheckouterrorview.jsp` comprueba si el cliente está registrado. Si el cliente está registrado, el sistema muestra un mensaje para solicitar al cliente que cree un perfil de caja rápida. Si no está registrado, el sistema solicita al cliente que se registre primero y que luego cree un perfil de caja rápida.

Carro de la compra de la tienda de ejemplo NewFashion

Los clientes pueden ver y editar los artículos que han seleccionado en el carro de la compra (`shoppingcart.jsp`), como se describe en el Ejemplo de uso de Visualizar carro de la compra. En el carro de la compra puede haber hasta dos pedidos.

Mandatos

`shoppingcart.jsp` utiliza los mandatos siguientes:

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

Beans

`shoppingcart.jsp` utiliza los beans siguientes:

- `OrderItemDataBean`
- `ErrorDataBean`
- `CatalogEntryDataBean`
- `OrderDataBean`
- `FormattedMonetaryAmountDataBean`
- `OrderAccessBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

El carro de la compra siempre se visualiza como un pedido, no obstante pueden haber varios pedidos en el carro de la compra que se hayan dividido. Esto puede pasar si el cliente ha ido anteriormente al flujo de pasar por caja y ha dividido el pedido. Cuando el cliente pulsa **Pasar por caja**, se llama a `OrderItemMove` para que traslade todos los pedidos a un solo pedido. `OrderItemMove` utiliza el parámetro `deleteIfEmpty` para suprimir todos los pedidos vacíos que queden después del traslado.

Cuando un cliente pulsa **CARRO DE LA COMPRA** en la cabecera o el pie de página, se llama al mandato `OrderItemDisplay`, que está registrado con `OrderItemDisplay.jsp` en la base de datos. `OrderItemDisplay.jsp` incluye diferentes archivos JSP según el parámetro `page`. Si no se proporciona un valor para `page`, se carga la página del carro de la compra, (`shoppingcart.jsp`).

El mandato `OrderItemUpdate` actualiza la cantidad para cada artículo del carro de la compra. `FormattedMonetaryAmountDataBean` da formato al precio total y los importes. `OrderDataBean` obtiene los artículos del carro de la compra para cada pedido. Cada artículo del carro de la compra es un `OrderItemBean`.

La página Carro de la compra incluye un botón **Actualizar totales** y un botón **Pasar por caja**. Al pulsar **Actualizar totales**, se actualiza la cantidad de artículos solicitados y luego se vuelve a visualizar la página Carro de la compra. Al pulsar **Pasar por caja**, se actualiza la cantidad de artículos solicitados y luego se visualiza la página Pasar por caja 1. Seleccionar dirección de facturación.

Nota: Los botones **Pasar por caja** y **Actualizar totales** utilizan el mismo formulario HTML.

Manejo de errores

Si la tienda no tiene un centro de despacho de pedidos asociado, `ErrorDataBean` proporcionará información sobre el error. `ErrorDataBean` también proporciona información si el cliente entra una cantidad no válida como, por ejemplo, un carácter no numérico.

`shoppingcart.jsp` se utiliza tanto en condiciones normales como en condiciones de error. El bean `ErrorDataBean` y la comprobación de errores determinan bajo qué condición se muestra `shoppingcart.jsp`. Si existe un error, `shoppingcart.jsp` muestra un mensaje de error adecuado. Para obtener más información sobre el manejo de errores, consulte los conceptos relacionados siguientes.

Página de método de envío de la tienda de ejemplo

En el tercer paso del proceso de pasar por caja de la tienda de ejemplo, como se describe en el Ejemplo de uso de Pasar por caja el carro de la compra, el cliente debe seleccionar un método de envío (`shipping.jsp`).

Mandatos

`shipping.jsp` utiliza los mandatos siguientes:

- `AddShipModeView`
- `OrderItemDisplay`

Beans

`shipping.jsp` utiliza los beans siguientes:

- `OrderBean`
- `OrderItemAccessBean`
- `ShipModeAccesBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente inicia el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplaza por una serie de páginas de pasar por caja. El mandato `OrderItemDisplay` se utiliza para determinar qué página se cargará a continuación. `OrderItemDisplay` devuelve el `OrderItemDisplay.jsp` de la base de datos. `OrderItemDisplay.jsp` incluye diferentes archivos JSP según el parámetro `page`. Si el valor de `page` es `shipmethod`, se carga la tercera página, **Pasar por caja 3. Seleccionar método de envío** (`shipping.jsp`).

La página **Pasar por caja 3. Seleccionar método de envío** (`shipping.jsp`) incluye un formulario que permite a los clientes seleccionar un método de envío. La acción para el formulario se establece en `AddShipModeVew`, el cual está registrado en la base de datos `VIEWREG` en asociación con `AddShipMode.jsp`. Cuando se somete el formulario con `AddShipModeVew`, se llama a `AddShipMode.jsp`.

`AddShipMode.jsp` ejecuta los mandatos siguientes:

- `OrderItemUpdate`
- `OrderPrepare`.

El mandato `OrderItemUpdate` actualiza el artículo de pedido con el método de envío seleccionado. A continuación, se llama al mandato `OrderPrepare` para el proceso previo del pedido. Posteriormente, se llama a `OrderDisplay` para visualizar la página siguiente del proceso de pasar por caja. Si el parámetro `status` está establecido en `P`, se visualiza a continuación la página **Pasar por caja 4: Resumen del pedido** (`OrderDisplayPending.jsp`).

La página **Pasar por caja 3. Seleccionar método de envío** (`shipping.jsp`) muestra la estructura de costes y el plazo de entrega aproximado para cada método de envío. Esta información está almacenada en la tabla `SHPMODEDSC` en los campos siguientes:

- `DESCRIPTION` almacena la descripción del método de envío.
- `SHPMODEDSC.FIELD1` almacena la descripción de la estructura de costes.
- `SHPMODEDSC.FIELD2` almacena el plazo de entrega aproximado.

Si cambia el coste de envío en la base de datos, no olvide cambiar también la descripción en la tabla `SHPMODEDSC` para que los valores actualizados se muestren en esta página.

Nota: `SHPMODEDSC.FIELD1` y `SHPMODEDSC.FIELD2` se han creado utilizando los campos personalizables de la tabla `SHPMODEDSC`.

Página de selección de método de envío de la tienda de ejemplo NewFashion

En el tercer paso del proceso de pasar por caja de la tienda de ejemplo, como se describe en el Ejemplo de uso de **Pasar por caja** el carro de la compra, el cliente debe seleccionar un método de envío (`shipping.jsp`). El cliente puede seleccionar un método de envío distinto para cada artículo del pedido.

Mandatos

shipping.jsp utiliza los mandatos siguientes:

- OrderItemUpdate
- Product Display
- OrderItemDisplay
- AllocationCheck

Beans

shipping.jsp utiliza los beans siguientes:

- OrderDataBean
- ShippingDataBean
- OrderItemDataBean
- ShippingModeDescriptionDataBean
- CatalogEntryDataBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando los clientes inician el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplazan por una serie de páginas de pasar por caja. El mandato OrderItemDisplay se utiliza para determinar qué página se cargará a continuación. OrderItemDisplay devuelve el OrderItemDisplay.jsp de la base de datos. OrderItemDisplay.jsp incluye diferentes archivos JSP según el parámetro page. Si el valor de page es shipmethod, se carga la tercera página, Pasar por caja 3. Seleccionar método de envío (shipping.jsp).

La página Pasar por caja 3. Seleccionar método de envío (shipping.jsp) permite a los clientes seleccionar un método de envío para cada artículo del pedido. Los clientes seleccionan un método de envío en el campo **Método de envío** para cada artículo.

El mandato OrderItemUpdate actualiza el artículo de pedido con el método de envío seleccionado. Cuando los clientes pulsan **Siguiente**, se llama al mandato OrderItemDisplay para que muestre la siguiente página del proceso de pasar por caja. Si algunos de los artículos no están disponibles, se visualiza a continuación la página Pasar por caja 3a, Disponibilidad de productos (ProductAvailability.jsp). La página Disponibilidad de productos se dirige a AllocationCheck.jsp, que determina la página que se ha de visualizar, dependiendo de si hay o no inventario disponible.

La página Pasar por caja 3. Seleccionar método de envío (shipping.jsp) muestra la estructura de costes para cada método de envío. Esta información está almacenada en la tabla SHPMODEDSC en los campos siguientes:

- DESCRIPTION almacena la descripción del método de envío.
- SHPMODEDSC.FIELD1 almacena la descripción de la estructura de costes.

Si cambia el coste de envío en la base de datos, no olvide cambiar también la descripción en la tabla SHPMODEDSC para que los valores actualizados se muestren en esta página.

Nota: SHPMODEDSC.FIELD1 es un campo personalizable de la tabla SHPMODEDSC.

Página de selección de dirección de envío de la tienda de ejemplo

En el segundo paso del proceso de pasar por caja de la tienda de ejemplo, la página Pasar por caja 2. Seleccionar dirección de envío (`shipaddress.jsp`), el cliente puede seleccionar una dirección existente como la dirección de envío, editar esa dirección o crear una nueva dirección para utilizarla como la dirección de envío.

Para obtener más información, consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Mandatos

`shipaddress.jsp` utiliza los mandatos siguientes:

- `OrderItemUpdate`
- `AddressForm`
- `OrderItemDisplay`

Beans

`shipaddress.jsp` utiliza los beans siguientes:

- `OrderBean`
- `AddressAccessBean`
- `OrderItemDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente inicia el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplaza por una serie de páginas de pasar por caja, la segunda de las cuales es la página Pasar por caja 2. Seleccionar dirección de envío (`shipaddress.jsp`). `shipaddress.jsp` muestra todas las direcciones existentes y permite que el cliente seleccione una como la dirección de envío.

El cliente también puede crear una nueva dirección, pulsando **Crear nueva dirección**. Al pulsar **Crear nueva dirección** se llama al mandato `AddressForm`, que está asociado en la base de datos con `AddressForm.jsp`. `AddressForm.jsp` llama a `address.jsp`, que carga la página Añadir dirección. `address.jsp` busca el parámetro `page` para determinar cuál es la página siguiente que debe cargar. Si el valor de `page` está establecido en `shipaddress`, el valor de URL del formulario `AddressAdd` se establecerá en `OrderItemDisplay`. `OrderItemDisplay` llama a `shipaddress.jsp`, que devuelve al cliente a la página Pasar por caja 2. Seleccionar dirección de envío cuando el cliente pulsa **Someter**.

Nota: Durante el registro, WebSphere Commerce requiere que se cree una dirección. Dado que la tienda de ejemplo no solicita una dirección durante el registro del cliente, algunos de los campos necesarios, como por ejemplo `address1`,

se establecen en "-". Al comprobar las direcciones, `billingaddress.jsp` comprueba si el valor de `address1` es "-". Si lo es, la dirección no se visualiza.

Página de selección de dirección de envío de la tienda de ejemplo NewFashion

En el segundo paso del proceso de pasar por caja de la tienda de ejemplo, la página Pasar por caja 2. Seleccionar dirección de envío (`shipaddress.jsp`), los clientes pueden seleccionar una dirección existente como la dirección de envío, editar esa dirección o crear una nueva dirección para utilizarla como la dirección de envío. La dirección existente se selecciona eligiendo un apodo en el campo **Enviar a**. Cada artículo del pedido puede tener una dirección de envío distinta.

Para obtener más información, consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Mandatos

`shipaddress.jsp` utiliza los mandatos siguientes:

- `OrderItemUpdate`
- `AddressForm`
- `AddressBookForm`
- `Product Display`
- `OrderItemDisplay`

Beans

`shipaddress.jsp` utiliza los beans siguientes:

- `OrderBean`
- `AddressDataBean`
- `OrderItemDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando los clientes inician el proceso de pasar por caja, pulsando **Pasar por caja** en la página del Carro de la compra, se desplazan por una serie de páginas de pasar por caja, la segunda de las cuales es la página Pasar por caja 2. Seleccionar dirección de envío (`shipaddress.jsp`). `shipaddress.jsp` muestra todos los artículos del pedido, y los apodos enlazados con las direcciones existentes, lo que permite al cliente seleccionar un apodo distinto como la dirección de envío para cada artículo.

El mandato `OrderItemDisplay` se utiliza para determinar qué página se cargará a continuación. `OrderItemDisplay` devuelve el `OrderItemDisplay.jsp` de la base de datos. `OrderItemDisplay.jsp` incluye diferentes archivos JSP según el parámetro `page`. Si el valor de `page` es `shipmethod`, se carga la tercera página, Pasar por caja 3. Seleccionar método de envío (`shipping.jsp`).

Los clientes pueden crear nuevas direcciones pulsando **Crear nueva dirección**. Al pulsar **Crear nueva dirección** se llama al mandato `AddressForm`, que está asociado

con AddressForm.jsp en la base de datos. AddressForm.jsp llama a address.jsp, que carga la página Añadir dirección. La página Añadir dirección devolverá a los clientes a la página Seleccionar dirección de envío cuando pulsen **Someter**.

Los clientes también pueden editar una dirección pulsando **Editar listín**. Al pulsar **Editar listín** se llama al mandato AddressBookForm, que está asociado en la base de datos con AddressBookForm.jsp. AddressBookForm.jsp llama a AddressBook.jsp, que carga la página Listín. En esta página se muestran todas las direcciones que están asociadas a ese cliente. A continuación, los clientes pueden pulsar **Editar** bajo la dirección elegida. Al pulsar **Editar** se visualiza la página Actualizar dirección, donde se pueden realizar los cambios. Los clientes regresan a la página Listín cuando pulsan **Someter**. La página Listín devuelve a los clientes a la página Pasar por caja 2. Seleccionar dirección de envío cuando pulsan **Volver a Pasar por caja**.

Nota: Durante el registro, WebSphere Commerce requiere que se cree una dirección. Dado que la tienda de ejemplo no solicita una dirección durante el registro del cliente, los campos de la dirección creada durante el registro están vacíos. Al comprobar las direcciones, shipaddress.jsp comprueba si el valor de LogonID es igual a NickName. Si lo es, significa que la dirección es una dirección automática, es decir, una dirección creada durante el registro, y la dirección no se visualiza. Si lo es, la dirección no se visualiza.

Página de resumen de caja rápida de la tienda de ejemplo

Cuando un cliente pulsa **Caja rápida** desde el Carro de la compra, se visualiza la página Caja rápida: Resumen del pedido (quickcheckoutsummary.jsp), con los detalles del pedido e información de envío, facturación y pago cumplimentada de antemano. El cliente debe entonces completar el pedido pulsando **Realizar pedido ahora**.

Para obtener más información, consulte el Ejemplo de uso de Caja rápida.

Mandatos

quickcheckoutsummary.jsp utiliza los mandatos siguientes:

- OrderProcess
- OrderItemDisplay
- PrivacyView
- ContactView

Beans

quickcheckoutsummary.jsp utiliza los beans siguientes:

- OrderBean
- AddressDataBean
- ErrorDataBean
- ProfileCassetteAccountDataBean
- OrderPaymentInfoAccessBean

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

El comportamiento de la página de resumen de caja rápida es igual que el de la Página de resumen del pedido de la tienda de ejemplo, con las siguientes diferencias:

- Los campos de información sobre el pago están cumplimentados de antemano con información del perfil de caja rápida.
- La información sobre el pago se recupera utilizando `OrderPaymentInfoAccessBean`.

Páginas para ver pedidos de las tiendas de ejemplo WebFashion y NewFashion

Después de que los clientes registrados han formalizado un pedido, pueden ver el estado de ese pedido cuando lo deseen. Para ver un pedido, el cliente pulsa **Mi cuenta** y, desde la página Mi cuenta, pulsa **Ver pedidos**. Se visualiza la página Estado de los pedidos (`trackorderstatus.jsp`) con la lista de pedidos que el cliente ha formalizado. Para obtener más información sobre un pedido concreto, los clientes pulsan en el pedido. Se visualiza la página Resumen del pedido (`orderdetail.jsp`).

Nota: Los pedidos temporales, pendientes, editados por un Representante de servicio al cliente, de perfil de pedido rápido, de lista de solicitudes privada, de lista de solicitudes compartida, sin inventario y cancelados no se pueden ver.

Para obtener más información, consulte el Ejemplo de uso de Ver pedidos.

Beans

`trackorderstatus.jsp` utiliza los beans siguientes:

- `OrderDataBean`
- `PayStatusListPMDDataBean`
- `OrderAccessBean`
- `PriceDataBean`

`orderdetail.jsp` utiliza los beans siguientes:

- `OrderDataBean`
- `OrderItemDataBean`

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Cuando un cliente pulsa **Ver pedidos**, se visualiza la página Estado de los pedidos (`trackorderstatus.jsp`) con una lista de los pedidos que el cliente ha formalizado. `OrderDataBean` recupera todos los pedidos formalizados por el cliente utilizando el código siguiente para desplazarse por todos los pedidos de la lista:

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

`OrderDataBean` muestra la siguiente información sobre el pedido:

- Número de pedido
- Fecha del pedido
- Importe total
- Estado del pago

OrderDataBean recupera el estado del pago llamando al método `.getStatus()`. El método `.getStatus()` devuelve una serie de estado de un carácter que representa el estado del pago tal como se muestra en la tabla siguiente.

Estado	Descripción breve	Significado
P	Pendiente	El cliente puede modificar el pedido.
I	Sometido	El cliente no puede modificar el pedido.
W	Pendiente de aprobación	El mandato de tarea <code>CheckOrderApproval</code> ha indicado que algunos artículos del pedido no están aprobados.
N	Aprobación denegada	El mandato de tarea <code>CheckOrderApproval</code> ha indicado que se ha denegado la aprobación para algunos artículos del pedido.
M	Pendiente de autorización de pago	En espera de la autorización de pago.
A	Autorización de pago requiere revisión	La autorización de pago ha encontrado una circunstancia inusual, como por ejemplo, un aviso de verificación de dirección. La autorización de pago debe revisarse y aceptarse, o cancelarse el pedido, utilizando la interfaz de usuario de Gestión de pedidos. Si la autorización es aceptada, la interfaz de usuario cambiará el estado del pedido a 'B' o 'C' según corresponda.
B	En espera	Es posible que la autorización de pago tenga que hacerse de nuevo, ya que el importe del pedido puede cambiar como resultado de la asignación de inventario de pedidos en espera.
C	Autorización de pago completada	La autorización de pago se ha completado. Todos los artículos de pedido se asignan del inventario existente. El importe del pedido no cambiará.

E	Editado por un CSR	Un representante de servicio al cliente está trabajando con el pedido.
R	Despachado	Todos los artículos de pedido han salido para despacho.
S	Enviado	Todos los artículos del pedido han sido enviados.
D	Depositado	Se ha capturado el pago.
L	Sin inventario	No hay existencias disponibles para uno o más artículos del pedido.
T	Temporal	Utilizado por la interfaz de usuario de Gestión de pedidos para hacer una copia de seguridad temporal de un pedido.
Q	Perfil de pedido rápido	El pedido contiene información de pedido por omisión que puede copiarse para crear rápidamente nuevos pedidos.
F	Listo para despacho remoto	El pedido está listo para enviarlo a un sistema remoto para su despacho. Este estado lo utiliza la característica MQAdapter.
G	Pendiente de despacho remoto	El pedido se ha enviado a un sistema remoto para su despacho. Este estado lo utiliza la característica MQAdapter.
Y	Lista de solicitudes privada	El pedido es una lista de solicitudes privada.
Z	Lista de solicitudes compartida	El pedido es una lista de solicitudes compartida.
X	Cancelado	El pedido se ha cancelado.

Nota: Payment Manager comprueba si el pago se ha rechazado. Si el estado del pago no es rechazado, se visualiza un mensaje que depende de lo que devuelva el método `.getStatus()`.

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

Una vez que los clientes han recuperado la lista de sus pedidos, pueden pulsar en un pedido específico para obtener más información (`orderdetail.jsp`). `OrderDataBean` recupera todos los artículos de pedido para el cliente, y `OrderItemDataBean` recupera los detalles sobre cada artículo. `OrderItemDataBean` recupera los detalles siguientes:

- Precio del pedido

- Contenido del pedido
- Fecha de envío estimada o real del pedido
- Número de seguimiento para cada artículo

Cada artículo puede consistir en varias piezas enviadas en diferentes cajas y puede tener más de un número de seguimiento. Varios artículos pueden compartir los mismos números de seguimiento.

OrderDataBean recupera la fecha de envío estimada o real del pedido, tal como se muestra en el código siguiente:

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

Páginas de lista de deseos de las tiendas de ejemplo WebFashion y NewFashion

Las páginas de lista de deseos de la tienda de ejemplo permiten a los clientes registrados añadir artículos a la lista de deseos (lista de intereses), y ver y editar la lista (`interestitemdisplay.jsp`), añadiendo productos al carro de la compra cuando lo deseen. Los clientes también pueden enviar su lista de deseos a familiares y amigos a través del correo electrónico (`sendwishlistmsg.jsp`). En la tienda NewFashion, los clientes pueden enviar un mensaje personal junto con su lista de deseos.

Nota: En NewFashion se permite añadir artículos a la lista de deseos tanto a los clientes registrados como a los compradores invitados. En WebFashion sólo se permite añadir artículos a la lista de deseos a los clientes registrados.

Cuando los clientes reciben una lista de deseos a través del correo electrónico, ven la página de lista de deseos (`sharedwishlist.jsp`). Esta página es idéntica a `interestitemdisplay.jsp`, sin los botones **Enviar lista de deseos** y **Eliminar artículo**. El cliente, si desea adquirir un artículo, puede seleccionarlo de la lista y añadirlo a su carro de la compra.

Para obtener más información, vea el Ejemplo de uso de Ver lista de deseos y el Ejemplo de uso de Añadir artículo a la lista de deseos.

Los procedimientos de lista de deseos utilizan los siguientes archivos JSP:

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

Mandatos

`interestitemdisplay.jsp` utiliza los mandatos siguientes:

- `OrderItemAdd`
- `SendWishListMsg`

`sharedwishlist.jsp` utiliza los mandatos siguientes:

- `OrderItemAdd`

`sendwishlistmsg.jsp` utiliza los mandatos siguientes:

- SendMsgCmd

Beans

interestitemdisplay.jsp utiliza los beans siguientes:

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean
- UserRegistrationDataBean

sharedwishlist.jsp utiliza los beans siguientes:

- InterestItemListDataBean
- InterestItemDataBean
- CatalogEntryAccessBean

sendwishlistmsg.jsp utiliza los beans siguientes:

- UserRegistrationDataBean (sólo WebFashion)

Detalles sobre implementación

Nota: Para obtener información sobre las técnicas de implementación comunes a todas las páginas de la tienda de ejemplo, incluida la información multicultural, consulte Páginas de la tienda de ejemplo: técnicas de implementación comunes.

Añadir a la lista de deseos

Cuando los clientes pulsán **Añadir a la lista de deseos**, se llama al mandato InterestItemAdd. Para obtener más información, consulte Páginas de productos de la tienda de ejemplo.

Antes de añadir el artículo a la lista de deseos, el sistema utiliza UserRegistrationDataBean para comprobar que el cliente está registrado en la tienda. Si el cliente no está registrado, el JSP incluye registerfirst.jsp, que muestra a continuación un mensaje solicitando al cliente que se registre y que vuelva a intentarlo.

Ver lista de deseos

Cuando los clientes pulsán **Ver lista de deseos**, se visualiza la página Lista de deseos (interestitemdisplay.jsp) con una lista del contenido de la lista de deseos. InterestItemListDataBean e InterestItemDataBean recuperan la información sobre el contenido de la lista de deseos.

Añadir al carro de la compra y Eliminar artículo

En la página Lista de deseos (interestitemdisplay.jsp), los clientes pueden elegir añadir artículos al carro de la compra o suprimir artículos de la lista de deseos. Cuando un cliente pulsa **Añadir los artículos seleccionados al carro de la compra**, se llama al mandato OrderItemAdd. Cuando un cliente pulsa **Eliminar artículo**, se llama al mandato InterestItemDelete.

Enviar lista de deseos

Cuando un cliente rellena los campos **Nombre** y **Dirección de correo electrónico** y pulsa **Enviar lista de deseos**, se llama al mandato SendWishListMsg. SendWishListMsg está registrado en la tabla VIEWREG para mostrar sendwishlistmsg.jsp.

Antes de poder ejecutar el mandato SendWishListMsg, debe crear una

instancia. WebFashion y NewFashion crean una instancia utilizando lo siguiente:

```
SendMsgCmd sendMsgCmd = (SendMsgCmd)
CommandFactory.createCommand(cmdEntry);
```

Por ejemplo:

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

Puede establecer los parámetros del mandato de tarea "sendMsgCmd" para enviar el mensaje inmediatamente:

```
sendMsgCmd.sendImmediate();
```

Nota: WebFashion sólo tiene un campo Dirección de correo electrónico del destinatario. NewFashion tiene un campo Correo electrónico del destinatario, un campo Nombre del remitente y un campo Mensaje personal. El campo Correo electrónico del destinatario y el campo Nombre del remitente son obligatorios.

Manejo de errores

Si el cliente no ha seleccionado ningún artículo en la lista de deseos y pulsa **Añadir los artículos seleccionados al carro de la compra**, se visualiza un mensaje de error. Esta acción la lleva a cabo el código siguiente:

```
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>
```

Ejemplos de uso de la tienda de ejemplo

Los ejemplos de uso describen detalladamente el flujo de cada interacción del usuario en una tienda, por ejemplo, registrarse o pasar por caja. En la ayuda en línea se proporciona un conjunto de ejemplos de uso, que detallan las interacciones para las tiendas de ejemplo InFashion, WebFashion y NewFashion. Estos ejemplos

de uso pueden ayudarle a comprender con más exactitud el flujo de las tiendas de ejemplo y se pueden utilizar como guía para crear ejemplos de uso para su propia tienda.

Notas:

1. WebFashion es la versión Professional Edition de la tienda InFashion. Las páginas y características específicas de WebFashion se indican mediante .
2. Es posible que páginas de la lista siguiente que no tienen la indicación  contengan algunas características de Professional Edition.

Se proporcionan los siguientes ejemplos de uso:

- Ejemplo de uso de Página de presentación
- Ejemplo de uso de Registro
- Ejemplo de uso de Conexión
- Ejemplo de uso de Gestionar cuenta personal
- Ejemplo de uso de Cambiar información personal
- Ejemplo de uso de Ver categoría de producto
- Ejemplo de uso de Visualizar página de producto
-  Ejemplo de uso de Visualizar página de paquete
-  Ejemplo de uso de Visualizar página de paquete compuesto
- Ejemplo de uso de Visualizar carro de la compra
- Ejemplo de uso de Pasar por caja el carro de la compra
-  Ejemplo de uso de Caja rápida
- Ejemplo de uso de Editar dirección
- Ejemplo de uso de Añadir nueva dirección
-  Ejemplo de uso de Ver pedidos
-  Ejemplo de uso de Añadir artículo a la lista de deseos
-  Ejemplo de uso de Ver lista de deseos
-  Ejemplo de uso de Crear perfil de caja rápida

Para obtener más información sobre cómo funcionan estas páginas conjuntamente, consulte el diagrama de flujo de compra de la tienda de ejemplo.

Capítulo 5. Ejemplos de uso de la tienda de ejemplo

Ejemplo de uso de Añadir nueva dirección (Business Edition)

Business

Los clientes pueden añadir una nueva dirección a su listín.

Actor

Cliente

Flujo principal

El cliente pulsa **Añadir una nueva dirección**. El sistema muestra una página con los campos siguientes:

- Apodo (sólo NewFashion)
- Nombre
- Apellido
- Dirección (compuesta de dos cuadros de texto)
- Ciudad
- Estado o provincia
- Código postal
- País/Región
- Número de teléfono

El cliente escribe la información en los campos y pulsa **Someter**. El sistema añade una nueva dirección al listín (E1).

Flujo alternativo

Ninguno

Flujos de excepción

E1: Falta un campo obligatorio

Si falta cualquiera de los campos siguientes, el sistema emite un mensaje de error.

- Nombre
- Apellido
- Dirección
- Ciudad
- Estado o provincia
- Código postal
- País/Región

Si el apodo especificado ya existe en el listín del cliente, el sistema emitirá un mensaje de error.

El ejemplo de uso se reanuda desde el principio.

Ejemplo de uso de Añadir artículo a la lista de deseos de WebFashion y NewFashion

Los clientes pueden ver y editar los productos que han seleccionado para comprar mediante el carro de la compra.

En la tienda WebFashion, una lista de deseos (o lista de intereses) permite a los clientes registrados añadir productos que les gustaría pedir en el futuro a una lista. En la tienda NewFashion, los clientes no necesitan estar registrados para añadir un artículo a la lista de deseos. Una lista de deseos puede enviarse por correo electrónico a familiares y amigos, que pueden adquirir los artículos como regalos para el cliente. Una lista de deseos se diferencia de un carro de la compra en que los clientes tienen la intención de adquirir los productos del carro de la compra durante la sesión de compras actual.

Actor

Cliente registrado

Flujo principal

El cliente ve un producto, paquete o paquete compuesto (para más información, vea Ejemplo de uso de Visualizar página de producto, Ejemplo de uso de Visualizar página de paquete compuesto y Ejemplo de uso de Visualizar página de paquete) y, a continuación, pulsa **Añadir a la lista de deseos**. El sistema añade el producto, paquete o paquete compuesto a la lista de deseos (E1) y luego muestra la página de lista de deseos, como se describe en Ejemplo de uso de Ver lista de deseos.

Flujos de excepción

E1: Un comprador invitado intenta añadir un artículo a la lista de deseos

Si un comprador invitado intenta añadir un artículo a la lista de deseos, el sistema muestra el siguiente mensaje: Para guardar artículos en su lista de deseos, regístrese en WebFashion y luego vuelva a intentarlo.

En la tienda NewFashion, los compradores invitados pueden añadir artículos a la lista de deseos sin tener que registrarse.

Ejemplo de uso de Ver categoría de producto

Las páginas de grupo de catálogo muestran una lista de subcategorías y productos. Las páginas de grupo de catálogo permiten a los clientes desplazarse por los productos. Las páginas de grupo de catálogo iniciales conducen a temas más amplios y las páginas de grupo de catálogo posteriores reducen la búsqueda. Generalmente, hay tres tipos de páginas de grupo de catálogo:

- Páginas que muestran subcategorías de una categoría padre
- Páginas que muestran productos dentro de una subcategoría
- Páginas que muestran tanto las subcategorías como los productos

Actor

Cliente

Flujo principal

Las categorías de productos de nivel superior se listan en la página Inicio, por ejemplo, Caballeros, Señoras y Novedades. Cuando el cliente pulsa Moda para caballero o Moda para señora, el sistema recupera las subcategorías de la categoría seleccionada de la base de datos, y muestra la información de categoría correspondiente. Cuando el cliente pulsa el enlace Novedades, se visualizan productos nuevos. Se muestra la información siguiente para cada categoría y producto:

- Una imagen reducida (sólo productos)
- El nombre de cada categoría o producto
- Una breve descripción para identificar el producto
- El precio del producto.

Las páginas de categorías de NewFashion y WebFashion también incluyen una oferta estelar, en forma de paquete o paquete compuesto. La oferta estelar incluye una imagen y una breve descripción del paquete o paquete compuesto. La oferta estelar de InFashion es sólo un producto individual.

El cliente pulsa en la imagen reducida o en el nombre. A continuación, el sistema muestra la página correspondiente (A1, A2, A3, A4).

Flujo alternativo

A1: El cliente selecciona una categoría

Cuando el cliente selecciona una categoría, el sistema muestra otra página de categoría reanudando este ejemplo de uso desde el principio.

A2: El cliente selecciona un producto

Cuando el cliente selecciona un producto, se muestra la página del producto como se describe en Ejemplo de uso de Visualizar página de producto.

A3: El cliente selecciona un paquete compuesto (sólo WebFashion y NewFashion)

Cuando el cliente selecciona un paquete compuesto, se muestra la página del paquete compuesto como se describe en Ejemplo de uso de Visualizar página de paquete compuesto.

A4: El cliente selecciona un paquete (sólo WebFashion y NewFashion)

Cuando el cliente selecciona un paquete, se muestra la página del paquete como se describe en Ejemplo de uso de Visualizar página de paquete.

Flujos de excepción

Ninguno

Ejemplo de uso de Pasar por caja el carro de la compra

Los clientes pasan por caja y someten un pedido cuando pagan los productos que aparecen listados en su carro de la compra.

Actor

Cliente

Flujo principal

El cliente pulsa **Pasar por caja** para iniciar el ejemplo de uso.

El sistema muestra la página Dirección de facturación. Si el cliente tiene como mínimo una dirección en el listín, el sistema solicita al cliente que seleccione una de las direcciones del listín como la dirección de facturación. El cliente puede seleccionar una dirección o pulsar **Crear nueva dirección**. Si el cliente pulsa **Crear nueva dirección**, se añade una nueva dirección utilizando el Ejemplo de uso de Añadir nueva dirección. A continuación, el cliente selecciona una dirección. El sistema establece la dirección seleccionada como la dirección de facturación del pedido. Si el cliente no tiene ninguna dirección en el listín, se lleva a cabo la acción A1: Añadir dirección de facturación.

El sistema muestra la página Dirección de envío. Se muestra una lista de direcciones del listín. Si el cliente encuentra la dirección de envío correcta en la lista, selecciona la dirección. De lo contrario, el cliente pulsa **Crear nueva dirección** y añade una nueva dirección utilizando el Ejemplo de uso de Añadir nueva dirección. El sistema establece la dirección seleccionada como la dirección de envío del pedido.

El sistema muestra una lista de los métodos de envío adecuados para la dirección de envío. Se visualiza la información siguiente para cada método de envío:

- Descripción breve
- Coste del envío:
 - Coste de envío fijo por pedido
 - Coste de envío por artículo solicitado
- Plazo de entrega aproximado (sólo InFashion y WebFashion)

El cliente selecciona el método de envío. El sistema establece el método seleccionado como el método de envío del pedido.

Nota: En la tienda NewFashion, la dirección de envío y el método de envío se especifican para cada artículo del pedido.

Sólo para NewFashion

En la tienda de ejemplo NewFashion, el sistema muestra la disponibilidad de cada artículo del pedido, además de varias opciones para enviar los artículos en función de su disponibilidad. Para cada producto del pedido, el sistema muestra la información siguiente:

- Cantidad
- Descripción breve del producto
- Valores de atributo (por ejemplo, talla: x = grande; color: azul)
- Fecha de disponibilidad prevista basada en inventario futuro
- Botón Eliminar (para eliminar el artículo del pedido)

Si la cantidad solicitada para un artículo específico no está disponible en el almacén, el sistema divide el pedido en dos pedidos, uno que contiene los artículos de pedido disponibles y otro que contiene las partes no disponibles del inventario. El cliente elige una de tres preferencias de envío:

- Opción 1, esperar hasta que el pedido entero esté listo antes de enviarlo. El sistema muestra la fecha de disponibilidad prevista del pedido entero.
- Opción 2, enviar los artículos disponibles ahora y enviar el resto más adelante.

- Opción 3, enviar los artículos disponibles ahora y dejar el resto en el carro de la compra para adquirirlos en una fecha posterior.

A continuación, el cliente selecciona la preferencia de envío para el pedido.

Nota: Si todos los artículos están disponibles, el cliente no verá la fecha de disponibilidad para cada artículo.

Para obtener más información sobre el subsistema de inventario de NewFashion, consulte Subsistema de inventario.

Para InFashion, WebFashion y NewFashion

Además de mostrar la información sobre productos, el sistema también muestra los detalles de la información del pedido. Para cada artículo del pedido, el sistema muestra la información siguiente:

- Descripción breve del producto
- Valores de atributo (por ejemplo, talla: x = grande; color: azul)
- Cantidad
- Precio unitario
- Precio total
- El sistema también muestra:
 - Dirección de facturación
 - Dirección de envío
 - Subtotal (el coste total de todos los productos solicitados)
 - Descuentos, si los hay
 - Total de impuestos, si los hay (los diferentes impuestos deben mostrarse por separado)
 - Derechos de aduana, si los hay
 - Coste de envío, incluido el método de envío
 - Total general (lo que se cobrará al cliente)
- Fecha de envío estimada (visualizada en la parte superior) (sólo NewFashion)

Nota: En la tienda NewFashion, la dirección de envío y el método de envío se muestran para cada artículo del pedido.

El sistema solicita al usuario que escriba la información de la tarjeta de crédito. Se solicita la información siguiente:

- Tipo de tarjeta de crédito (por ejemplo, Visa(R) o MasterCard(R))
- Número de tarjeta
- Mes de caducidad
- Año de caducidad

El cliente escribe la información de la tarjeta de crédito y pulsa **Realizar pedido ahora**. El sistema almacena la información sobre el pago (E1) y muestra una página de confirmación con la siguiente información sobre el pedido:

- Número de pedido
- Subtotal (coste total de todos los productos solicitados)
- Total de impuestos
- Envío

- Descuentos, si los hay
- Total general

El cliente imprime la página para futuras consultas.

Flujo alternativo

A1: Añadir dirección de facturación

Se le solicita al cliente que añada una nueva dirección. El cliente entra una dirección utilizando el Ejemplo de uso de Añadir nueva dirección. El sistema establece la nueva dirección como la dirección de facturación del pedido. El ejemplo de uso continúa.

Flujos de excepción

E1: Número de tarjeta de crédito no válido

El sistema comprueba la validez del número de tarjeta de crédito. Si la comprobación no tiene un resultado satisfactorio, el sistema muestra un mensaje de error que indica:

Número de tarjeta de crédito no válido

y solicita al cliente que vuelva a escribir la información. El ejemplo de uso continúa.

E2: Tarjeta de crédito caducada

Si la fecha de caducidad de la tarjeta de crédito es anterior a la fecha actual, el sistema muestra un mensaje de error. El ejemplo de uso se reanuda desde el principio.

Nota: Si un artículo no está disponible, la página Disponibilidad de productos se visualizará para el cliente.

Ejemplo de uso de Visualizar página de paquete compuesto de WebFashion y NewFashion

La página de paquete compuesto muestra más detalles sobre los productos o artículos que componen un paquete compuesto.

Actor

Cliente

Flujo principal

El cliente selecciona un paquete compuesto en la página de categorías. A continuación, el sistema recupera de la base de datos la información del paquete compuesto y muestra una página con información detallada sobre el paquete compuesto seleccionado. La página de paquete compuesto muestra la información siguiente:

- Descripción breve
- Descripción completa
- Imagen de tamaño normal del paquete compuesto
- Imágenes de los productos que componen el paquete compuesto
- Precio de cada producto del paquete compuesto

- Una lista de atributos (talla y color) si los productos que componen el paquete compuesto tienen variaciones, y una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla).

El cliente selecciona a continuación los valores apropiados para cada atributo y pulsa **Añadir al carro de la compra**. En la tienda NewFashion, el cliente puede especificar la cantidad a añadir al carro de la compra. El sistema añade el paquete compuesto al carro de la compra (E-1), y muestra la página Carro de la compra como se describe en el Ejemplo de uso de Visualizar carro de la compra.

Nota: La cantidad por omisión es 1. Los clientes pueden modificar la cantidad después de que el paquete compuesto se haya añadido al carro de la compra.

El cliente también tiene la opción de pulsar **Añadir a la lista de deseos**. Si lo hace, el sistema añade el paquete compuesto a la lista de deseos y muestra la página de lista de deseos, como se describe en el Ejemplo de uso de Añadir artículo a la lista de deseos.

Flujo alternativo

Ninguno

Flujos de excepción

Ninguno

Ejemplo de uso de Visualizar página de paquete de WebFashion y NewFashion

La página de paquete muestra más detalles sobre los productos o artículos que componen el paquete.

Actor

Cliente

Flujo principal

El cliente selecciona un paquete en la página de categorías. El sistema recupera de la base de datos la información del paquete y muestra una página con información detallada sobre el paquete seleccionado. La página de paquete muestra la información siguiente:

- Descripción breve
- Descripción completa
- Imágenes de los productos que componen el paquete
- Precio del paquete
- Una lista de atributos (talla y color) si los productos que componen el paquete tienen variaciones, y una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla)

El cliente selecciona a continuación los valores apropiados para cada atributo y pulsa **Añadir al carro de la compra**. El sistema añade el paquete al carro de la compra (E-1) y muestra la página Carro de la compra, como se describe en el

Ejemplo de uso de Visualizar carro de la compra. En la tienda NewFashion, el cliente puede especificar la cantidad a añadir al carro de la compra. La cantidad por omisión es uno.

Nota: Puesto que los artículos de un paquete no se pueden vender por separado, un paquete cuenta como un artículo en el carro de la compra.

El cliente también tiene la opción de pulsar **Añadir a la lista de deseos**. Si lo hace, el sistema añade el paquete a la lista de deseos y muestra la página de lista de deseos, como se describe en el Ejemplo de uso de Añadir artículo a la lista de deseos.

Flujo alternativo

Ninguno

Flujos de excepción

E1: No se ha seleccionado ningún atributo

Para productos que tienen diferentes atributos, el cliente debe seleccionar un valor para cada atributo. Si el cliente no selecciona un valor, el sistema selecciona un valor por omisión y añade el paquete al carro de la compra o a la lista de deseos.

Ejemplo de uso de Editar una dirección (Business Edition)

► Business

Los clientes pueden editar direcciones en su listín.

Actor

Cliente

Flujo principal

El cliente selecciona una dirección del listín para editarla. El sistema recupera y muestra los detalles de la dirección seleccionada:

- Apodo (sólo NewFashion)
- Nombre
- Apellido
- Dirección
- Ciudad
- Estado o provincia
- Código postal
- País o región
- Número de teléfono

En NewFashion, el apodo que ha entrado el cliente aparece en la parte superior de la página.

El cliente efectúa los cambios deseados y pulsa **Someter**. El sistema actualiza la dirección (E1).

Flujo alternativo

Ninguno

Flujos de excepción

E1: El sistema notifica al cliente que falta información obligatoria y solicita dicha información, que puede incluir lo siguiente:

- Nombre
- Apellido
- Dirección
- Ciudad
- Estado o provincia
- Código postal
- País o región

Si el apodo ya existe en el listín, se visualizará un mensaje de error.

Los clientes entran la información obligatoria que falta.

Ejemplo de uso de Página de presentación

La página de presentación actúa como el escaparate y atrae clientes a la tienda.

Actor

Cliente

Flujo principal

Un cliente entra el URL de la tienda en el navegador Web. El sistema muestra la página de presentación.

La página de presentación incluye lo siguiente:

- Una barra de navegación con enlaces con las páginas siguientes:
 - Página de inicio
 - Página de información de contacto
 - Página de seguridad y privacidad
 - Página del carro de la compra
 - Página de registro
 - Página de cuenta
 - Página de ayuda
- Enlaces con las categorías primarias o de nivel superior. Para cada categoría se visualiza la información siguiente:
 - Una imagen
 - Enlaces con productos dirigidos a un público concreto
Si durante el registro los clientes han especificado a qué sexo pertenecen, se visualizan productos dirigidos al sexo al que pertenecen los clientes. Si no, se muestra un conjunto de productos genéricos.
Para cada producto se visualiza la información siguiente:
 - Una imagen

- Una breve descripción

El cliente pulsa en la imagen. A continuación, el sistema muestra la página correspondiente (A1, A2).

Flujo alternativo

A1: El cliente selecciona una categoría

Cuando el cliente selecciona una categoría, el sistema muestra una página de categoría como se describe en el Ejemplo de uso de Ver categoría de producto.

A2: El cliente selecciona un producto

Cuando el cliente selecciona un producto, se muestra la página del producto como se describe en Ejemplo de uso de Visualizar página de producto.

Ejemplo de uso de Conexión

El proceso de conexión permite que los compradores registrados puedan acceder a sus cuentas escribiendo su nombre de usuario y su contraseña.

Actor

Cliente

Flujo principal

El cliente selecciona **Mi cuenta**. A continuación, el sistema muestra una página con los campos siguientes:

- Dirección de correo electrónico
- Contraseña

Los clientes entran la información adecuada en los campos anteriores y seleccionan **Conexión**. El sistema comprueba que las direcciones de correo electrónico y las contraseñas de los clientes sean correctas y, a continuación, permite a los clientes acceder a sus cuentas. Si los clientes olvidan la contraseña, seleccionan **¿Ha olvidado su contraseña?** y se lleva a cabo el flujo alternativo A1.

Flujo alternativo

A1: Ha olvidado la contraseña

Si los clientes olvidan la contraseña, seleccionan **¿Ha olvidado su contraseña?**. El sistema muestra seguidamente una página que solicita los clientes que escriban su dirección de correo electrónico. Los clientes escriben la dirección de correo electrónico y pulsan **Envíenme mi contraseña**. El sistema enviará la contraseña a la dirección de correo electrónico (E1).

Flujos de excepción

E1: No hay ninguna dirección de correo electrónico coincidente en el sistema

Si el sistema no puede localizar un cliente con la dirección de correo electrónico coincidente, se visualiza el siguiente mensaje de error: No se ha podido localizar un cliente con ese correo electrónico. El ejemplo de uso se cancela.

Si el cliente realiza seis intentos de conexión fallidos con el mismo nombre de usuario, el cliente quedará bloqueado. Se visualizará un mensaje para advertir que la cuenta está bloqueada y que el cliente debe ponerse en contacto con un representante de la tienda para volver a activarla.

Ejemplo de uso de Gestionar cuenta personal

Los clientes gestionan sus cuentas a través de las páginas de cuenta.

Actor

Cliente

Flujo principal

El cliente selecciona **Mi cuenta**. A continuación, el sistema muestra la página Mi cuenta con las opciones siguientes:

- Modificar información personal
- Editar mi listín
- Crear o actualizar perfil de caja rápida (sólo WebFashion)
- Ver lista de deseos
- Ver pedidos

Si el cliente selecciona **Modificar información personal**, se ejecuta el flujo alternativo A1: Modificar dirección de correo electrónico y contraseña.

Si el cliente selecciona **Editar mi listín**, se ejecuta el flujo alternativo A2: Editar listín.

Si el cliente selecciona **Crear o actualizar perfil** (el perfil de Caja rápida), se visualiza la página Perfil de caja rápida, como se describe en el Ejemplo de uso de Crear perfil de caja rápida.

Si el cliente selecciona **Ver lista de deseos**, se visualiza la página Lista de deseos, como se describe en el Ejemplo de uso de Ver lista de deseos.

Si el cliente selecciona **Ver pedidos**, se visualiza la página Estado de los pedidos, como se describe en el Ejemplo de uso de Ver pedidos.

Flujo alternativo

A1: Editar dirección de correo electrónico y contraseña

El sistema solicita al cliente que modifique la dirección de correo electrónico y la contraseña utilizando el procedimiento que se describe en el Ejemplo de uso de Modificar información personal. El ejemplo de uso se reanuda desde el principio.

A2: Editar listín

El sistema muestra una página que lista todas las direcciones que ya se han añadido al listín. Junto a cada dirección hay dos botones: **Editar** y **Suprimir**. Debajo de la lista de direcciones hay un botón **Añadir una nueva dirección**.

Si el cliente pulsa **Suprimir**, el sistema suprime la dirección correspondiente de la base de datos y, a continuación, muestra un mensaje indicando que la dirección se ha suprimido correctamente. El flujo alternativo se reanuda desde el principio.

Si el cliente pulsa **Editar**, el sistema solicita al cliente que edite la dirección utilizando el procedimiento que se describe en el ejemplo de uso de Editar dirección. A continuación, el sistema muestra un mensaje para confirmar que la dirección se ha actualizado correctamente, y el flujo alternativo se reanuda desde el principio.

Si el cliente pulsa **Añadir una nueva dirección**, el sistema solicita al cliente que escriba una nueva dirección utilizando el procedimiento que se describe en el Ejemplo de uso de Añadir nueva dirección. Este flujo alternativo se reanuda desde el principio.

Flujos de excepción

Ninguno.

Ejemplo de uso de Ver pedidos de WebFashion y NewFashion

Los clientes pueden hacer un seguimiento de sus pedidos y ver el estado de los mismos.

Actor

Cliente

Flujo principal

El cliente pulsa **Mi cuenta** y luego **Ver pedidos**. El sistema muestra la página Estado de los pedidos, que lista todos los pedidos formalizados por el cliente. Si el cliente no ha formalizado ningún pedido, se lleva a cabo la acción E1: No se ha encontrado ningún pedido.

Se muestra la información siguiente para cada pedido:

- Número de pedido (enlazado con la página Detalles del pedido)
- Fecha del pedido
- Estado del pedido
- Importe total

Para obtener más información sobre el pedido, el cliente puede pulsar en el número de pedido. El sistema muestra la página Detalles del pedido. La página Detalles del pedido muestra la siguiente información para cada producto del pedido:

- Cantidad
- Descripción breve del producto
- Los atributos (talla y color) y los valores para cada atributo (rojo y azul para el color, y L y XL para la talla).
- Precio unitario
- Precio total
- Fecha de envío para el pedido entero (sólo NewFashion)
- Números de ID de seguimiento (sólo NewFashion)

Flujos de excepción

E1: No se ha encontrado ningún pedido

Si el sistema no puede encontrar ningún pedido asociado a este cliente, el sistema muestra el mensaje de error:

No se ha encontrado ningún pedido

El cliente puede entonces pulsar **Volver a la página Inicio** o **Mi cuenta**.

Ejemplo de uso de Visualizar página de producto

La página de producto muestra más detalles sobre un producto.

Actor

Cliente

Flujo principal

El cliente selecciona un producto de la página de categorías. A continuación, el sistema recupera la información del producto de la base de datos y muestra una página con información detallada sobre el producto seleccionado. La página Producto muestra la información siguiente:

- Nombre del producto
- Descripción detallada
- Precio
- Imagen de tamaño normal del producto
- Una lista de atributos (talla y color) si el producto tiene variaciones, y una lista de valores para cada atributo (rojo y azul para el color y L y XL para la talla)

El cliente selecciona a continuación los valores apropiados para cada atributo y pulsa **Añadir al carro de la compra**. El sistema añade el artículo seleccionado al carro de la compra (E1) y muestra la página del carro de la compra, como se describe en el Ejemplo de uso de Visualizar carro de la compra. En la tienda NewFashion, el cliente puede especificar la cantidad a añadir al carro de la compra. La cantidad por omisión es 1.

En WebFashion y NewFashion, el cliente tiene la opción de pulsar **Añadir a la lista de deseos**. Si lo hace, el sistema añade el paquete a la lista de deseos y muestra la página de lista de deseos, como se describe en el Ejemplo de uso de Añadir artículo a la lista de deseos.

Flujo alternativo

Ninguno

Flujos de excepción

Ninguno

Ejemplo de uso de Modificar información personal

Los clientes pueden modificar su información personal, como por ejemplo, su dirección de correo electrónico y su contraseña.

Actor

Cliente

Flujo principal

El cliente pulsa **Modificar información personal** (E1). El sistema muestra la página Modificar información personal, que incluye los campos siguientes:

- **Nombre**, cumplimentado de antemano con el nombre actual
- **Apellido**, cumplimentado de antemano con el apellido actual
- **Dirección de correo electrónico**, cumplimentado de antemano con la dirección de correo electrónico actual
- **Contraseña**
- **Verificar contraseña**
- **Edad**, cumplimentado de antemano con la edad (sólo WebFashion)
- **Sexo**, cumplimentado de antemano con el sexo (sólo WebFashion)
- **Idioma preferido** (sólo NewFashion)
- **Moneda preferida** (sólo NewFashion)

El cliente altera los campos según sea necesario. El campo de contraseña se puede dejar vacío. Si el cliente opta por dejar vacío el campo de contraseña, el sistema no modifica la contraseña actual. A continuación, el cliente selecciona **Someter**, y el sistema actualiza la dirección de correo electrónico y la contraseña (E2). Si el campo de contraseña está vacío, el sistema deja la contraseña anterior sin modificar.

Flujo alternativo

Ninguno

Flujos de excepción

La contraseña entrada debe tener como mínimo seis caracteres, debe contener al menos un dígito y una letra, y no debe contener un mismo carácter que aparezca un total cinco veces, ni cuatro veces consecutivas, o se visualizará un mensaje de error. También se visualizará un mensaje de error si la contraseña es igual que la dirección de correo electrónico, o si el cliente decide cambiar la contraseña pero especifica una contraseña igual que la anterior. Para más información, consulte las tareas relacionadas listadas más abajo.

E1: El cliente no está conectado

Para cambiar la dirección de correo electrónico y la contraseña, los clientes deben estar conectados. Si los clientes no están conectados, el sistema les solicita que se conecten utilizando el procedimiento que se describe en el Ejemplo de uso de Conexión. A continuación, este ejemplo de uso se reanuda desde el principio.

E2: La dirección de correo electrónico ya existe para otro cliente

La dirección de correo electrónico especificada no puede coincidir con la dirección de correo electrónico de ningún otro cliente existente. Si el

sistema encuentra una coincidencia, solicita al cliente que seleccione una dirección de correo electrónico diferente. El ejemplo de uso se reanuda desde el principio.

E3: No se puede verificar la contraseña

La contraseña especificada debe coincidir con la contraseña que se ha entrado para verificar la contraseña. De lo contrario, el sistema muestra el siguiente mensaje de error: La contraseña no coincide con la contraseña de verificación, inténtelo de nuevo.

El ejemplo de uso se reanuda desde el principio.

Ejemplo de uso de Registro

El proceso de registro permite a los clientes entrar información personal en la base de datos.

Actor

Cliente

Flujo principal

El cliente selecciona **Regístrate**. A continuación, el sistema muestra una página con los campos siguientes:

- Dirección de correo electrónico
- Contraseña
- Verificar contraseña
- Nombre
- Apellido
- Edad (opcional) (sólo WebFashion)
- Sexo (opcional) (sólo WebFashion)
- Idioma preferido (sólo NewFashion)
- Moneda preferida (sólo NewFashion)

El cliente entra la información adecuada en los campos anteriores y selecciona **Someter**. El sistema crea un nuevo cliente y guarda la información del cliente (E1, E2, E3).

El sistema solicita a los clientes que gestionen sus cuentas siguiendo el proceso que se describe en el Ejemplo de uso de Gestionar cuenta personal

Flujo alternativo

Ninguno.

Flujos de excepción

E1: La dirección de correo electrónico ya existe

Si la dirección de correo electrónico ya existe en el sistema, el sistema muestra un mensaje de error solicitando al usuario que escriba otra dirección de correo electrónico. El ejemplo de uso se reanuda desde el principio.

E2: Faltan campos obligatorios

Si no se ha cumplimentado alguno de los campos siguientes (Dirección de

correo electrónico, Contraseña, Verificar contraseña, Nombre, Apellido), el sistema emite un mensaje de error. El ejemplo de uso se reanuda desde el principio.

E3: Contraseña no válida

Si la contraseña no coincide con la contraseña de verificación, el sistema emite un aviso.

Nota: La contraseña entrada debe tener como mínimo seis caracteres, debe contener al menos un dígito y una letra, y no debe contener un mismo carácter que aparezca un total cinco veces, ni cuatro veces consecutivas, o se visualizará un mensaje de error. También se visualizará un mensaje de error si la contraseña es igual que la dirección de correo electrónico.

Ejemplo de uso de Visualizar carro de la compra

Los clientes pueden ver y editar los productos que han seleccionado para comprar mediante el carro de la compra.

Actor

Cliente

Flujo principal

El cliente pulsa **Carro de la compra**. A continuación, el sistema presenta una página que muestra el contenido del carro de la compra. Se visualiza una lista de los productos que hay en el carro de la compra (E1). Para cada uno de estos productos se muestra la información siguiente.

- Descripción breve del producto
- Parejas de nombre de atributo y valor de atributo del artículo, como por ejemplo, talla y grande.
- Cantidad
- Precio unitario
- Precio total

El sistema también muestra el coste total del pedido antes de impuestos y gastos de envío.

Si el cliente pulsa **Actualizar totales**, se lleva a cabo la acción A1: Actualizar totales.

Si el cliente pulsa **Eliminar artículo**, se lleva a cabo la acción A2: Eliminar un artículo del carro de la compra.

Si el cliente pulsa **Pasar por caja**, se lleva a cabo la acción A3: Actualizar carro de la compra y pasar por caja.

Si el cliente pulsa **Reanudar las compras**, el sistema muestra la página desde la cual el cliente inició la página del carro de la compra. El ejemplo de uso finaliza.

Si el cliente pulsa **Caja rápida**, el sistema muestra la página **Caja rápida: Resumen del pedido**, como se describe en el **Ejemplo de uso de Caja rápida**

Flujos secundarios

A1: Actualizar totales

Para actualizar la cantidad de cualquier artículo, el cliente escribe la nueva cantidad en el recuadro de texto y luego pulsa **Actualizar totales**. El sistema actualiza la cantidad de los artículos en el carro de la compra. Si la nueva cantidad de un artículo es 0, el artículo se elimina del carro de la compra.

A2: Eliminar un artículo del carro de la compra

Para eliminar un artículo del carro de la compra, el cliente pulsa **Eliminar artículo** para el artículo. El sistema elimina el artículo del carro de la compra.

A3: Actualizar carro de la compra y pasar por caja

El sistema actualiza la cantidad de los artículos en el carro de la compra e inicia el proceso de pasar por caja. Consulte el Ejemplo de uso de Pasar por caja el carro de la compra.

Flujos de excepción

E1: Carro de la compra vacío

Si no hay ningún artículo en el carro de la compra, el sistema muestra el mensaje siguiente: Su carro está vacío.

Si se entra un número no válido en el campo de cantidad, por ejemplo un carácter, la página de carro de la compra se volverá a visualizar con un mensaje de error.

Ejemplo de uso de Ver lista de deseos de WebFashion y NewFashion

Los clientes pueden ver y editar los productos que han añadido a su lista de deseos (o lista de intereses).

Actor

Cliente

Flujo principal

Los clientes pulsán **Mi cuenta** y luego **Ver lista de deseos**. A continuación, el sistema presenta una página que muestra el contenido de la lista de deseos. Se visualiza una lista de los productos que hay en la lista de deseos (E1). Para cada producto, paquete o paquete compuesto, se muestra la información siguiente:

- Descripción breve del producto
- Atributo y valor del artículo, por ejemplo, valores de talla y color
- Precio unitario
- Enlace **Eliminar**

El sistema muestra también los siguientes elementos en la página:

- Botón **Reanudar las compras**
- Botón **Añadir los artículos seleccionados al carro de la compra**
- Botón **Enviar lista de deseos**
- Campo **Dirección de correo electrónico** (Para: / destinatario)
- Campo **Mensaje de correo electrónico** (sólo NewFashion)
- **De:/Nombre remitente**(sólo NewFashion)
- **Dirección de correo electrónico**(sólo NewFashion)

Para eliminar un artículo de la lista de deseos, el cliente pulsa **Eliminar**. El sistema realiza la acción A1, Eliminar artículo.

Para enviar la lista de deseos a familiares y amigos por correo electrónico, los clientes entran la dirección de correo electrónico de los destinatarios en el campo **Dirección de correo electrónico** y pulsan **Enviar lista de deseos**. El sistema realiza la acción A2, Enviar lista de deseos. En la tienda NewFashion, los clientes pueden enviar un mensaje de correo electrónico utilizando el campo **Mensaje de correo electrónico**. Los clientes deben especificar su nombre y pueden especificar opcionalmente su dirección de correo electrónico. Los clientes registrados tienen el campo **Nombre** cumplimentado de antemano.

Para añadir artículos de la lista de deseos al carro de la compra, los clientes seleccionan cada artículo a añadir al carro de la compra y pulsan **Añadir los artículos seleccionados al carro de la compra**. El sistema realiza la acción A3, Añadir al carro de la compra.

Flujo alternativo

A1: Eliminar artículo

Para eliminar un artículo de la lista de deseos, el cliente pulsa **Eliminar** para el artículo. El sistema elimina el artículo de la lista de deseos. El ejemplo de uso se reanuda desde el principio.

A2: Enviar lista de deseos

Para enviar la lista de deseos a familiares y amigos por correo electrónico, los clientes entran la dirección de correo electrónico de los destinatarios en el campo **Dirección de correo electrónico**. Para enviar la lista de deseos a varios amigos, cada dirección de correo electrónico debe ir separada por una coma, por ejemplo: info@infashion.com, wcs@infashion.com. A continuación, los clientes pulsan **Enviar lista de deseos**.

El sistema compone un mensaje de correo electrónico que contiene la siguiente información:

- instrucciones sobre cómo adquirir los artículos de la lista de deseos
- un enlace con la lista de deseos, que contiene la siguiente información:
 - descripción breve de cada producto, paquete o paquete compuesto de la lista de deseos
 - un enlace con la página de visualización del producto, paquete o paquete compuesto
 - precio
 - botón **Añadir los artículos seleccionados al carro de la compra**

En la tienda NewFashion, los clientes pueden enviar un mensaje de correo electrónico personalizado junto con el mensaje del sistema, entrando el mensaje en el campo **Mensaje de correo electrónico**.

A3: Añadir al carro de la compra

Para añadir un artículo de la lista de deseos al carro de la compra, los clientes seleccionan el artículo a añadir y pulsan **Añadir los artículos seleccionados al**

carro de la compra. El sistema añade todos los artículos seleccionados al carro de la compra y luego muestra el carro de la compra, como se describe en el Ejemplo de uso de Visualizar carro de la compra.

Flujos de excepción

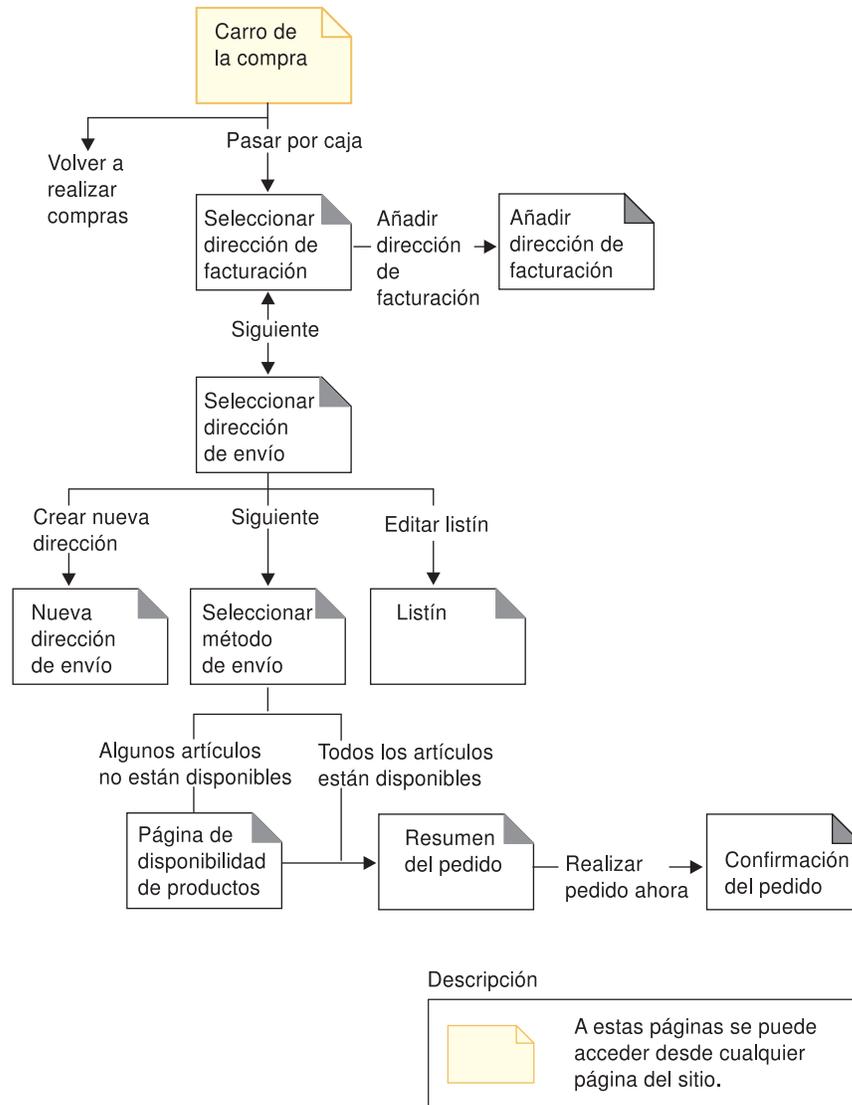
E1: Lista de deseos vacía

Si no hay ningún artículo en la lista de deseos, el sistema muestra el mensaje siguiente: Lista de deseos vacía.
El ejemplo de uso finaliza.

Capítulo 6. Flujo de compra de la tienda NewFashion

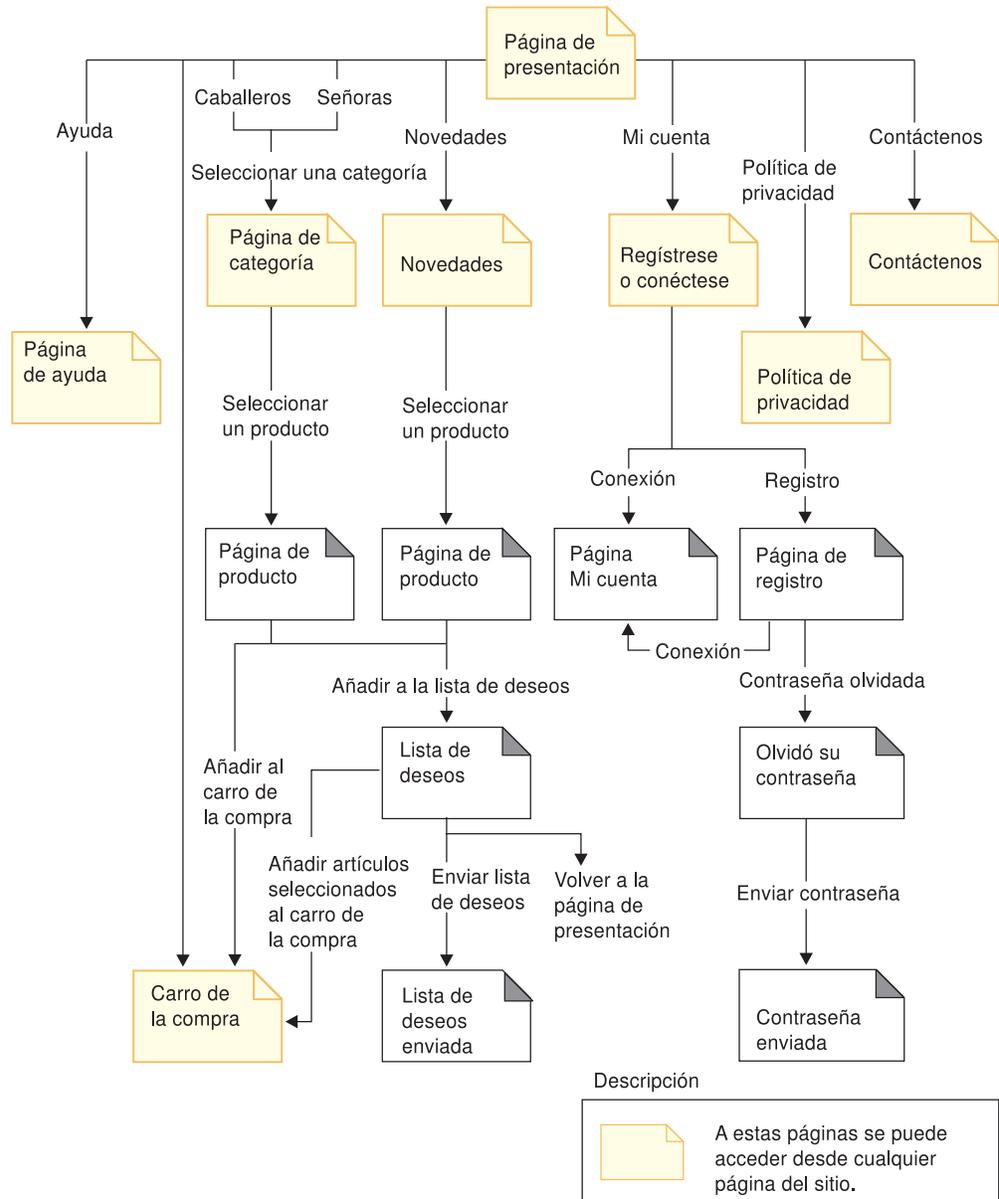
Flujo de carro de la compra de la tienda de ejemplo NewFashion

El diagrama siguiente detalla el flujo de carro de la compra de NewFashion. Para obtener más información sobre el flujo de compra completo, consulte las referencias relacionadas que aparecen más abajo.



Flujo de compra de la tienda de ejemplo NewFashion

El diagrama siguiente detalla las páginas principales del flujo de compra de NewFashion. Para obtener más información sobre los flujos de carro de la compra y de mi cuenta, consulte las referencias relacionadas que aparecen más abajo.



Avisos

Cualquier referencia hecha en este documento a un programa bajo licencia de IBM no pretende afirmar ni implica que únicamente se pueda utilizar dicho programa bajo licencia de IBM. En lugar de dicho producto, programa o servicio de IBM, puede utilizarse cualquier otro producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Es responsabilidad del usuario la evaluación y verificación del funcionamiento junto con otros productos, excepto aquellos designados expresamente por IBM.

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

Director of Licensing
Intellectual Property & Licensing
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

Los propietarios de licencias de este programa que deseen obtener información sobre el mismo con el propósito de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido el aquí citado) y (ii) el uso mutuo de la información que se ha intercambiado, deberán ponerse en contacto con:

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una cantidad.

Este documento puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales cotidianas. Para ilustrar los ejemplos de la forma más completa posible, éstos pueden incluir nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es puramente casual.

Este documento puede contener información acerca de productos de otras empresas, incluidas referencias a los sitios Internet de dichas empresas. IBM no es responsable de la exactitud de esa información, de que esté completa ni del uso que se haga de ella.

Este producto está basado en el protocolo SET.

Nota a los usuarios del gobierno de los EE.UU. — Documentación relacionada con derechos restringidos — El uso, duplicación o divulgación está sujeto a las restricciones indicadas en el GSA ADP Schedule Contract con IBM Corp.

Marcas registradas

Los siguientes términos son marcas comerciales o marcas registradas de International Business Machines Corporation en los Estados Unidos y/o en otros países:

AIX	CICS	DB2
DB2 Extenders	Encina	HotMedia
IBM	iSeries	MQSeries
SecureWay	VisualAge	WebSphere
400		

Blaze Advisor es una marca registrada de HNC Software, Inc. en los Estados Unidos y/o en otros países.

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

Lotus y Domino son marcas registradas de Lotus Development Corporation en los Estados Unidos y/o en otros países.

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

Oracle es una marca registrada de Oracle Corporation.

SET y el logotipo de SET son marcas registradas propiedad de SET Secure Electronic Transaction LLC. Para obtener más información consulte <http://www.setco.org/aboutmark.html>.

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

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

IBM