



# Aplicación del caso práctico de tienda al por menor de iSeries

*Versión 5.1 para Windows*





# Aplicación del caso práctico de tienda al por menor de iSeries

*Versión 5.1 para Windows*



---

# Contenido

<b>Capítulo 1. Introducción a la aplicación del caso práctico de tienda al por menor de iSeries . . . . .</b>	<b>1</b>
Casos prácticos basados en proyectos . . . . .	2

<b>Capítulo 2. Conceptos de tecnología . . . . .</b>	<b>5</b>
--	----------

<b>Capítulo 3. Ejecutar el caso práctico . . . . .</b>	<b>9</b>
--	----------

Antes de empezar . . . . .	9
Instalar los archivos de ejemplo . . . . .	9
Restaurar los archivos .savf . . . . .	10
Modificar la lista de bibliotecas . . . . .	11
Importar y ejecutar la aplicación en Development Studio Client . . . . .	12
Extraer los archivos .zip en el sistema local . . . . .	12
Crear los proyectos Web . . . . .	13
Importar los archivos de los proyectos Web . . . . .	14
Importar los archivos de proyecto WebFacing y crear el proyecto WebFacing . . . . .	14
Configuración de servidor . . . . .	15
Configurar el proyecto WebFacing . . . . .	16
Definir información de servidor . . . . .	17
Ejecutar la aplicación en el entorno de trabajo . . . . .	17
Ejecutar la aplicación en calidad de cliente en el entorno de trabajo . . . . .	17
Ejecutar la aplicación en calidad de administrador en el entorno de trabajo . . . . .	18
Desplegar la aplicación del caso práctico en WebSphere Application Server . . . . .	19
Configurar WebSphere Application Server . . . . .	19
Proteger la página del administrador . . . . .	20
Crear archivos EAR para el despliegue en WebSphere Application Server para iSeries . . . . .	21
Desplegar los archivos EAR en WebSphere Application Server para iSeries . . . . .	22
Ejecutar la aplicación en WebSphere Application Server . . . . .	23

<b>Capítulo 4. Módulo 1 paso a paso: Crear un servicio Web que devuelva los precios de los productos (SV000514) . . . . .</b>	<b>25</b>
---	-----------

Introducción . . . . .	25
Antes de empezar . . . . .	25
Crear un proyecto Web nuevo . . . . .	26
Definir la información del servidor iSeries . . . . .	26
Crear el programa de servicio RPG . . . . .	26
Crear los parámetros y generar el bean Java . . . . .	27
Crear un servicio Web a partir del bean Java . . . . .	28
Probar el ejemplo . . . . .	29

<b>Capítulo 5. Módulo 2 paso a paso: Crear una interfaz para ver el inventario y formular pedidos de artículos (SV000501) . . . . .</b>	<b>31</b>
---	-----------

Introducción . . . . .	31
Antes de empezar . . . . .	32
Crear un proyecto WebFacing . . . . .	32
Convertir el fuente DDS . . . . .	33
Configurar el soporte UTF-8 en el entorno de trabajo — Sólo para los usuarios de WAS Versión 4.0 . . . . .	34
Configurar el soporte UTF-8 para WebSphere Application Server . . . . .	35
Crear una hoja de estilo . . . . .	36
(Opcional) Personalizar manualmente la hoja de estilo en cascada (CSS) . . . . .	36
Ampliar y mejorar el proyecto WebFacing con una interacción Web . . . . .	36
Enlazar el proyecto con una interacción Web . . . . .	39
Publicar los archivos y reiniciar el servidor . . . . .	40
Probar la interfaz . . . . .	41

<b>Capítulo 6. Módulo 1 avanzado: Crear código HTML, servlets y archivos JSP que formulan pedidos de cliente en un servidor iSeries (SV001585) . . . . .</b>	<b>43</b>
--	-----------

Introducción . . . . .	43
Resumen de los pasos de alto nivel . . . . .	43
Antes de empezar . . . . .	44
Crear la página Web, los servlets y los archivos JSP . . . . .	44

<b>Capítulo 7. Módulo 2 avanzado: Crear el proyecto Web que utiliza los servicios Web SV000514 y SV001586 (SV000618) . . . . .</b>	<b>53</b>
--	-----------

Introducción . . . . .	53
Antes de empezar . . . . .	55
Crear las páginas Web, los servlets, los archivos JSP y el código RPG . . . . .	55
Antes de desplegar en WebSphere Application Server . . . . .	57

<b>Capítulo 8. Avisos . . . . .</b>	<b>59</b>
-------------------------------------	-----------

LICENCIA DE COPYRIGHT: . . . . .	60
Información acerca de las interfaces de programación. . . . .	61
Marcas registradas y marcas de servicio . . . . .	61



---

## Capítulo 1. Introducción a la aplicación del caso práctico de tienda al por menor de iSeries

El paquete del caso práctico es una aplicación de ejemplo desarrollada mediante IBM WebSphere Development Studio Client para iSeries, la tecnología basada en Eclipse del servidor iSeries. El caso práctico está destinado a los desarrolladores que deseen utilizar un servidor iSeries para:

- El desarrollo Java
- El desarrollo Web
- La gestión y la transformación de código RPG en aplicaciones Web

La aplicación consta de dos archivos .savf, que se restaurarán en el servidor iSeries, y de cinco archivos .zip que se importarán en el producto cliente. Tienen el formato de una serie de páginas Web con direcciones URL para los diversos puntos del proceso.

**Nota importante:** consulte nuestra página de soporte para obtener todos los paquetes de servicio, los PTF y otra información importante:  
[ibm.com/software/awdtools/wdt400/support/](http://ibm.com/software/awdtools/wdt400/support/).

Este caso práctico le proporcionará indicaciones sobre las diversas partes del producto, haciendo hincapié en componentes específicos del iSeries, como pueden ser:

- La herramienta IBM WebFacing
- Los servicios Web
- Las herramientas de desarrollo Web para iSeries (incluidos el asistente Interacción Web y el asistente Llamada a programa)
- Las herramientas de desarrollo Java para iSeries
- IBM Toolbox para Java

Este caso práctico ilustra una situación en la que dos empresas, un suministrador al por mayor y una tienda de ropa al por menor, desarrollan negocios entre sí y utilizan el servidor iSeries para la lógica comercial y los datos. En el pasado, las dos empresas se comunicaban por correo electrónico, teléfono y fax para comprobar el inventario, enviar pedidos y controlar la entrega de estos. Ahora esperan utilizar la Web para llevar a cabo las transacciones comerciales normales.

La tienda al por menor desea tener un sitio Web que:

- Permita a los clientes comprar productos
- Permita a los empleados pedir artículos del inventario al suministrador al por mayor

El suministrador al por mayor desea:

- Recibir en línea los pedidos del minorista para llevar un control
- Prestar servicio a múltiples clientes potenciales

En este caso práctico, usted desempeñará el papel de consultor de programación para las dos empresas, con objeto de ayudarles a llevar sus negocios a la Web.

La aplicación, en función del tipo de usuario, tiene dos puntos de entrada distintos. Si entra como usuario de tipo cliente, empezará por ver los productos que ofrece la tienda, que en este caso son artículos de ropa. Si desea comprar algo, puede pulsar un enlace para acceder a la pantalla de pedidos. Después de formular su pedido, se genera una página de resumen y podrá seguir visitando la tienda, cancelar el pedido o bien enviarlo.

Si entra como usuario de tipo administrador, tendrá un ID de usuario seguro para la aplicación, lo que exige definir la política de seguridad de la aplicación durante el despliegue. Su punto de entrada es una pantalla de inicio de sesión en la que puede ver sus pedidos, ver el inventario y comprar al suministrador al por mayor. Podrá seleccionar artículos, comprobar el último precio de mayorista y pedir la talla y la cantidad que desee. La aplicación verifica si el mayorista tiene la talla y la cantidad deseada, y o bien confirma el pedido o bien le indica la imposibilidad de satisfacer su pedido en esta ocasión.

Bajo la superficie de la aplicación, se llevan a cabo muchas acciones en las diversas partes del producto. La siguiente tabla ilustra el proceso e indica qué componente del producto es el responsable de cada parte de la aplicación. Continúe en la información detallada sobre cómo realizar cada tarea.

Tabla 1.

Tareas de la aplicación de cliente	Tareas de la aplicación de administrador	Proceso subyacente
Visualizar los precios de los productos.		Utilizar un programa RPG iSeries para crear un servicio Web, y las herramientas de desarrollo Web para visualizar los precios.
Formular un pedido desde la tienda		Utilizar servlets y archivos JSP junto con las herramientas de desarrollo Java para iSeries, las herramientas de desarrollo Web para iSeries e IBM Toolbox para Java para acceder al inventario y verlo en un servidor iSeries, así como para formular pedidos y mostrar el resumen de la compra.
	Ver inventario	Utilizar la herramienta IBM WebFacing para convertir un programa RPG existente en una aplicación Web y utilizar las herramientas de desarrollo Web para personalizar la página Web.
	Formular pedido al suministrador al por mayor con el ID de la mercancía y la cantidad.	Crear un servicio Web que se invoca al pulsar el botón <b>Comprar</b> .
Ver la página Web inicial de la tienda.	Ver la página Web inicial del inventario de pedidos.	Utilizar las herramientas de desarrollo Web para iSeries con el fin de crear las dos páginas iniciales.

## Casos prácticos basados en proyectos

La aplicación del caso práctico de iSeries se compone de cinco proyectos que se llaman SV000501, SV000514, SV000618, SV001585 y SV001586. En esta guía se dan instrucciones para ejecutar la aplicación prefabricada. La guía también contiene módulos paso a paso y módulos avanzados que le enseñarán a construir por sí



mismo los diversos proyectos. Los módulos paso a paso están destinados a los desarrolladores que sean relativamente nuevos en el desarrollo de aplicaciones, así como en Development Studio Client. Los módulos avanzados están destinados a los desarrolladores que tengan experiencia en el desarrollo de aplicaciones y estén más familiarizados con Development Studio Client.

**Nota:** Aunque existen cinco proyectos, solo hay cuatro módulos porque los proyectos SV001586 y SV000514 se obtienen combinando los módulos de servicios Web.

Este caso práctico viene en formato de cinco proyectos:

**Proyecto SV000501: crear un proyecto Web para ver los pedidos pendientes, el inventario y los detalles del producto** – Este proyecto se crea con las herramientas de desarrollo Web para iSeries y la herramienta IBM WebFacing y está destinado a los programadores en RPG que posean un conocimiento limitado del desarrollo de aplicaciones Web y deseen utilizar la herramienta IBM WebFacing para llevar sus aplicaciones RPG a la Web.

**Proyecto SV001585: Crear código HTML, servlets y archivos JSP que formulen los pedidos de los clientes en un servidor iSeries** – Este proyecto utiliza las clases SQL y JDBC de IBM Toolbox para Java, el bean RecordIOManager de herramientas de desarrollo Java para iSeries y el asistente de llamada a programa de iSeries. Estos elementos muestran las diversas formas de acceder y manipular los datos y programas que existen en el servidor iSeries. Este proyecto está destinado a los programadores en Java y a los desarrolladores de aplicaciones Web que deseen desarrollar páginas Web para acceder a los datos y al código de iSeries. Además, hay que tener un conocimiento suficiente de la gestión de servidores iSeries y la programación en RPG.

**Proyecto SV000514: Crear un servicio Web de iSeries para proporcionar los precios de los productos** – El asistente de servicios Web utiliza un bean Java generado por el asistente de llamada a programa de iSeries para llamar a uno o más procedimientos de programa en el servidor iSeries y transmitir la información de nuevo al navegador. Este proyecto está diseñado para los programadores en RPG que deseen utilizar servicios Web para crear aplicaciones modulares y autocontenidas que se puedan describir, publicar, localizar e invocar en la World Wide Web.

**Proyecto SV001586: Crear un servicio Web para formular pedidos al suministrador al por mayor mediante un servidor iSeries** – El servicio Web acepta un ID de mercancía, además de la cantidad requerida, y luego formula el pedido al suministrador al por mayor. Este proyecto es un componente de SV000514 y está destinado a los programadores en RPG que deseen crear servicios Web.

**Proyecto SV000618: crear un proyecto Web que funcione como interfaz del formulario de pedido, del formulario de inventario y del pedido de compra generado por la herramienta IBM WebFacing** – Para este proyecto se necesitan las herramientas de desarrollo Web para iSeries; en él se crean archivos HTML y JSP para utilizar y conectar los servicios Web desarrollados en SV000514 y SV001586. El proyecto está destinado a los desarrolladores que deseen trabajar con los servicios Web y posean conocimientos de programación en RPG y Java.



---

## Capítulo 2. Conceptos de tecnología

Para poder trabajar con la aplicación del caso práctico, debe estar familiarizado con algunos conceptos de tecnología, sobre todo si es usted nuevo en el desarrollo de aplicaciones Web. A continuación figura una pequeña lista de algunos de los elementos con los que se encontrará al trabajar con la aplicación.

### Archivo Enterprise de archivado (EAR)

Un archivo EAR es un archivo Java de archivado (JAR) estándar que tiene la extensión .ear. En él puede haber múltiples proyectos Web y se utiliza para empaquetar y desplegar las aplicaciones Web en el servidor administrativo de WebSphere (WAS). **Nota:** en la versión GUI de la herramienta de despliegue de aplicaciones SDK J2EE, primero se crea un archivo EAR y se añaden archivos JAR y WAR (archivador Web) al archivo EAR. Sin embargo, si utiliza las herramientas de empaquetador de línea de mandatos, primero creará los archivos JAR y WAR y después el archivo EAR.

### Herramienta IBM WebFacing

La herramienta IBM WebFacing convierte las interfaces 5250 existentes en interfaces gráficas de usuario basadas en navegador. Con muy poca o ninguna modificación de las aplicaciones de iSeries originales, podrá llevar el uso de sus programas a Internet o a una intranet.

### Archivo Java de archivado (JAR)

Un archivo JAR es un paquete comprimido de archivos Java, parecido a un archivo .zip. Contiene los archivos de clase, de imagen y de sonido de un applet Java, reunidos en un solo archivo y comprimidos para que resulte más rápido bajarlos al navegador.

### Archivo JSP (JavaServer Pages)

Los archivos JSP proporcionan capacidad para visualizar contenido dinámico en páginas HTML estáticas. Escritos en Java, los archivos JSP son independientes del servidor y de la plataforma. Al separar eficazmente la presentación Web del contenido Web, los archivos JSP sirven de ayuda a los desarrolladores que tienen que cambiar rápidamente el diseño y la visualización de sus páginas Web.

### Bean de llamada a programa

Son beans Java generados por el asistente de llamada a programa. Uno de los tipos es un bean Java normal que se emplea en las aplicaciones Java. El otro tipo se puede utilizar en el asistente de servicios Web para crear un servicio Web.

### Asistente de llamada a programa

El asistente de llamada a programa sirve de ayuda para crear los beans Java y el archivo PCML asociado que se necesita para invocar un procedimiento o un programa de iSeries. Este asistente solicita información acerca de los objetos programa o programa de servicio, junto con los parámetros de los objetos, y luego crea los beans Java deseados (y el archivo PCML).

## **RPG (programa generador de informes)**

Lenguaje de programación orientado a procedimientos que utilizan los programadores del iSeries. El lenguaje RPG le permite crear aplicaciones comerciales como los programas de facturas y los programas de entrada de pedidos. La última versión, ILE RPG IV, amplía las posibilidades del lenguaje RPG, mientras da soporte a la experiencia de los programadores en las versiones anteriores.

## **Servlet**

Programa del lado del servidor, escrito en Java, que se ejecuta en servidores habilitados para Java o en servidores de aplicaciones, como los servidores IBM WebSphere Application Server. Los servlets realizan tareas especificadas por el servidor, como las de responder a las peticiones generando una respuesta HTML. Por ejemplo, podría utilizar servlets en una aplicación bancaria en línea para responder al usuario mientras se envían datos al servidor.

## **Componentes Web**

Los componentes Web permiten definir objetos de iSeries, como por ejemplo campos de entrada de datos y pulsadores, que pueden intercambiar información entre los programas de servidor iSeries y la página Web. Los desarrolladores pueden utilizar los componentes Web para capturar eventos de usuario como la comprobación de sintaxis de los campos de entrada y la pulsación de botones.

## **Asistente Interacción Web**

Este asistente forma parte de las herramientas de desarrollo Web para iSeries. Crea y gestiona las interacciones entre los programas ILE y las páginas Web. El asistente controla dónde se visualizan los datos de entrada, los datos de salida y los mensajes de error, y dirige los datos de los campos de entrada y salida a los programas ILE. El asistente Interacción Web también le permite correlacionar los mensajes de error con el área en la que se produjo el error para que así el usuario pueda identificar fácilmente el origen del error.

## **Servicios Web**

Los servicios Web son aplicaciones autocontenidas diseñadas e implementadas para utilizarse en Internet. Se crean con estándares abiertos como SOAP, WSDL y XML. Se dan numerosas situaciones comerciales en las que es posible utilizar un servicio Web, entre ellas un sistema de gestión de inventario en el que los clientes pueden comprobar sus niveles de inventario por Internet, o cuando interesa controlar los pedidos de un producto directamente desde un suministrador.

## **Lenguaje de definición de servicios Web (WSDL)**

WSDL es un lenguaje basado en XML que define la interfaz de un servicio Web. WSDL interpreta un servicio Web y gestiona el flujo de información entre el servicio Web y el programa del servidor. Por ejemplo, un desarrollador utilizaría WSDL con el fin de crear una interfaz para un sitio Web que mostraría cotizaciones bursátiles actualizadas.

## **WebSphere Studio Workbench**

IBM WebSphere Development Studio Client para iSeries está construido sobre WebSphere Studio Workbench, que es la implementación IBM de la plataforma Eclipse. En el entorno de trabajo universal ampliable se integran todas las herramientas necesarias para construir y mantener las aplicaciones. Los desarrolladores pueden utilizar Development Studio Client para incorporar nuevos objetos en el entorno de desarrollo mediante el uso de conectores y añadir sin fisuras archivos Java, gráficos, vídeo, etcétera.



---

## Capítulo 3. Ejecutar el caso práctico

Puede ejecutar las aplicaciones de *Venta al por mayor* (Wholesale) y *Tienda al por menor* (Retail store) dentro del entorno de trabajo de Development Studio Client o en WebSphere Application Server, sea cual sea la plataforma, incluida la plataforma iSeries. En el tema Capítulo 1, "Introducción a la aplicación del caso práctico de tienda al por menor de iSeries", en la página 1 hallará una visión general de las aplicaciones.

Cuando llegue al final de este capítulo, habrá aprendido a:

- Restaurar los archivos de ejemplo y los objetos de Development Studio Client en el servidor iSeries
- Crear proyectos Web y un proyecto WebFacing destinados a contener los archivos
- Importar en Development Studio Client los archivos restaurados desde el servidor iSeries
- Configurar WebSphere Application Server (WAS)
- Configurar el proyecto WebFacing
- Ejecutar la aplicación en calidad de cliente en el entorno de prueba WebSphere
- Ejecutar la aplicación en calidad de administrador en el entorno de prueba WebSphere
- Configurar WebSphere Application Server
- Proteger la página del administrador
- Crear archivos EAR para el despliegue externo en WAS
- Desplegar la aplicación en WAS
- Ejecutar la aplicación en WAS

---

### Antes de empezar

Para probar las aplicaciones desde el entorno de trabajo, debe asegurarse de que:

- Tiene un servidor iSeries V5R1 o posterior
- Tiene aplicados los PTF más recientes de WebFacing. Consulte nuestra página de soporte para obtener los PTF:  
<http://www.ibm.com/software/awdtools/wdt400/support/>
- Tiene acceso NET USE al Servidor iSeries

---

### Instalar los archivos de ejemplo

Para utilizar la aplicación del caso práctico de iSeries, tendrá que trabajar con los siguientes archivos:

- Wholesale.savf
- Retailstor.savf
- Qdtssfl.savf
- SV000501.zip
- SV000514.zip
- SV000618.zip
- SV001585.zip

- SV001586.zip

Los archivos .savf contienen datos y programas RPG del iSeries, y los archivos .zip contienen las aplicaciones Web que interactúan con los programas del iSeries para manipular los datos del iSeries. Primero tendrá que restaurar los archivos .savf en el servidor iSeries y luego podrá importar los archivos .zip en Development Studio Client y ejecutar la aplicación en el entorno de trabajo.

## Restaurar los archivos .savf

Para trabajar con los ejemplos de esta guía, debe restaurar las bibliotecas WHOLESale y RETAILSTOR en el sistema principal iSeries. Debe hacerlo aunque ya las haya restaurado en un release anterior del producto, porque tienen un contenido diferente.

**Nota:** Los archivos .savf que se utilizan para instalar la biblioteca de los ejemplos son para utilizarse con un sistema principal iSeries de la V5R1 o posterior. Para la finalidad de este caso práctico, las dos bibliotecas se restauran en el mismo sistema principal iSeries, pero si la aplicación se estuviese desarrollando para una empresa real, habría que restaurar las dos bibliotecas en dos sistemas principales iSeries distintos. La biblioteca WHOLESale se tendría que restaurar en el sistema principal iSeries que proporciona los servicios Web y la biblioteca RETAILSTOR se restauraría en el sistema principal iSeries que pertenece a la tienda al por menor.

Para restaurar el archivo Wholesale.savf:

1. Inicie sesión en el sistema principal iSeries mediante un emulador de pantalla verde.
  - a. Cree una biblioteca destinada a contener los archivos de salvar. Para crear una biblioteca nueva en el emulador, escriba CRTLIB.
  - b. Dé a la biblioteca el nombre SCENARIO.
  - c. Con el tabulador, pase a la siguiente línea; especifique \*TEST como tipo de biblioteca y pulse Intro para guardar los cambios.
  - d. Cree dos archivos de salvar emitiendo el mandato CRTSAVE, pulsando Intro entre las dos líneas:

```
CRTSAVF FILE(SCENARIO/WHOLESale)
CRTSAVF FILE(SCENARIO/RETAILSTOR)
```

Estas líneas indican que desea crear los archivos de salvar en la biblioteca SCENARIO.

2. En la estación de trabajo, abra una ventana de indicador de mandatos.
  - a. Debe cambiar al directorio en el que residen los archivos .savf. Por omisión, debe especificar `cd c:\wdsc\wdscsampl`. Si ha instalado el producto en otra unidad o elige no utilizar "wdsc" como directorio inicial de Development Studio Client, cambie al directorio `wdscsampl` de la ubicación correcta en la que ha instalado el producto.
  - b. En la línea de mandatos, escriba: `ftp nombresistpral`, siendo `nombresistpral` el nombre de su sistema principal iSeries; por ejemplo, `PROD400`.
  - c. Escriba su ID de usuario y su contraseña del servidor iSeries.
  - d. En la línea de mandatos, entre `cd /qsys.lib/scenario.lib` para pasar a la biblioteca SCENARIO.
  - e. Entre las siguientes líneas:

```
bin
put WHOLESale.savf WHOLESale.savf
put RETAILSTOR.savf RETAILSTOR.savf
quit
```



Estas líneas (antes de salir) especifican que desea tomar los archivos de salvar del sistema local y colocarlos en el servidor iSeries.

3. De nuevo en la consola de iSeries, restaure la biblioteca Wholesale:
  - a. Entre RSTLIB y pulse F4 para definir cómo desea restaurar la biblioteca.
  - b. En el campo **Biblioteca salvada**, entre WHOLESALE y pulse la tecla de tabulación.
  - c. En el campo **Dispositivo**, entre \*savf y pulse la tecla de tabulación.
  - d. En el siguiente campo, pulse Intro para visualizar valores adicionales y pulse la tecla de tabulación para ir al campo **Archivo de salvar**.
  - e. Especifique WHOLESALE en el campo **Archivo de salvar** y pulse el tabulador.
  - f. En el campo **Biblioteca**, suprima el valor existente y escriba escenario.
  - g. Pulse Intro para restaurar la biblioteca WHOLESALE en el servidor iSeries.
4. Repita este procedimiento para el archivo Retailstor.savf:
  - a. Especifique RSTLIB y pulse F4 para definir cómo desea restaurar la biblioteca.
  - b. En el campo **Biblioteca salvada**, entre RETAILSTOR y pulse la tecla de tabulación.
  - c. En el campo **Dispositivo**, entre \*savf y pulse la tecla de tabulación.
  - d. En el siguiente campo, pulse Intro para visualizar valores adicionales y pulse la tecla de tabulación para ir al campo **Archivo de salvar**.
  - e. Especifique RETAILSTOR en el campo **Archivo de salvar** y pulse el tabulador.
  - f. En el campo **Biblioteca**, suprima el valor existente y escriba escenario.
  - g. Pulse Intro para guardar su acción y restaurar la biblioteca RETAILSTOR en el servidor iSeries.

## Modificar la lista de bibliotecas

Aunque ya ha restaurado las bibliotecas RETAILSTOR y WHOLESALE en el servidor iSeries, tiene que asegurarse de que figuran en la lista de bibliotecas que utilizará cuando inicie sesión. De lo contrario, las aplicaciones Web tal vez no podrían localizar las bibliotecas y se producirían problemas al ejecutar o construir las aplicaciones. También debe asegurarse de que la biblioteca QGPL figura en la lista de bibliotecas.

Para ello, tendrá que actualizar la descripción de trabajo y el perfil de usuario para que los cambios realizados en la lista de bibliotecas sean permanentes.

Cada usuario tiene su manera peculiar de modificar la lista de bibliotecas, pero ahora le proporcionamos un ejemplo de cómo hacerlo:

1. En el servidor iSeries, escriba `dsp1ib1` y pulse Intro para visualizar la lista de bibliotecas.
2. Ve a si en la lista figuran las bibliotecas RETAILSTOR, WHOLESALE y QGPL. Si se encuentran allí, puede salir y cerrar la ventana del emulador. Si no las ve, siga estos pasos:
  - a. Pulse F12 para cancelar la pantalla actual.
  - b. Localice el nombre de la descripción de trabajo. Para ello, escriba `dspusrprf nombreusuario` y pulse Intro para visualizar el perfil de usuario, donde `nombreusuario` es el ID utilizado para iniciar la sesión en el servidor iSeries. La descripción del trabajo se encuentra en una de las páginas del perfil de usuario (utilice Avance página). Cuando encuentre la descripción de trabajo, tome nota de ella y pulse F12 para salir de la pantalla.

- c. En la línea de mandatos, especifique `chgjobd` y pulse F4.
- d. En el campo **descripción de trabajo**, teclee el nombre de la descripción del trabajo y pulse F10 para visualizar parámetros adicionales.
- e. Avance página hasta que vea **Lista de bibliotecas inicial**.
- f. En el campo que figura debajo de **Lista inicial de bibliotecas**, verá "+ para más valores". Teclee un signo más y pulse Intro.
- g. En la página **Especificar más valores**, suprima las entradas existentes y escriba RETAILSTOR, WHOLESale y QGPL, utilizando el tabulador para desplazarse de un valor a otro. Es obligatorio escribir las bibliotecas en este orden, porque las aplicaciones tienen que encontrar la biblioteca RETAILSTOR en primer lugar. Pulse Intro.
- h. Pulse Intro de nuevo para guardar los cambios.
- i. En la línea de mandatos, especifique: `chgusrprf` y pulse Intro.
- j. Escriba el nombre de usuario y pulse Intro.
- k. Pulse F10 para ver los parámetros adicionales.
- l. Avance página hasta que vea **Descripción de trabajo**.
- m. Si el nombre de la descripción de trabajo no coincide con la descripción de trabajo encontrada con el mandato `dspusrprf`, escriba de nuevo el nombre de la descripción del trabajo que ha encontrado con el mandato y pulse Intro para guardar los cambios.

---

## Importar y ejecutar la aplicación en Development Studio Client

En los archivos de programa instalados existen cinco archivos zip para Development Studio Client, que se encuentran por omisión en el directorio `c:\wdsc\wdscsampl`. Estos archivos son:

- SV000514.zip
- SV000618.zip
- SV001585.zip
- SV001586.zip
- SV000501.zip

Importará cada uno de ellos por separado en el entorno de trabajo de Development Studio Client y especificará un archivo Enterprise de archivado (EAR) correspondiente a cada uno de ellos. Un archivo EAR es un archivo Java de archivado (JAR) estándar que tiene la extensión `.ear`. En él puede haber múltiples proyectos Web y se utiliza para empaquetar y desplegar las aplicaciones Web en el servidor administrativo de WebSphere (WAS).

Los cuatro primeros archivos se corresponden con los proyectos Web, y el quinto se corresponde con un proyecto WebFacing. Creará primero los proyectos Web y luego el proyecto WebFacing. Después importará el contenido de los archivos zip a los cinco proyectos.

### Extraer los archivos .zip en el sistema local

Para poder importar el contenido de los archivos .zip del caso práctico en el entorno de trabajo de Development Studio Client, es necesario extraerlos en directorios independientes para poder importarlos como sistemas de archivos, en lugar de como archivos .zip.

**Nota:** Si está utilizando un programa que no sea WinZip, lleve a cabo las tareas que corresponden a ese programa.

1. En el sistema local, navegue a c:\wdsc\wdscsampl. El directorio c:\wdsc puede ser otro si ha elegido efectuar la instalación en una unidad o carpeta base diferente. Si es así, navegue a ese directorio y busque \wdscsampl.
2. Efectúe una doble pulsación en el archivo .zip: SV000501.zip.
3. Pulse el botón Extract y navegue a la misma carpeta donde se encuentran los archivos zip, por omisión, c:\wdsc\wdscsampl.
4. Pulse el botón de selección **All Files** y pulse **Extract**.
5. Si observa el directorio wdscsampl, verá que se ha creado una carpeta SV000501, que contiene todos los archivos necesarios.
6. Ahora que se han extraído todos los archivos en c:\scenario\SV000501, suprima el archivo SV000501.zip para evitar seleccionarlo accidentalmente al importar el sistema de archivos.
7. Repita los pasos de esta sección para los otros cuatro archivos .zip (SV000618.zip, SV001585.zip, SV001586.zip y SV000514.zip.)

## Crear los proyectos Web

En este apartado creará los proyectos Web correspondientes a SV000514, SV000618, SV001585 y SV001586. Los pasos para crear cada proyecto se indican por separado. Aunque le parezca que los pasos se repiten, tenga presente que hay algunas diferencias secundarias que son importantes y de las que debe tomar nota al crear los cuatro proyectos (diferencias que atañen a qué archivos EAR están asociados a cada uno de los proyectos).

Para crear los proyectos Web:

1. Abra IBM WebSphere Development Studio Client para iSeries en el menú Inicio.
2. En Development Studio Client, abra la perspectiva Web con la barra de menús, pulsando **Ventana > Abrir perspectiva > Otras > Web** y, por último, **Aceptar**.
3. Pulse **Archivo > Nuevo > Proyecto Web dinámico**.
  - a. En el campo **Nombre de proyecto**, escriba SV000514.
  - b. Marque el recuadro de selección **Configurar opciones avanzadas** y pulse **Siguiente**.
  - c. Pulse el botón **Nuevo** situado junto al campo **Proyecto EAR**.
  - d. En el campo **Nombre de proyecto**, escriba SVWholeSaleEAR y pulse **Finalizar** en ambos recuadros de diálogo.
4. Pulse **Archivo > Nuevo > Proyecto Web dinámico**.
  - a. En el campo **Nombre de proyecto**, escriba SV001586 y pulse **Siguiente** (El recuadro de selección **Configurar opciones avanzadas** debe seleccionarse automáticamente).
  - b. Compruebe que SWholeSaleEAR está seleccionado en el campo desplegable **Proyecto EAR**.
  - c. Pulse **Finalizar**.
5. Pulse **Archivo > Nuevo > Proyecto Web dinámico**.
  - a. En el campo **Nombre de proyecto**, escriba SV000618 y pulse **Siguiente**.
  - b. Pulse el botón **Nuevo** situado junto al campo **Proyecto EAR**.
  - c. En el campo **Nombre de proyecto**, escriba SVStoreEAR y pulse **Finalizar** en ambos recuadros de diálogo.
6. Pulse **Archivo > Nuevo > Proyecto Web dinámico**.
  - a. En el campo **Nombre de proyecto**, escriba SV001585 y pulse **Siguiente**.

- b. Compruebe que **SVStoreEAR** está seleccionado en el campo desplegable **Proyecto EAR**.
- c. Pulse **Finalizar**.

## Importar los archivos de los proyectos Web

Para importar los sistemas de archivos en el área de trabajo, le indicamos el procedimiento para importar los archivos en SV000514. El procedimiento es igual para los otros tres sistemas de archivos de proyecto Web: SV001586, SV001585 y SV000618. La sección que figura a continuación de ésta describe cómo crear e importar el proyecto WebFacing.

Para importar los archivos:

1. Pulse **Examinar** y navegue a
2. Pulse la carpeta **SV000514** con el botón derecho del ratón y seleccione **Importar**.
3. En el asistente **Importar**, pulse **Sistema de archivos** y, después, **Siguiente**.
4. Desplácese al directorio en el que ha extraído los archivos del caso práctico; por omisión, debe ser `c:\wdsc\wdscsampl`.
5. Pulse **SV000514** y **Aceptar**.
6. Pulse **Seleccionar todo** para asegurarse de que están seleccionados todos los componentes del sistema de archivos SV000501.
7. En el campo **En la carpeta**, escriba **SV000514** si todavía no figura por omisión este valor.
8. Marque el recuadro de selección **Sobrescribir recursos existentes sin avisar**.
9. Pulse **Finalizar** y, si recibe diálogos que le preguntan si desea modificar el directorio raíz de contexto, pulse **Sí**.
10. Repita los pasos del 2 al 9 para SV001586, SV001585 y SV000618, asegurándose de utilizar nombres de proyecto diferentes al completar los pasos.

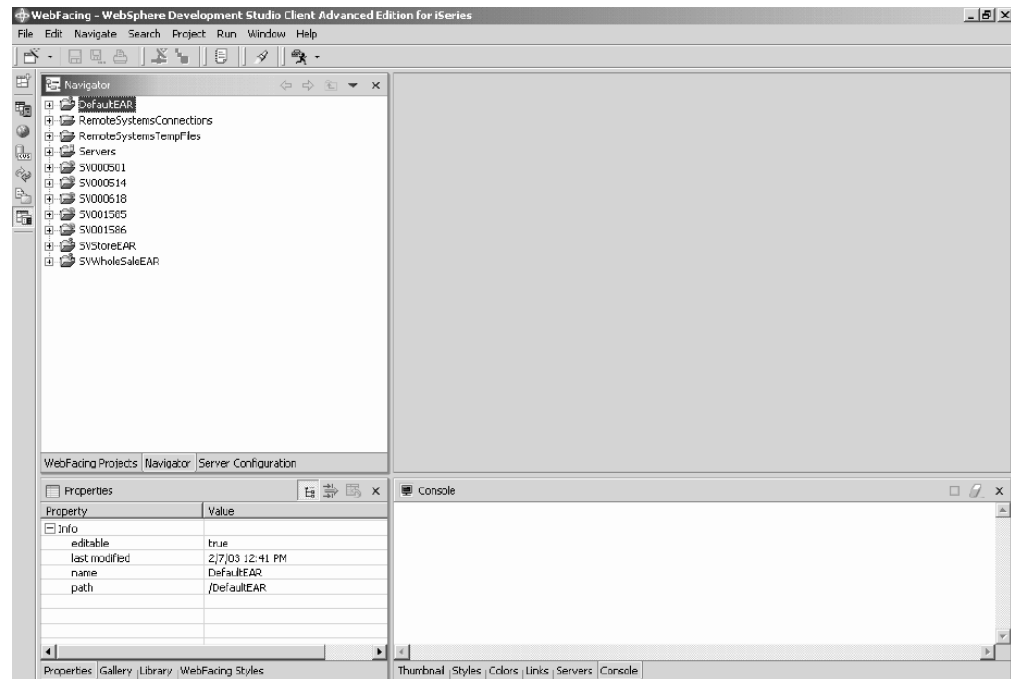
## Importar los archivos de proyecto WebFacing y crear el proyecto WebFacing

El proyecto SV000501 es distinto de los cuatro proyectos Web que acaba de crear, porque SV000501 es un proyecto WebFacing. Los proyectos WebFacing son distintos porque incluyen archivos distintos y tienen una estructura exclusiva específica para manejar los archivos DDS del servidor iSeries.

Para importar los archivos y crear un proyecto WebFacing:

1. Si no está en la vista **Navegador de proyectos**, pulse la pestaña **Navegador de proyectos**.
2. Pulse dentro de la vista con el botón derecho del ratón y seleccione **Importar**.
3. En la ventana **Importar**, seleccione **Proyectos WebFacing** y pulse **Siguiente**.
4. Desplácese al directorio en el que ha extraído los archivos del caso práctico; por omisión, debe ser `c:\wdsc\wdscsampl`.
5. Seleccione **SV000501** y pulse **Aceptar**. Pulse **Siguiente**.
6. En el área **Proyectos WebFacing encontrados** de la ventana, marque el recuadro de selección **SV000501**.
7. En el campo **Proyecto de aplicación de empresa (EAR)**, escriba **SVStoreEAR**.
8. Pulse **Finalizar**.

Ahora que ya ha importado todos los archivos, el área de trabajo debe parecerse a la de esta figura:



Si observa errores junto a los proyectos importados, (indicados mediante x rojas), pulse cada objeto con el botón derecho del ratón y seleccione **Reconstruir proyecto**. Dependiendo de lo que haya en el área de trabajo, esta operación debe borrar la mayoría de los errores para que la aplicación se ejecute correctamente. Los errores que persistan pueden deberse a que parte del código no se ha migrado totalmente al release 5.1 de Development Studio Client. Puede continuar con los ejercicios de este capítulo, que facilitan una introducción al producto, o continuar con el próximo capítulo para intentar construir las aplicaciones desde cero.

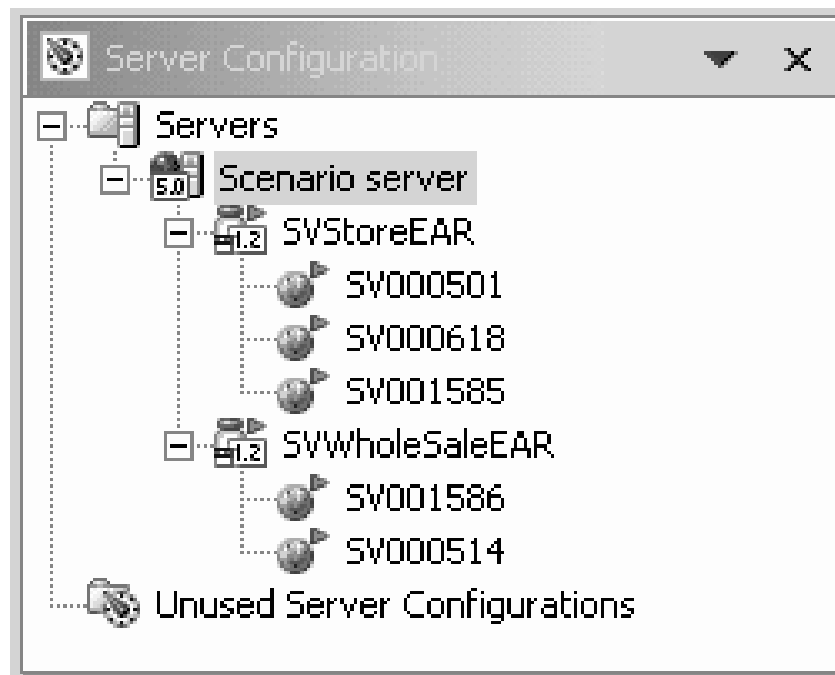
## Configuración de servidor

Ahora que ya ha importado todos los archivos fuente, tendrá que configurar el servidor del entorno de prueba WebSphere para que reconozca las aplicaciones SVWholesaleEAR y SVStoreEAR.

Para configurar el servidor del entorno de prueba WebSphere:

1. Pase a la perspectiva Servidores. En la barra de menú del entorno de trabajo, pulse **Ventana > Abrir perspectiva > Otras > Servidor** y, después, **Aceptar**.
2. En la vista Configuración de servidor, pulse **Servidores** con el botón derecho del ratón y seleccione **Nuevo > Servidor y configuración de servidor**.
3. En el campo **Nombre de servidor**, escriba Servidor Scenario.
4. En el campo **Carpeta**, escriba Carpeta Scenario.
5. En el recuadro **Tipo de servidor**, expanda **WebSphere Versión 5.0** y pulse **Entorno de prueba**, si es que todavía no está seleccionado por omisión.
6. Pulse **Finalizar** y, si recibe mensajes, pulse **Sí**.
7. Para añadir los archivos EAR a la configuración de servidor:
  - a. Expanda **Servidores**.
  - b. Pulse **Scenario server** con el botón derecho del ratón y seleccione **Añadir > SVWholeSaleEAR**.

- c. Expanda **Servidores**.
  - d. Pulse de nuevo **Scenario server** con el botón derecho del ratón y seleccione **Añadir y eliminar proyectos**.
  - e. Pulse el botón **Añadir todo** para trasladar SVStoreEAR y SVWholeSaleEAR al lado derecho de la ventana.
  - f. Pulse **Finalizar**.
8. En la vista Configuración de servidor, expanda **Servidor Scenario**, y luego expanda **SVStoreEAR** y **SVWholeSaleEAR** para ver todas las aplicaciones listadas:



Para comprobar que el servidor ha recogido los proyectos:

1. En la vista Servidores, en la parte derecha de la vista Configuración de servidor (pulse la pestaña **Servidores** si no puede ver la vista Servidores), pulse **Scenario server** con el botón derecho del ratón y seleccione **Publicar**. Aparece un diálogo **Publicación** que muestra cómo avanza el proceso de publicación.
2. Pulse Aceptar para publicar las aplicaciones en el entorno de prueba de WebSphere Application Server.
3. Cuando haya terminado el proceso de publicación, pulse **Aceptar** en el diálogo.
4. En la misma vista, pulse **Servidor Scenario** de nuevo con el botón derecho del ratón y seleccione **Iniciar**. Se abre la vista Consola para mostrar las actividades del servidor. El servidor estará iniciado si ve el mensaje "Servidor *nombreServidor* abierto para e-business" en la parte inferior de la vista. Desplácese hacia abajo para ver todos los mensajes.

## Configurar el proyecto WebFacing

Antes de ejecutar la aplicación o de trabajar con el módulo SV000501, tendrá que iniciar el servidor WebFacing para que se ejecute la aplicación, y deberá configurar el proyecto para que utilice el servidor iSeries correcto.

Para iniciar el servidor WebFacing:

1. Abra un emulador 5250 de pantalla verde e inicie sesión con su ID de usuario y su contraseña.

2. En la línea de mandatos, entre `strtcpsvr *webfacing`.
3. En la parte inferior de la pantalla debe ver el mensaje "Se está iniciando el servidor WEBFACING".

Ahora tendrá que cambiar las propiedades de WebFacing, pasando de nuevo al entorno de trabajo de Development Studio Client:

1. Pase a la perspectiva Web (para pasar de una perspectiva a otra, puede pulsar los iconos agrupados en la parte izquierda de la pantalla).
2. Pulse la pestaña **Navegador de proyectos** para poder ver las estructuras de los proyectos.
3. Expanda **SV000501** y pulse dos veces en **Descriptor de despliegue Web**.
4. Pulse la pestaña **Parámetros**.
5. Pulse **WFDefaultHost**.
6. En el campo **Valor**, borre el valor existente y escriba el nombre de su servidor iSeries.
7. En la barra de menús del entorno de trabajo, pulse el icono de Guardar o bien pulse **Archivo > Guardar descriptor de despliegue Web**. Pulse **Aceptar** si recibe algún mensaje. Cierre el archivo.

## Definir información de servidor

Ahora tendrá que definir la información de sistema principal iSeries para los cinco proyectos, asegurándose de que están configurados para ejecutarse con su sistema principal iSeries bajo su ID de usuario y su contraseña. Para definir información de servidor:

1. Asegúrese de que está en la perspectiva Web.
2. En la vista Navegador de proyectos, pulse **SV000501** con el botón derecho del ratón y seleccione **Especificar configuración de tiempo de ejecución de herramientas Web de iSeries**.
3. En el campo **Nombre de servidor iSeries**, suprima el valor existente y escriba el nombre del Servidor iSeries.
4. Suprima los otros valores existentes y entre su ID de usuario y su contraseña.
5. Pulse **Finalizar** (y, si es necesario, vuelva a pulsar **Finalizar**).
6. Repita los pasos del 2 al 5 para los otros cuatro proyectos: SV000514, SV000618, SV001585 y SV001586.

---

## Ejecutar la aplicación en el entorno de trabajo

Ahora ya puede ejecutar la aplicación en el entorno de prueba WebSphere, en el entorno de trabajo. Cuando compruebe que se está ejecutando satisfactoriamente, podrá exportar el archivo EAR al servidor iSeries y desplegarlo en el servidor WebSphere Application Server de iSeries.

## Ejecutar la aplicación en calidad de cliente en el entorno de trabajo

Como cliente, empezaría por la página Web de compras en la tienda al por menor, donde examinaría los artículos y luego solicitaría una cantidad y una talla.

**Nota:** pueden surgir problemas si intenta ejecutar la aplicación detrás de un cortafuegos, porque el archivo `web.xml` busca el archivo

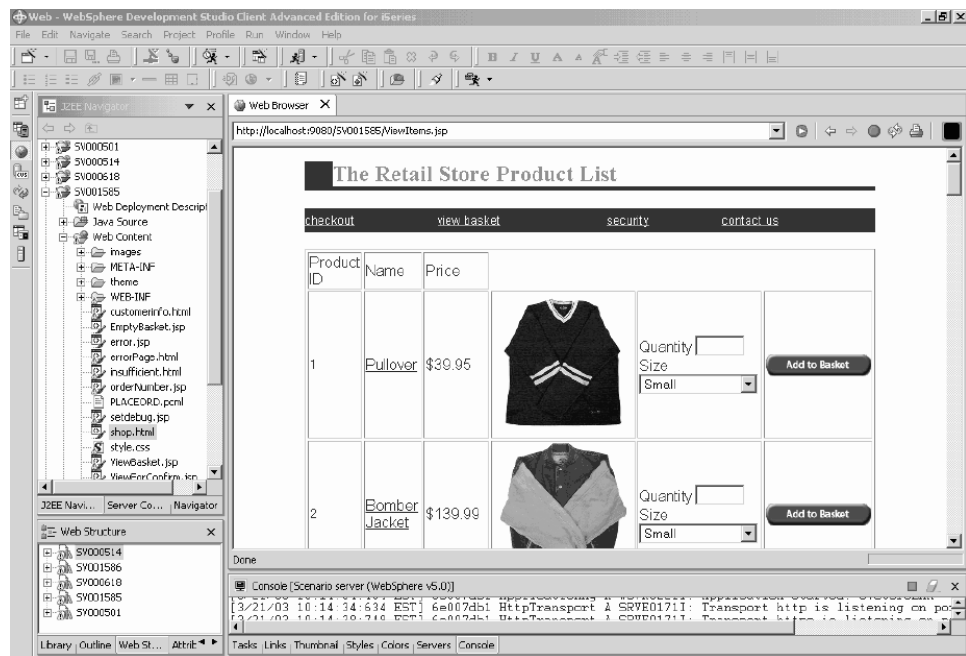
http://java.sun.com/j2ee/dtds/web-app\_2\_2.dtd. Para solucionar este problema, antes de ejecutar la aplicación, cambie la sentencia DOCTYPE de todos los archivos web.xml para que sea:

```
!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"x:/wdsc/eclipse/plugins/com.ibm.etools.j2ee/dtds/web-app_2_2.dtd"
```

Aquí x:/wdsc es el directorio en el que ha instalado el producto. Normalmente, encontrará sus archivos web.xml bajo *Nombre de proyecto* > **Web Content** > **WEB-INF**. Después de abrir el archivo, pulse la pestaña **Fuente** para editar el archivo directamente.

Para ejecutar la aplicación como cliente en el entorno de trabajo:

1. En la perspectiva Web, expanda **SV001585** > **Web Content**.
2. Pulse **shop.html** con el botón derecho del ratón y seleccione **Ejecutar en servidor**. Así se lanza la aplicación en el navegador del entorno de trabajo.
3. Pulse **Finalizar**.
4. Entre en la aplicación pulsando la imagen de las camisetas.
5. Intente escribir valores en la próxima página, haciendo ver que es usted un cliente que solicita un pullóver o una chaqueta, seleccionando una talla y la cantidad, y añadiendo artículos a la cesta de la compra:



La página que visualiza el texto "Su pedido se ha procesado" es la última del ejemplo. Cuando haya terminado de utilizar el ejemplo, puede cerrar el navegador.

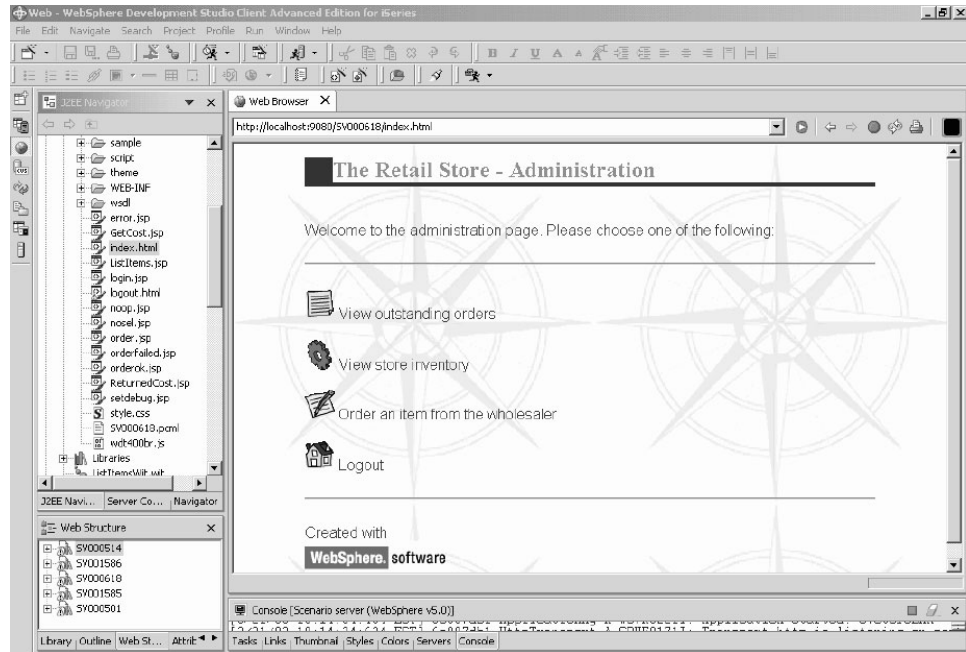
## Ejecutar la aplicación en calidad de administrador en el entorno de trabajo

Como administrador, empezaría por la página Web de administración de la tienda al por menor y comprobaría los pedidos o el inventario de la tienda. Para ejecutar la aplicación como administrador en el entorno de trabajo:

1. En la perspectiva Web, expanda **SV000618** > **Web Content**.
2. Pulse **index.html** con el botón derecho del ratón y seleccione **Ejecutar en servidor**. Así se lanza la aplicación en el navegador del entorno de trabajo.



3. Pulse el icono situado a la izquierda de **Ver inventario de la tienda** para visualizar la página siguiente, como si fuese un administrador que decide lo que hay que comprar para su tienda:



Vaya pulsando los iconos para visualizar las páginas administrativas.

---

## Desplegar la aplicación del caso práctico en WebSphere Application Server

Ahora que ya ha ejecutado la aplicación del caso práctico de iSeries en el entorno de prueba del entorno de trabajo, puede desplegar la aplicación en WebSphere Application Server, como si estuviera ejecutándola en la vida real. Sin embargo, antes de desplegar la aplicación, hay que hacer algunos ajustes, como proteger la página del administrador y cambiar el URL de un servicio Web para que la aplicación señale hacia el lugar apropiado, tal como se explica en los siguientes apartados.

**Nota:** La tarea de desplegar en WebSphere Application Server es opcional; aunque no pruebe la aplicación en WebSphere Application Server, puede continuar en el próximo capítulo y llevar a cabo los módulos.

### Configurar WebSphere Application Server

Anteriormente, en este mismo capítulo, vimos el apartado “Ejecutar la aplicación en el entorno de trabajo” en la página 17. Ahora que ya ha desplegado los archivos en el servidor iSeries, puede ejecutar la aplicación en el iSeries mediante WebSphere Application Server.

Para desplegar la aplicación en WebSphere Application Server (opcional), debe asegurarse de que:

- Ha creado (o tiene acceso a) una instancia de la versión 4.0 o de la versión 5.0 de WebSphere Application Server, o una instancia de WebSphere Application Server Express en el sistema principal iSeries, y la instancia se está ejecutando (solo si desea probar el despliegue en WebSphere Application Server).

- Sabe cuáles son los números de puerto de las instancias de HTTP y WebSphere Application Server en el sistema principal iSeries.
- Tiene instalada la versión 4.0 ó 5.0 de la consola administrativa de WebSphere en la estación de trabajo.

**Nota:** En el caso de la versión 5.0 de WAS, la consola está basada en navegador, por lo que no necesita tener una consola en la estación de trabajo. Consulte la documentación en línea para obtener más información sobre las diversas versiones de WAS.

Las aplicaciones de iSeries habilitadas para Web utilizan WebSphere Application Server para ejecutar los servlets Java y los archivos JavaServer Pages™ (JSP) que sirven para comunicar entre el navegador Web del usuario y los programas o datos del iSeries. Para servir sus páginas HTML y sus archivos JSP desde el mismo sistema iSeries, también necesitará un servidor HTTP en ese sistema. Le recomendamos que utilice IBM HTTP Server (powered by Apache). Para obtener documentación sobre cómo utilizar este servidor, visite el Centro de documentación de IBM HTTP Server para iSeries en [http://publib.boulder.ibm.com/pubs/html/iseries\\_http/v5r1/index.htm](http://publib.boulder.ibm.com/pubs/html/iseries_http/v5r1/index.htm).

WebSphere Application Server ejecuta los archivos JavaServer Pages, los beans Java y los servlets Java que se generan en los diversos procesos. Los recursos de documentación primaria de IBM WebSphere Application Server para iSeries y de la consola administrativa de IBM WebSphere para iSeries están disponibles en los siguientes sitios Web:

- IBM WebSphere Application Server Versión 4.0 Advanced Edition para iSeries en <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Versión 4.0 Advanced Single Server Edition para iSeries en <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>

Le recomendamos encarecidamente que se familiarice con la documentación de IBM WebSphere Application Server, concretamente, con los apartados que tratan sobre *Módulos J2EE*, *Instalación de WebSphere Application Server* y *Configurar múltiples instancias del servidor administrativo de WebSphere*. Como mínimo, deberá seguir los pasos que figuran en el enlace *Instalación*.

Utilice el mapa del sitio para localizar información sobre cómo instalar, configurar y obtener los PTF necesarios para la consola administrativa de WebSphere.

## Proteger la página del administrador

La página del administrador, `index.html`, solo debe estar accesible para las personas autorizadas y, por lo tanto, hay que protegerla como es debido. Esto se puede hacer programáticamente, en la lógica de la aplicación Web, o bien mediante la característica de seguridad de WebSphere. En este caso práctico, hemos utilizado la seguridad de WebSphere para proteger la página. Observe que hemos utilizado WebSphere Application Server V4.0 Advanced Edition. Podrá encontrar información sobre WebSphere Application Server en los correspondientes sitios Web:

- IBM WebSphere Application Server Versión 4.0 Advanced Edition para iSeries en <http://publib.boulder.ibm.com/was400/40/AE/english/docs/>
- IBM WebSphere Application Server Versión 4.0 Advanced Single Server Edition para iSeries en <http://publib.boulder.ibm.com/was400/40/AEs/english/docs/>
- Página de RedBooks de WebSphere: Busque los RedBooks de WebSphere Application Server Versión 5.0 y WebSphere Application Server Express Edition

Versión 5.0 en <http://publib-b.boulder.ibm.com/redbooks.nsf/portals/WebSphere>

Si está utilizando una versión distinta de WebSphere Application Server, consulte la documentación de su versión para saber cómo proteger los recursos Web.

Puede configurar la seguridad de recursos Web, como las páginas Web y los servlets, en Development Studio Client o bien en la herramienta de ensamblado de aplicaciones (AAT). En este caso práctico utilizamos Development Studio Client.

Para revisar la configuración y las propiedades de la seguridad de esta aplicación Web:

1. En Development Studio Client, pase a la perspectiva Web.
2. En la vista Navegador, expanda **SV000618 > Web Content > WEB-INF**.
3. Pulse dos veces en el archivo **web.xml** para abrir la vista de web.xml.
4. Pulse la pestaña **Seguridad**.
5. Para proteger el archivo **index.html**, que es la página de administración, se ha definido también una restricción de seguridad. En la parte superior de la vista, pulse **Restricciones de seguridad**.
6. Pulse la primera instancia de **SecurityConstraint** que figura en la lista.
7. En la parte derecha, pulse **AdminPage**.
8. Pulse **Editar** para invocar el diálogo **Colecciones de recursos Web**. Observe que los métodos GET y POST de **index.html** están preseleccionados.
9. Pulse **Aceptar**.

En la sección **Roles de seguridad**, fíjese en el rol de seguridad definido que se llama "Administrador". Durante el despliegue, se asigna este rol a las personas que, por ello, obtienen acceso a la página **index.html**. En la sección **Roles autorizados**, observe que hemos otorgado al rol de Administrador acceso a esta restricción de seguridad. Con la aplicación de esta medida de seguridad, solo se otorga acceso a la página **index.html** a los usuarios a quienes se asigna el rol de Administrador, después de que hayan proporcionado las debidas credenciales, como pueden ser un ID de usuario y una contraseña. Cuando un recurso está protegido, WebSphere Application Server intenta en primer lugar autenticar al usuario. La autenticación se lleva cabo mediante certificados o solicitando un ID de usuario y una contraseña al usuario. Esta solicitud se puede realizar con el diálogo de autenticación básico o bien con un formulario personalizado.

En este caso práctico, hemos diseñado nuestra propia página de inicio de sesión, que se llama **login.jsp**. Para configurar la solicitud de autenticación de esta página, seleccione la pestaña **Páginas** en la vista **web.xml**. En la sección **Inicio de sesión**, observe que el método de autenticación preseleccionado es **Formulario**. Fíjese también en que el nombre de la página de inicio de sesión es **login.jsp**. Cuando el inicio de sesión no es satisfactorio, se visualiza la página Error. En este caso, la aplicación vuelve a visualizar la página **login.jsp**.

## Crear archivos EAR para el despliegue en WebSphere Application Server para iSeries

Necesita crear archivos EAR para desplegar su aplicación en el producto WebSphere Application Server para iSeries. Un archivo EAR es un archivo Java de archivado (JAR) estándar que tiene la extensión **.ear**. En él puede haber múltiples proyectos Web y se utiliza para empaquetar y desplegar las aplicaciones Web en el servidor administrativo de WebSphere (WAS).

Para crear los archivos EAR:

1. Pase a la perspectiva Web.
2. En la vista Navegador, pulse **SVStoreEAR** con el botón derecho del ratón y seleccione **Exportar**.
3. En la ventana de exportación, pulse **Archivo EAR** y, después, **Siguiente**.
4. Bajo **¿A dónde desea exportar los recursos?**, pulse **Examinar** y navegue hasta un directorio del sistema de archivos integrado de iSeries en el que pueda guardar los archivos EAR. (Debe haber correlacionado su unidad de red con un IFS de iSeries).
5. Entre **SVRetailStorEAR.ear** en el campo **Nombre de archivo** y pulse **Abrir**.
6. Pulse **Finalizar**.
7. Repita los pasos del 2 al 6 para **SVWholeSaleEAR**.

## Desplegar los archivos EAR en WebSphere Application Server para iSeries

Ahora que ya ha creado los archivos EAR, puede desplegarlos en WebSphere Application Server.

1. Abra la consola administrativa de WebSphere.
2. Pulse **Aplicaciones de empresa** con el botón derecho del ratón y seleccione **Instalar aplicación de empresa**.
3. Marque el botón de selección **Instalar aplicación (\*ear)**.
4. Pulse el botón **Examinar** superior (el inferior no está disponible).
5. Navegue hasta el directorio del IFS al que exportó los archivos EAR.
6. Seleccione **SVRetailStorEAR.ear**.
7. Escriba "RETAILSTOR" en el campo **Nombre de aplicación**.
8. Pulse **Siguiente** y escriba su ID de usuario del servidor iSeries para el cometido de administrador.
9. Pulse **Siguiente** repetidas veces hasta llegar a la página titulada **Seleccionar sistemas principales virtuales para módulos Web**.
10. Para los tres módulos Web, pulse **Seleccionar sistema principal virtual** y seleccione su servidor virtual preferido en la lista desplegable. (Si no está seguro de cuál debe seleccionar, utilice el valor **default** o **default\_host**).
11. Pulse **Siguiente**.
12. Para los tres módulos Web, pulse **Seleccionar servidor** y seleccione el servidor que desea utilizar. (Si no está seguro de cuál debe seleccionar, utilice el valor **Default Server**).
13. Pulse **Siguiente**.
14. Pulse **Finalizar** y, después, **Aceptar** en el diálogo.
15. Pulse de nuevo **Aplicaciones de empresa** con el botón derecho del ratón y seleccione **Instalar aplicación de empresa**.
16. Pulse el botón **Examinar** inferior (el superior no está disponible).
17. Navegue hasta el directorio del IFS en el que colocó los archivos EAR.
18. Seleccione **SVWholesaleEAR.ear**.
19. Escriba "WHOLESALE" en el campo **Nombre de aplicación**.
20. Pulse **Siguiente** repetidas veces, hasta que se habilite **Finalizar**.
21. Pulse **Finalizar** y, después, **Aceptar** en el diálogo.

**Nota:** Podrían pasar unos minutos antes de que aparezca un mensaje de confirmación.

## **Ejecutar la aplicación en WebSphere Application Server**

Para desempeñar el rol de cliente y ejecutar el punto de entrada de la tienda al por menor, escriba el siguiente URL en un navegador Web:

`http://suNombreSistPraliSeries:suNúmeroPuertoHTTP/SV001585/shop.html`

Para desempeñar el rol de administrador y ejecutar el punto de entrada del suministrador al por mayor, escriba el siguiente URL en un navegador Web:

`http://suNombreSistPraliSeries:suNúmeroPuertoHTTP/SV000618/index.html`

Si no sabe cuál es el número de puerto HTTP, pregúnteselo al administrador de WebSphere Application Server.



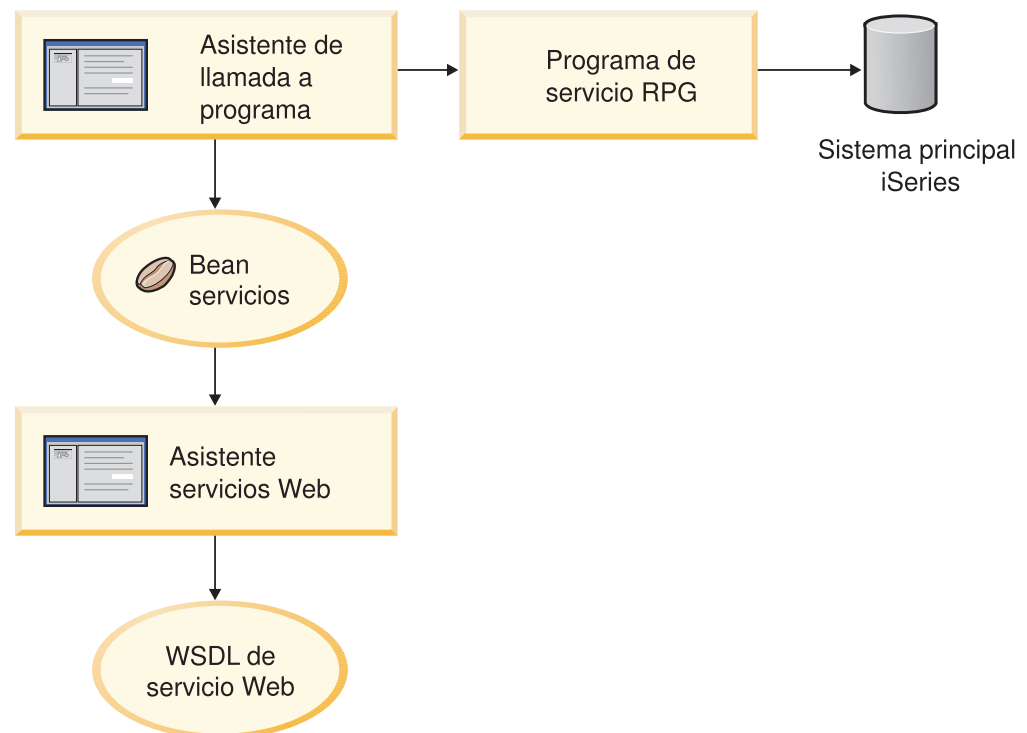
---

## Capítulo 4. Módulo 1 paso a paso: Crear un servicio Web que devuelva los precios de los productos (SV000514)

---

### Introducción

En este módulo, a partir de un programa RPG existente en el servidor iSeries, creará un servicio Web para visualizar los precios de los productos de su inventario. En primer lugar creará un programa de servicio RPG con un procedimiento que pueda recuperar el coste de un artículo a partir de una base de datos de iSeries, dado el número del artículo. Utilizará el asistente de llamada a programa de las herramientas de desarrollo Java para iSeries con el fin de invocar el programa RPG y crear un bean de servicios. Después utilizará el asistente de servicios Web para crear un servicio Web, y lo verificará mediante el ejemplo generado.



---

### Antes de empezar

Solo podrá llevar a cabo los ejercicios si se cumplen los siguientes prerequisites. En el tema Capítulo 3, “Ejecutar el caso práctico”, en la página 9 hallará más detalles sobre los prerequisites.

- Tiene acceso TCP/IP a un servidor iSeries.
- Tiene aplicados los PTF más recientes del servidor iSeries. Consulte nuestra página de soporte para obtener información acerca de los PTF y paquetes de servicio: <http://www.ibm.com/software/awdtools/wdt400/support/>
- Ha iniciado los servidores iSeries con el mandato STRTCPSVR \*ALL.
- Ha iniciado el servidor WebFacing con el mandato STRTCPSVR \*WEBFACING.

- Ha llevado a cabo todas las tareas del Capítulo 3, “Ejecutar el caso práctico”, en la página 9 (salvo las tareas de WebSphere Application Server opcionales, que no son necesarias para probar la aplicación en el entorno de trabajo).

---

## Crear un proyecto Web nuevo

El primer paso para crear este servicio Web consiste en crear un proyecto Web nuevo en el que se guardará la información.

1. En el IDE del entorno de trabajo, pase a la perspectiva Web o ábrala pulsando **Ventana > Abrir perspectiva > Otras > Web > Aceptar**.
2. Pulse **Archivo > Nuevo > Proyecto Web dinámico**.
3. En el campo **Nombre de proyecto**, escriba Project514.
4. Marque el recuadro de selección **Configurar opciones avanzadas** y pulse **Siguiente**.
5. Pulse el botón **Nuevo** situado junto al campo **Proyecto EAR**.
6. En el campo **Nombre de proyecto**, escriba Project514EAR y pulse **Finalizar** en ambos recuadros de diálogo.

Ahora podrá ver que los proyectos Project514 y Project514EAR se han añadido a su área de trabajo en la vista Navegador de proyectos.

---

## Definir la información del servidor iSeries

Después de crear el proyecto Web, tiene que definir el servidor iSeries utilizado por el proyecto para obtener información.

1. Pulse el proyecto **Project514** con el botón derecho del ratón y seleccione **Especificar configuración de tiempo de ejecución de herramientas Web de iSeries**.
2. Escriba el nombre del servidor iSeries en el que reside la biblioteca WHOLESALE restaurada; por ejemplo, PROD400.
3. Escriba el ID de usuario y la contraseña de este servidor iSeries.
4. Escriba Wholesale en el campo **Biblioteca** y pulse **Añadir**.
5. Pulse **Finalizar**.

---

## Crear el programa de servicio RPG

Quiere que su aplicación recupere el precio de un artículo, dado el número del artículo. Esto lo lleva a cabo un programa de servicio RPG que contiene un procedimiento que se llama QryProdCost. La biblioteca WHOLESALE contiene un programa de servicio RPG que se llama CWWSSRV. Este programa contiene un procedimiento QryProdCost que puede tomar los números de los artículos como entrada, abrir el archivo de inventario dentro de la biblioteca WHOLESALE, recuperar el precio a partir de la base de datos del inventario y devolver el precio. Para ajustarse a esta tarea, la interfaz tiene dos parámetros, uno para el número de artículo y otro para el precio. Si no se encuentra el número de artículo o el precio, el programa RPG devuelve un mensaje a la interfaz.

Para crear este servicio Web, primero creará con el asistente de llamada a programa un bean Java que invoque el procedimiento RPG QryProdCost. Luego utilizará los servicios Web para habilitar el procedimiento RPG como servicio Web mediante el bean Java.

Para crear el bean Java:



1. En la perspectiva Web, pulse **Project514** con el botón derecho del ratón y seleccione **Nuevo > Otros**.
2. Seleccione **iSeries > Java** en el panel izquierdo de la ventana y, a continuación, seleccione **Bean de llamada a programa** en el panel derecho.
3. Pulse **Siguiente** para invocar el asistente de llamada a programa.
4. En el campo **Nombre de bean Java**, debajo del título **Añadir programa**, escriba **Inventory**.
5. En el campo **Objeto programa**, escriba **CWSSRV**, que es el nombre del programa de servicio RPG.
6. En el campo **Biblioteca**, escriba **WHOLESALE**.
7. En la lista desplegable **Tipo de programa**, seleccione **\*SRVPGM**.
8. En el campo **Punto de entrada**, escriba **QryProdCost**.
9. Pulse **Aceptar** para añadir la definición del programa.

## Crear los parámetros y generar el bean Java

Ahora que ya ha especificado el programa, puede añadir los parámetros. El programa RPG CWSSRV contiene estos dos parámetros:

- Parámetro de número de artículo: el programa emplea este número para localizar el artículo en la base de datos.
- Parámetro de coste del artículo: el programa emplea este número para localizar el coste del artículo en la base de datos.

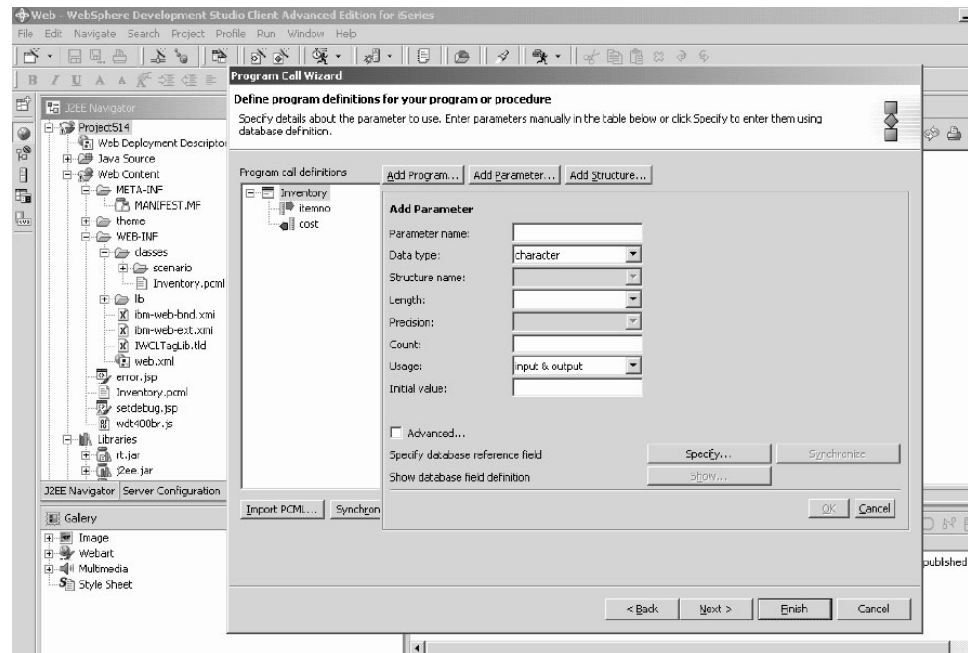
Para añadir el parámetro de número de artículo:

1. En el panel izquierdo del asistente de llamada a programa, pulse la definición de llamada a programa **Inventory** para seleccionarla. Con esta acción se repueblan los campos de la parte derecha de la página.
2. Pulse **Añadir parámetro**.
3. En el campo **Nombre de parámetro**, escriba **itemno**.
4. En la lista desplegable **Tipo de datos**, seleccione **decimal empaquetado**.
5. En el campo **Longitud**, escriba **5**.
6. En el campo **Precisión**, escriba **0**.
7. En la lista desplegable **Utilización**, seleccione **entrada**.
8. Pulse **Aceptar** para añadir este parámetro. En el panel de la izquierda, fíjese en que aparece **itemno** bajo **Inventory**. Ahora ya está listo para añadir el segundo parámetro.

Para entrar el parámetro de coste de artículo:

1. En el panel izquierdo del asistente de llamada a programa, pulse de nuevo la definición de llamada a programa **Inventory** para seleccionarla. Con esta acción se repueblan los campos de la parte derecha de la página.
2. Pulse **Añadir parámetro**.
3. En el campo **Nombre de parámetro**, escriba **cost**.
4. En la lista desplegable **Tipo de datos**, seleccione **decimal empaquetado**.
5. En el campo **Longitud**, escriba **7**.
6. En el campo **Precisión**, escriba **2**.
7. En la lista desplegable **Utilización**, seleccione **salida**.
8. Pulse **Aceptar** para añadir este parámetro. En el panel de la izquierda, fíjese en que aparece **cost** bajo **Inventory**. En este momento, el aspecto del asistente debe ser como el de esta figura. Observe que el icono situado a la izquierda del

parámetro indica si es de tipo entrada, entrada y salida, o salida:



Para definir el nombre de paquete de estos parámetros, todavía en el asistente de llamada a programa:

1. Pulse **Siguiente**.
2. En el campo **Paquete**, escriba `scenario` como nombre de paquete. Deje los valores por omisión en los otros campos.
3. Quite la marca del recuadro de selección **Aplicación Java**.

**Nota:** Revise la lista de archivos bajo "El asistente generará estos archivos" y observe que el nombre del bean Java generado será `InventoryServices.java`.

4. Pulse **Finalizar** para generar los archivos.

## Crear un servicio Web a partir del bean Java

Tras crear el bean Java que invoca el programa RPG, el paso siguiente consiste en convertir el bean en un servicio Web para que los otros programas puedan acceder al mismo programa RPG por Internet.

Al crear el servicio Web, el asistente de servicios Web genera los archivos WSDL que se distribuyen a los usuarios que necesitan utilizar el servicio Web. Para crear el servicio:

1. En la vista Navegador de la perspectiva Web, expanda **Project514 > Java Source > scenario**.
2. Pulse **InventoryServices.java** con el botón derecho del ratón y seleccione **Nuevo > Otros**.
3. En la ventana Nuevo, pulse **Servicios Web**; pulse **Servicio Web** en la parte derecha de la ventana y luego pulse **Siguiente**.
4. Marque los recuadros de selección **Generar un proxy** y **Probar el proxy generado**.
5. Pulse **Siguiente** tres veces y observe que se generan los archivos necesarios. Cuando llegue a la ventana Identidad de bean Java de servicio Web, pulse

**Finalizar.** Puede que sea necesario pulsar **Finalizar** una segunda vez si observa errores. Si no puede finalizar el asistente, pulse simplemente **Cancelar**; la mayoría de los archivos necesarios se crearán de todas formas.

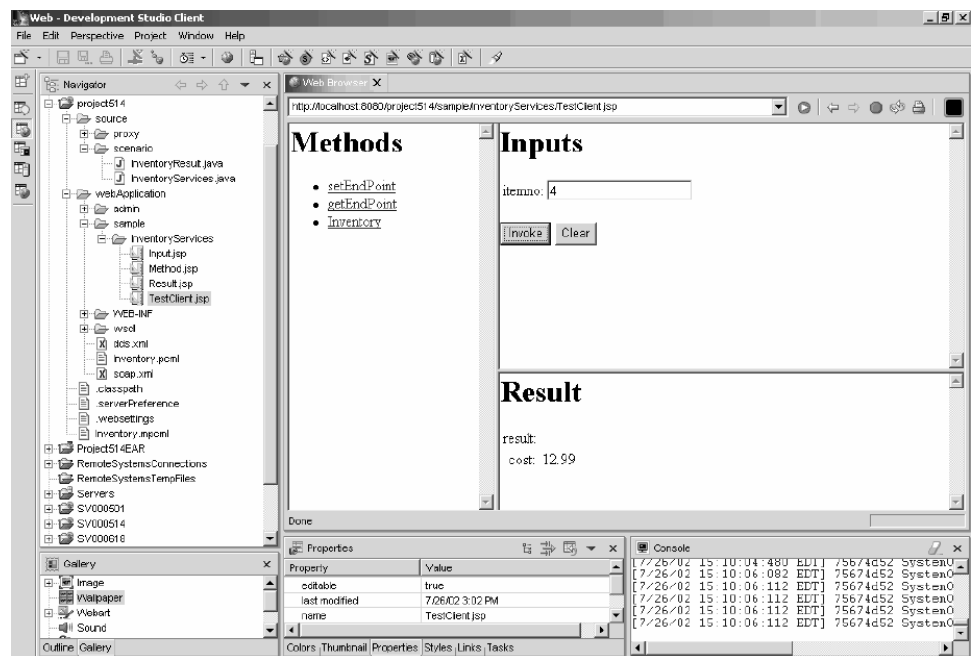
## Probar el ejemplo

Cuando creó el servicio Web, una de las instrucciones era solicitar la generación de un ejemplo. Dado que la opción estaba seleccionada, el asistente de servicios Web creó un proyecto nuevo con páginas de prueba que le permitirán probar el servicio Web. En la vista **Navegador**, observe que el nombre de este proyecto nuevo es **Project514Client**. Fíjese asimismo en que la herramienta ha abierto automáticamente el archivo `TestClient.jsp` en la parte derecha del entorno de trabajo. Este archivo está situado bajo **Project514Client > Web Content > sample > InventoryServices**.

Para probar el ejemplo:

1. En la página cargada, desplácese hacia abajo en el panel de métodos hasta que llegue al final del todo. Pulse el método **Inventory (java.math.BigDecimal)**.
2. En el campo **itemno**, escriba 4 y pulse **Invocar**.
3. Verifique que como salida se obtiene:

```
result:  
cost: 12.99
```



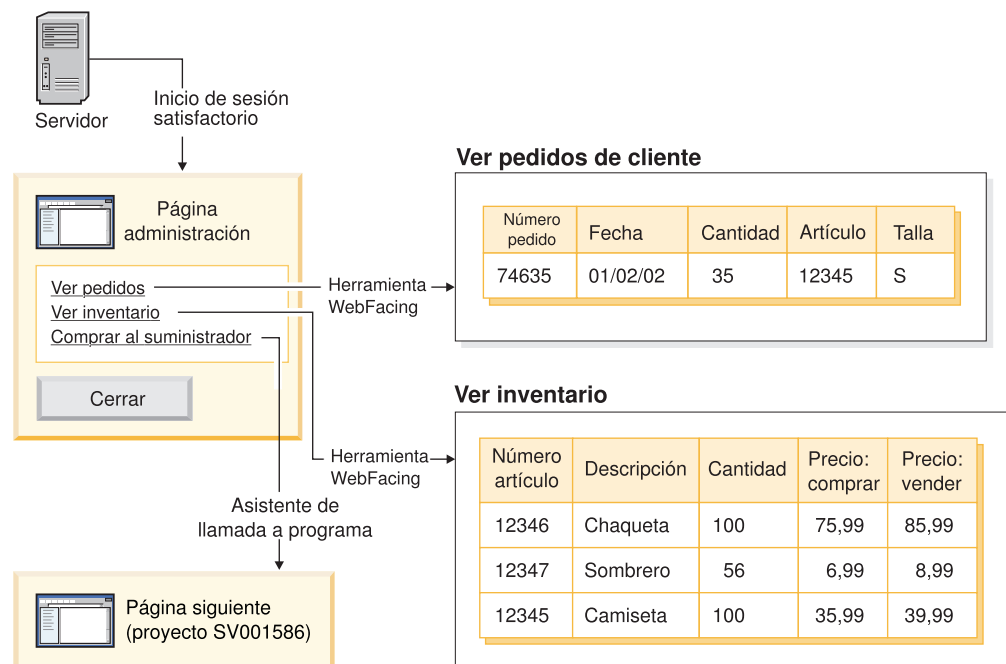
Una vez que haya confirmado el resultado, habrá terminado con el módulo. Ha creado un servicio Web que devuelve precios de productos en la interfaz de la aplicación.



# Capítulo 5. Módulo 2 paso a paso: Crear una interfaz para ver el inventario y formular pedidos de artículos (SV000501)

## Introducción

En este módulo creará una interfaz que visualiza las opciones de operación de un administrador mediante la herramienta IBM WebFacing. Después de un inicio de sesión satisfactorio en el servidor iSeries, la interfaz visualiza el inventario disponible y se pueden comprobar los pedidos existentes. Como administrador, podrá entonces comprar mercancía al suministrador al por mayor.



En este proyecto se trabaja con dos programas y dos archivos de pantalla restaurados en el servidor iSeries. Los nombres de los programas son ViewInventory y ViewOrder. Los programas utilizan los dos siguientes archivos de pantalla: ORDERDSP y QUERY. Estos dos archivos contienen valores Web personalizados para los archivos JSP generados por la Herramienta WebFacing, y se utilizan para las imágenes y los hipervínculos. El valor Web de imagen le permite utilizar el contenido de un campo para generar el nombre del archivo de imagen y visualizar la imagen que hay en el archivo JSP. El valor Web de hipervínculo le permite invocar otra aplicación Web cuando pulsa la imagen en el archivo JSP. Puede utilizar el Editor CODE o el Diseñador CODE para comprobar el código fuente del archivo de pantalla con objeto de determinar cómo especificar los valores Web.

---

## Antes de empezar

Solo podrá llevar a cabo los ejercicios si se cumplen los siguientes prerequisites. En el tema Capítulo 3, "Ejecutar el caso práctico", en la página 9 hallará más detalles sobre los prerequisites.

- Tiene acceso TCP/IP a un servidor iSeries.
- Ha iniciado los servidores iSeries con el mandato STRTCPSVR \*ALL.
- Ha iniciado el servidor WebFacing con el mandato STRTCPSVR \*WEBFACING.
- Ha llevado a cabo todas las tareas del Capítulo 3, "Ejecutar el caso práctico", en la página 9 (salvo las tareas de WebSphere Application Server opcionales, que no son necesarias para probar la aplicación en el entorno de trabajo).

---

## Crear un proyecto WebFacing

Lo primero que debe hacer es crear un proyecto WebFacing y especificar los mandatos CL relevantes. Para crear el proyecto WebFacing:

1. En el entorno de trabajo, pase a la perspectiva WebFacing pulsando uno de los iconos de perspectiva que están en la parte inferior de la parte izquierda del área de trabajo o bien pulsando **Ventana > Abrir perspectiva > Otras > WebFacing** y, por último, **Aceptar**.
2. Cree un proyecto WebFacing nuevo pulsando **Archivo > Nuevo > Proyecto WebFacing**.
3. Dé el nombre Project501 al proyecto.
4. Pulse el botón de selección **Existente** situado junto a **Proyecto de aplicación de empresa**.
5. Escriba SVStoreEAR en el valor del campo **Nombre de proyecto existente**; tenga en cuenta que el valor es sensible a mayúsculas y minúsculas. Pulse **Siguiente**.
6. En la lista desplegable **Nivel de J2EE**, seleccione **1.3**. Pulse **Siguiente**.
7. En el campo **Conexión**, debe haberse escrito automáticamente el nombre del servidor iSeries. Si no es así, pulse **Nuevo**, especifique el nombre del servidor iSeries en el diálogo y pulse **Finalizar**.
8. Pulse **Renovar lista de DDS** y escriba su ID de usuario y su contraseña en el diálogo emergente para renovar la lista.
9. En la lista de bibliotecas generadas, expanda **RETAILSTOR**.
10. Pulse **QDDSSRC** y, a continuación, la flecha que señala hacia la derecha (>>) para mover los archivos.
11. Pulse **Siguiente** dos veces, hasta que llegue a la página **Especificar mandatos CL**.

Ahora tendrá que añadir un mandato y una etiqueta de mandato para el administrador:

12. En el campo **Mandato CL**, escriba la llamada call viewinvent.
13. En el campo **Etiqueta de mandato**, suprima el valor existente y escriba Ver inventario.
14. Pulse el botón de selección **Iniciar sesión con valores especificados**.
15. Pulse **Añadir** y observe la adición en la lista de la parte inferior de la ventana. Ahora tendrá que añadir un segundo mandato y su etiqueta para el cliente:
16. En el campo **Mandato CL**, suprima el valor existente y escriba call vieworder.
17. En el campo **Etiqueta de mandato**, suprima el valor existente y escriba Ver pedidos.

18. Marque el botón de selección **Solicitud de inicio de sesión**.
19. Pulse **Añadir** y observe la adición en la lista de la parte inferior de la ventana. Pulse **Siguiente**.
20. En la ventana **Elegir un estilo Web**, vaya desplazándose por los estilos para ver los valores disponibles. Dado que más adelante tendremos que recuperar información a partir de este estilo, seleccione **avenue** para este módulo. Pulse **Siguiente**.
21. Verifique que hay una marca de selección en el recuadro **No. Solo deseo crear el proyecto ahora** y pulse **Finalizar**.
22. Si ve un mensaje que indica Reparar configuración de servidor, pulse **Aceptar**. Este mensaje tan solo sirve para confirmar que desea que el proyecto se añada al archivo SVStoreEAR.

Puesto que está construyendo Project501 como un duplicado del proyecto completo SV000501, ahora tendrá que copiar algunos de los archivos de imagen de SV000501 en Project501, para que Project501 se visualice correctamente. (Sin embargo, si lo desea, puede añadir sus propias imágenes para la interfaz). Para copiar las imágenes:

1. Pase a la perspectiva Web.
2. En la vista Navegador, expanda **SV000501 > Web Content > images**.
3. (Opcional) Expanda **generated** para tomar nota de todas las imágenes que importa al proyecto.
4. Pulse **generated** con el botón derecho del ratón y seleccione **Copiar**.
5. Desplácese hacia arriba en la vista Navegador hasta que pueda volver a ver **Project501**.
6. Expanda **Project501 > Web Content**.
7. Pulse la carpeta **images** con el botón derecho del ratón y seleccione **Pegar** en el menú emergente.
8. Ahora ya existe la carpeta **generated** para Project501. Expanda **generated** para tomar nota de las imágenes que añadió.

---

## Convertir el fuente DDS

Ahora que ya ha creado el proyecto, podrá convertir los archivos de pantalla DDS en archivos JSP para la página Web. Cuando convierte los archivos de pantalla DDS, la herramienta WebFacing genera archivos JSP y archivos XML que sustituyen el código DDS y hacen posible el acceso Web. Los archivos generados contienen los datos del formato de registro (o bien controlan su aspecto), visualizan la versión Web de la pantalla, solicitan datos y manejan los errores de entrada. El asistente también genera una página de presentación de la aplicación para lanzar la versión del programa habilitada para la Web.

Sin embargo, primero tendrá que modificar el código fuente para que haga referencia al servidor iSeries correcto. Concretamente, tendrá que realizar una modificación en los archivos de pantalla DDS para que funcione el enlace de la aplicación. Deberá cambiar el nombre por omisión, que es SV000501, por el nombre de su proyecto, Project501:

1. Pase de nuevo a la perspectiva WebFacing y abra la vista Proyectos WebFacing, si no está abierta por omisión.
2. Expanda **Project501 > DDS**.
3. Efectúe una doble pulsación en la segunda entrada, `<servidor_iSeries>RETAILSTOR/QDDSSRC(QUERY)`

4. Desplácese hacia abajo hasta más o menos la línea 16, donde verá la línea:  
A\*%WB 12 FLD 1 next ('/SV000501/DetailPage.do?PRODNO=&{PRODNO}')
5. Suprima SV000501 y escriba Project501.
6. Guarde y cierre el archivo.

**Nota:** Si en un momento dado desea ejecutar de nuevo el proyecto SV000501, debe cambiar el valor Project501 para que de nuevo sea SV000501, ya que este archivo existe en el servidor iSeries y los dos proyectos acceden a él.

Para convertir el fuente DDS:

1. En la perspectiva WebFacing, pulse la pestaña **Proyectos WebFacing** para pasar a la vista Proyectos WebFacing, si aún no está abierta por omisión.
2. Expanda **Project501 > DDS**.
3. Seleccione *servidoriSeries > RETAILSTOR/QDDSSRC(ORDERDSP)* y *servidoriSeries > RETAILSTOR/QDDSSRC(QUERY)*; para ello, pulse la primera y, mientras mantiene pulsada la tecla Mayús, pulse la segunda.
4. Pulse el botón derecho del ratón y seleccione **Convertir** para dar comienzo a la conversión. Es posible que deba especificar su ID de usuario y su contraseña si en algún momento queda desconectado del servidor iSeries.

## Configurar el soporte UTF-8 en el entorno de trabajo — Sólo para los usuarios de WAS Versión 4.0

Las aplicaciones de la herramienta WebFacing permiten visualizar múltiples idiomas en una pantalla. Dado que los idiomas utilizan distintos juegos de caracteres, las corrientes de datos entre el navegador y WebSphere Application Server están codificadas en UTF-8. Para que la herramienta IBM WebFacing funcione como es debido, tendrá que configurar el soporte UTF-8 en el archivo de propiedades de la aplicación en el entorno de trabajo.

**Nota:** Este apartado tan solo atañe a los usuarios de WAS 4.0. En WAS Versión 5.0, esta tarea se realiza automáticamente.

Para configurar el soporte UTF-8:

1. Pase a la perspectiva Web (para pasar de una perspectiva a otra, puede pulsar los iconos agrupados en la parte izquierda de la pantalla).
2. Pulse la pestaña **Navegador de proyectos** para poder ver las estructuras de los proyectos.
3. Expanda **Project501** y pulse dos veces en **Descriptor de despliegue Web**.
4. Pulse la pestaña **Entorno**.
5. Pulse el botón **Añadir**.
- 6.
7. Pase a la perspectiva Servidores pulsando el correspondiente icono en la barra de tareas de la parte superior izquierda, o bien pulse **Perspectiva > Abrir > Otras > Servidor** en la barra de menús y después pulse **Aceptar**.
8. En la vista Navegador, expanda la carpeta **Servidores**.
9. Pulse dos veces en **defaultInstance.wsi** para abrir este archivo en el editor por omisión.
10. Pulse la pestaña **Entorno** y, después, el botón **Añadir**.
11. Aparece el valor editable "(Variable Nueva)". Suprima la serie que aparece por omisión y escriba: `client.encoding.override`.



12. En el campo **Valor**, escriba UTF-8.
13. Pulse el icono de Guardar o bien pulse **Archivo > Guardar descriptor de despliegue Web**.

## Configurar el soporte UTF-8 para WebSphere Application Server

(Opcional) Si desea desplegar su aplicación de iSeries en WebSphere Application Server, también tendrá que configurar el soporte UTF-8 en WebSphere Application Server, además de en el entorno de trabajo.

### Para configurar el soporte UTF-8 en WebSphere Application Server 4.0 Advanced Edition

1. Inicie la consola administrativa de WebSphere.
2. Expanda el icono de **Nodos** y, después, **Nombre de nodo > Servidores de aplicaciones > Default Server**.
3. Seleccione la pestaña **Valores de JVM** y pulse el botón **Valores avanzados de JVM** para abrir el diálogo **Valores avanzados de JVM**.
4. En el campo **Argumentos de línea de mandatos**, escriba:  
`-Dclient.encoding.override=UTF-8`
5. Pulse **Aceptar** y, después, **Aplicar** bajo la pestaña **Valores de JVM**.
6. Para que este cambio entre en vigor en las aplicaciones de WebSphere, detenga el servidor por omisión y vuelva a iniciarlo. Para detener el servidor, pulse **Default Server** con el botón derecho del ratón y seleccione **Detener**. Una vez realizado este proceso, pulse **Default Server** con el botón derecho del ratón y seleccione **Iniciar**.

### Para configurar el soporte UTF-8 en WebSphere Application Server 4.0 Advanced Single Server Edition

1. Inicie la consola administrativa de WebSphere.
2. En la consola administrativa basada en navegador, expanda el icono de **Nodos** y, después, **Nombre de nodo > Servidores de aplicaciones > Default Server > Definición de proceso > Valores de JVM**.
3. Desplácese hasta la sección de valores avanzados de la página **Valores de JVM** y pulse el enlace **Propiedades del sistema**. Se visualiza la página **Propiedades del sistema**.
4. Pulse **Nueva** para añadir una nueva propiedad del sistema.
5. En el campo **Nombre**, escriba `client.encoding.override`.
6. En el campo **Valor**, escriba UTF-8.
7. Pulse **Aceptar**. Si recibe un mensaje que indica que **hay que guardar la configuración**, con un enlace en la parte superior de la página **Valores de JVM**, pulse el enlace para ir a la página **Guardar configuración**. Seleccione **Guardar** y luego pulse **Aceptar**.
8. Para que este cambio entre en vigor en las aplicaciones de WebSphere, detenga el servidor de aplicaciones y vuelva a iniciarlo. La manera de detener e iniciar el servidor de aplicaciones puede variar en función de la plataforma en la que tenga instalado WebSphere Application Server. En la documentación de WebSphere Application Server correspondiente a su plataforma hallará información sobre cómo detener e iniciar el servidor de aplicaciones.

---

## Crear una hoja de estilo

Si desea integrar páginas adicionales con una hoja de estilo en cascada (CSS), tendrá que personalizar ya sea el estilo del proyecto WebFacing o bien la hoja de estilo en cascada (CSS) para que tengan un aspecto equivalente. Cuando haya terminado de personalizar la hoja de estilo, puede utilizar el asistente Interacción Web para crear una página Web detallada, mediante la hoja de estilo, para visualizar información sobre los artículos de la tienda, como el precio y el color. En lo que al módulo se refiere, incorporará la hoja de estilo `DetailPageResults.jsp` del proyecto SV000501. Sin embargo, para consulta futura, puede personalizar manualmente el archivo CSS (tal como se indica en el próximo apartado).

### (Opcional) Personalizar manualmente la hoja de estilo en cascada (CSS)

Tal como se ha indicado anteriormente, en vez de incorporar la hoja de estilo `DetailPageResults.jsp` del proyecto SV000501 para `Project501`, puede personalizar sus propias hojas de estilo:

1. Pase a la vista Navegador de proyectos y expanda **Project501 > Web Content > styles > apparea**.
2. Pulse dos veces en **apparea.css** para abrir el archivo en el diseñador de CSS. Esta hoja de estilo controla el aspecto del área de su aplicación.
3. Modifique el archivo y guárdelo.
4. De nuevo en la vista Navegador de proyectos, navegue hasta **Project501 > Web Content > styles > chrome**.
5. Pulse dos veces en **avenue.css** para abrir el archivo en el diseñador de CSS. Esta hoja de estilo controla el aspecto de toda la página.
6. Modifique el archivo y guárdelo.

---

## Ampliar y mejorar el proyecto WebFacing con una interacción Web

En este apartado mejorará el proyecto WebFacing. Cuando se visualiza la lista de artículos, le interesará pulsar la imagen de un artículo para averiguar sus detalles. Para ello, utilizará el asistente Interacción Web para invocar el programa RPG con el fin de recuperar los detalles del artículo y visualizarlos en otra página Web. Llevará a cabo estos pasos:

- definir la información del servidor
- copiar la hoja de estilo correcta
- crear la interacción
- añadir programas y parámetros a la interacción
- cambiar la utilización de los parámetros

Primero tendrá que definir la información del servidor:

1. Pase a la perspectiva Web.
2. En la vista Navegador, pulse **Project501** con el botón derecho del ratón y seleccione **Especificar configuración de tiempo de ejecución de herramientas Web de iSeries**.
3. Escriba el nombre del servidor iSeries, su ID de usuario y su contraseña.
4. Escriba `Retailstor` en el campo **Biblioteca** y pulse **Añadir**.
5. Pulse **Finalizar**. Pulse nuevamente **Finalizar**, si es necesario.

Antes de crear la interacción Web, tendrá que copiar los estilos correctos y la hoja de estilo para que Project501 se visualice con el debido formato de archivo JSP. (No tendría que realizar esta tarea si hubiese llevado a cabo el ejercicio del apartado “(Opcional) Personalizar manualmente la hoja de estilo en cascada (CSS)” en la página 36). Primero copiará la carpeta de estilos y, después, el archivo `DetailPageResults.jsp`:

1. En la vista Navegador, expanda **SV000501 > Web Content**.
2. Pulse **Styles** con el botón derecho del ratón y seleccione **Copiar**.
3. Desplácese hacia arriba en la vista Navegador hasta que vuelva a ver **Project501**.
4. Expanda el proyecto, pulse **Web Content** con el botón derecho del ratón y seleccione **Pegar** en el menú emergente. Pulse **Sí** para sobrescribir los estilos existentes.
5. Desplácese hacia abajo hasta **SV000501 > Web Content**.
6. Pulse **DetailPageResults.jsp** con el botón derecho del ratón y seleccione **Copiar**.
7. Desplácese hacia arriba hasta que vuelva a ver su proyecto **Project501**.
8. Expanda el proyecto, pulse **Web Content** con el botón derecho del ratón y seleccione **Pegar** en el menú emergente.

Observe los nuevos elementos que se han añadido bajo la carpeta Web Content de Project501.

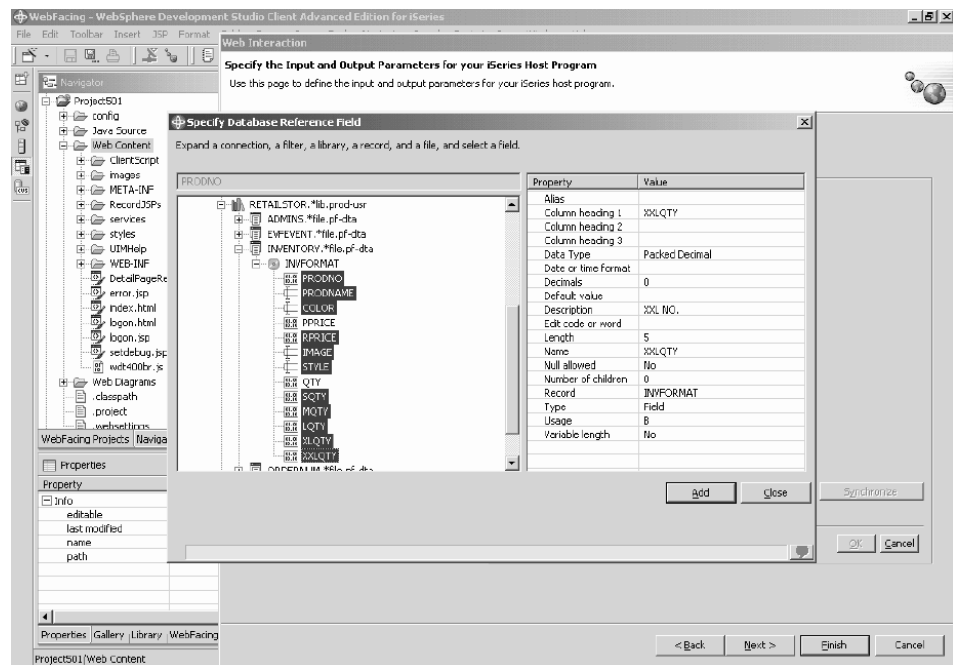
Ahora ya puede crear la interacción Web:

1. Pulse **Archivo > Nuevo > Otros**.
2. En el diálogo **Nuevo**, pulse **Web** en la parte izquierda, y luego pulse **Interacción Web** en la parte de la derecha. Pulse **Siguiente**.
3. En el campo **Nombre de interacción Web**, escriba `DetailPage` y pulse **Siguiente**.
4. Marque el botón de selección **Generar JSP de entrada**.
5. Marque el botón de selección **Utilizar páginas de salida** (si es que todavía no está marcado) y pulse **Añadir**.
6. En el diálogo **JSP de salida**, expanda **Web Content**, seleccione **DetailPageResults.jsp**, que es su página de salida preformateada, y pulse **Aceptar**.
7. Pulse **Siguiente**.

Ahora ya puede añadir programas y parámetros a la interacción. Tendrá que añadir 11 parámetros al mismo programa. En vez de ir añadiendo manualmente cada parámetro con sus valores individuales, puede añadirlos con un procedimiento algo más rápido:

1. Seleccione **Utilizar un programa ILE de iSeries**, si es que todavía no está seleccionado.
2. Pulse **Añadir programa**.
3. En el campo **Alias de programa**, escriba `DetailPage`.
4. Para el campo **Objeto de programa**, pulse **Examinar**.
  - a. Expanda `servidoriSeries > *LIBL > RETAILSTOR`.
  - b. Pulse `DETAILPAGE.*pgm.rpgle` (que es el primer archivo bajo RETAILSTOR) y pulse **Aceptar**.

5. De nuevo en el asistente Interacción Web, pulse **Aceptar** (hacia la parte inferior de la derecha).  
Fíjese en que **DetailPage** se añade en la parte izquierda del asistente bajo **Definiciones de llamada a programa**.
6. En la sección **Definiciones de llamada a programa**, en el panel izquierdo del asistente, pulse **DetailPage** para que quede seleccionado.
7. Pulse **Añadir parámetro**.
8. Junto a **Especificar campo de referencia a base de datos**, hacia la parte inferior del asistente, pulse **Especificar**.
9. Expanda *servidoriSeries* > \*LIBL > RETAILSTOR > INVENTORY.\*file.pf-dta > INVFORMAT para visualizar una lista de 13 parámetros. Tendrá que añadir los 11 parámetros que se indican a continuación; para ello, pulse una vez en cada uno de ellos y seleccione **Añadir**, o bien, sin soltar la tecla Control, pulse cada uno de ellos y luego seleccione **Añadir**: PRODNO, PRODNAME, COLOR, RPRICE, IMAGE, STYLE, SQTY, MQTY, LQTY, XLQTY, XXLQTY. Lo que está haciendo en realidad es seleccionar todos los campos salvo PPRICE y QTY.

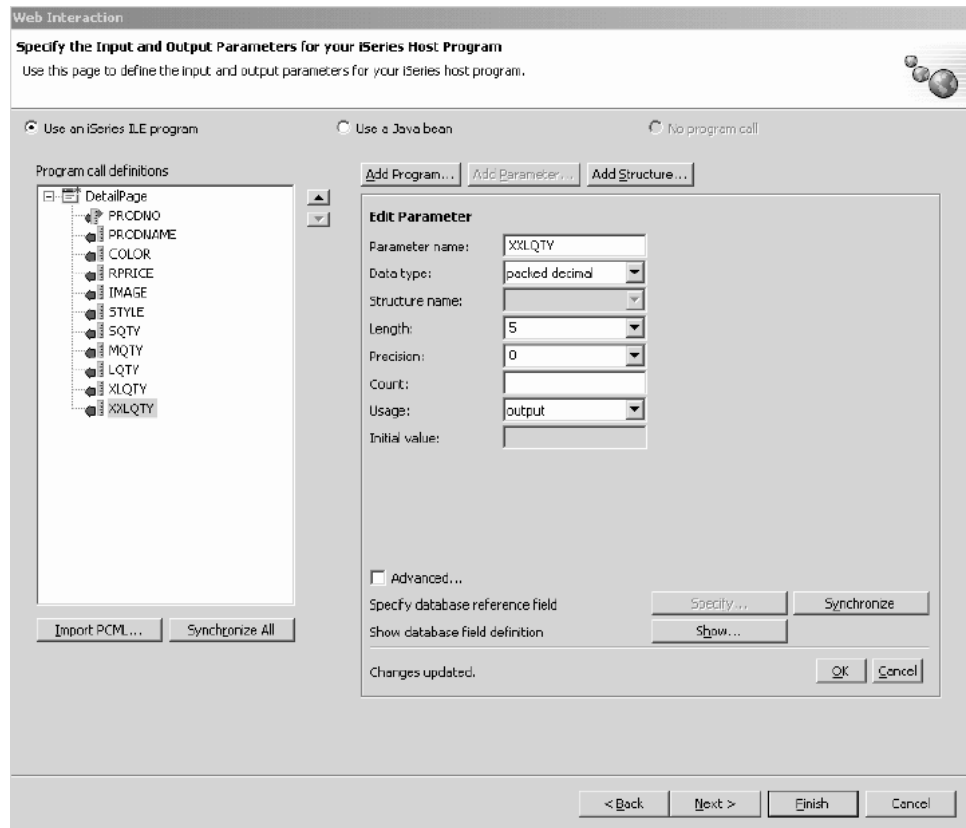


10. Pulse **Cerrar**.

Ahora tendrá que cambiar la utilización de 10 de los 11 parámetros para que sean de tipo "salida" (de todos los parámetros salvo PRODNO).

1. Todavía en el asistente Interacción Web, pulse **PRODNAME** para seleccionarlo.
2. En el cuadro combinado **Utilización**, haga que el valor seleccionado pase a ser **salida** y pulse **Aceptar**. Repita este paso (junto con el paso anterior) para todos los parámetros salvo el primero (PRODNO). Una vez que haya ajustado todos los parámetros, su área de trabajo debe parecerse a la de la siguiente figura. Observe que el icono situado a la izquierda de cada parámetro indica si este es

de tipo entrada, entrada y salida, o salida.



3. Pulse **Siguiente** para obtener la vista previa del **Formulario de entrada**.
4. Pulse **Finalizar** para crear la interacción Web.
5. Seleccione **Sí a todo** o **Aceptar** y, de nuevo, **Finalizar**, si recibe mensajes.

## Enlazar el proyecto con una interacción Web

Ahora que ya ha creado un archivo JSP de interacción Web que utiliza parámetros de entrada y salida, tendrá que personalizar el archivo JSP para que también funcione con el componente WebFacing. Deberá entrar código para crear desde la aplicación WebFacing un enlace que invoque esta interacción Web. Para ello, tendrá que añadir una función JavaScript en el archivo `webface.js` para que se pueda llamar al servlet `DetailPageServlet` con el parámetro `PRODNO` en una ventana nueva.

Para crear el enlace:

1. En la vista Navegador, expanda **Project501 > Web Content > ClientScript**.
2. Pulse dos veces en **webface.js** para abrir este archivo en un editor.
3. Desplácese hasta el final del archivo y escriba las siguientes líneas:

```
var mywindow
function next(app)
{
mywindow = window.open(app,"Details","RESIZABLE=YES, HEIGHT=700, WIDTH=800");
}
```
4. Pulse el icono de Guardar o bien pulse **Archivo > Guardar webface.js**.

Si desea crear una aplicación similar a SV000501 en el futuro, también tiene que cambiar un valor Web en su fuente DDS para habilitar la imagen que ha añadido, y habilitar asimismo el enlace de cerrar ventana para la función JavaScript.

El código RPG incluido en esta aplicación se ha modificado para que muestre el cambio, pero usted tendrá que realizar el cambio manualmente en las futuras aplicaciones. Además, tras cambiar los valores Web, deberá reconvertir el fuente DDS.

Puede comprobar el código para duplicar el resultado.

Para ver el fuente DDS:

1. Pase a la perspectiva WebFacing.
2. En la vista Proyectos WebFacing, expanda **Project501 > DDS**.
3. Pulse **<servidoriSeries > RETAILSTOR/QDDSSRC(QUERY)** con el botón derecho del ratón y seleccione **Abrir con > CODE Designer**.
4. Una vez abierto el Diseñador CODE, expanda **SCREEN1 > ITEMSUB**.
5. Pulse **IMAGESRC** para seleccionarlo.
6. Pulse la pestaña **Fuente**.
7. Pulse la pestaña **Valores Web**, en la parte inferior derecha de la ventana.

**Nota:** Compruebe las propiedades de los valores Web; por ejemplo, la anchura en píxels y el nombre del archivo. En el futuro, tendrá que hacer los mismos cambios en el fuente DDS y luego reconvertir el fuente.

8. Fíjese en las siguientes líneas del fuente:

```
A   PRODNO  R   0 5 6
A   PRODNAME R   0 5 16
A   IMAGESRC 19A 0 5 33
A*%WB 13 FLD 100|100|&{IMAGESRC}
A*%WB 12 FLD 1 javascript:next
      ('/Project501/DetailPageServlet?PRODNO=&{PRODNO}')
```

Observe en especial que el proyecto Project501 está especificado en la última línea. Tal como se ha mencionado anteriormente, si desea ejecutar de nuevo la aplicación SV000501, tendrá que cambiar el valor Project501 para que sea de nuevo SV000501.

9. Guarde el archivo pulsando el icono de guardar o bien pulsando **Archivo > Guardar** en la barra de menús.
10. Cierre el archivo y, después, el Diseñador CODE.

Si desea más información sobre cómo trabajar con el fuente DDS, pase a la perspectiva Ayuda del entorno de trabajo y vea la documentación de la herramienta IBM WebFacing.

---

## Publicar los archivos y reiniciar el servidor

En este apartado publicará sus archivos y reiniciará el servidor para que todos los cambios entren en vigor en la aplicación Project501.

Sin embargo, antes de reiniciar el servidor, debe verificar que Project501.war se ha añadido al archivo SVStoreEAR.

Para verificar la configuración:

1. En la vista Navegador, expanda la **carpeta Scenario**.
2. Pulse dos veces en **Scenario server.wsi**. Podría ver un diálogo con la pregunta de si desea que el editor arregle las entradas del proyecto automáticamente. Pulse **Sí**. Si no ve el diálogo, es que no hace falta llevar a cabo ninguna acción. De todos modos, el archivo se modifica automáticamente en el momento de abrirse.
3. Guarde y cierre el archivo.

Para reiniciar el servidor:

1. Pase a la perspectiva Servidores.
2. En la vista Configuración de servidor, hacia la parte inferior izquierda, expanda **Servidores** y pulse dos veces en **Servidor Scenario**.
3. Pulse la pestaña **Servidores**, en la parte inferior derecha de la pantalla, para ver el estado del servidor en la vista Servidores.
4. Pulse **Servidor Scenario** con el botón derecho del ratón y seleccione **Publicar**. Pulse **Aceptar** una vez llevada a cabo la publicación.
5. En la vista Servidores, pulse su servidor con el botón derecho del ratón y seleccione **Iniciar** o **Reiniciar** (según la opción que esté habilitada). Si se le pregunta acerca de los archivos no guardados, cancele el diálogo, guarde todos los archivos abiertos, y vuelva a intentarlo.
6. Compruebe la información de anotaciones de la consola (se abre automáticamente). Sabrá que el servidor se ha iniciado cuando vea la siguiente línea en la parte inferior del archivo de anotaciones: Servidor *nombre\_servidor* abierto para e-business.

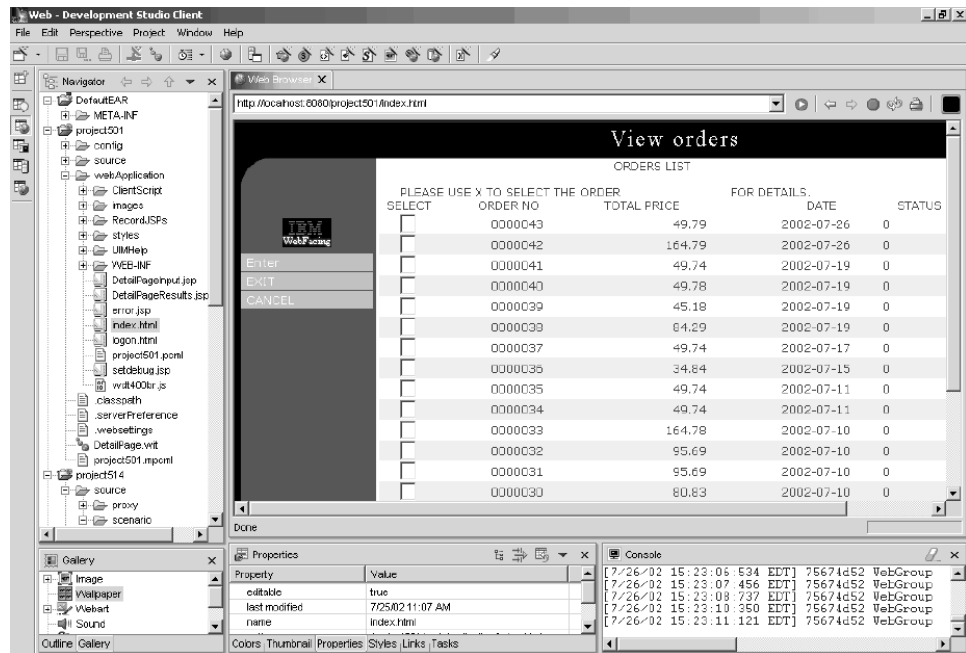
---

## Probar la interfaz

Ahora ya ha llevado a cabo los pasos necesarios para crear una interfaz que permite ver los pedidos y ver el inventario. Para probar la interfaz:

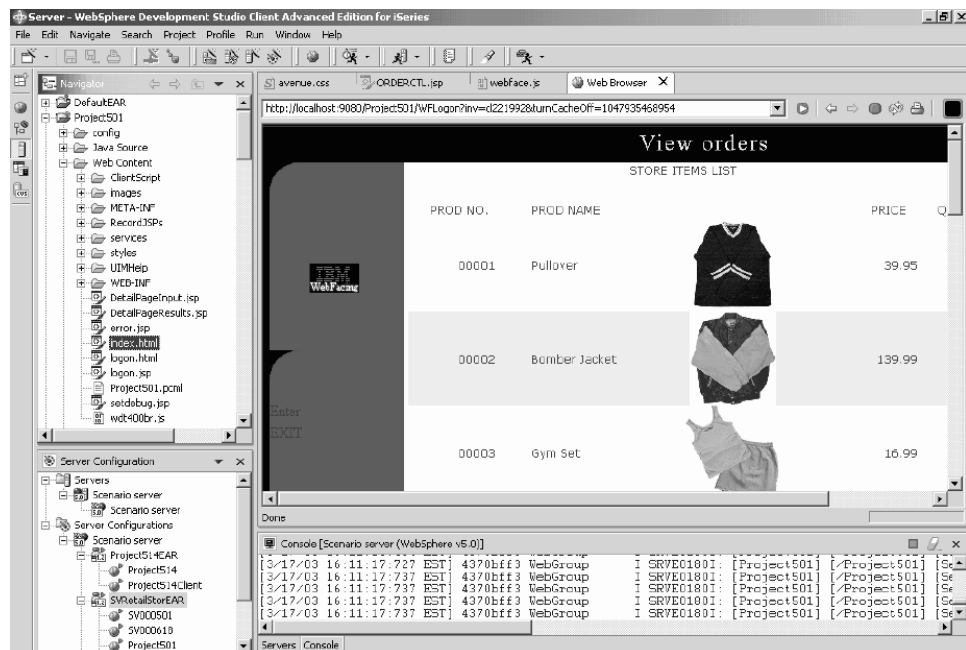
1. Pase a la perspectiva Web.
2. Expanda **Project501 > Web Content**.
3. Pulse **index.html** con el botón derecho del ratón y seleccione **Ejecutar en servidor**.
4. Se abre el diálogo **Selección de servidor**. Verifique que el **Servidor Scenario** está seleccionado por omisión y pulse **Finalizar**.
5. Pulse **Ver pedidos – Lanzar en la ventana principal del navegador**, para invocar la aplicación del administrador. Tras iniciar sesión con su ID de usuario y su contraseña de iSeries (en el servidor iSeries que utilizó mientras

desarrollaba este proyecto), se le dirigirá a la siguiente página:



Intente escribir una X junto a uno de los números de producto para ver los detalles de ese producto.

6. Pulse la flecha atrás para acceder a la página index.html, y pulse **Ver inventario – Lanzar en la ventana principal del navegador** para invocar la aplicación del cliente; se le dirigirá a la siguiente página (después de que haya iniciado sesión en el servidor iSeries con su ID de usuario y su contraseña):



Intente pulsar una imagen para visualizar los detalles de un artículo concreto.

Ya ha llegado al final del módulo. Ha creado una interfaz de usuario para ver el inventario y formular pedidos de artículos.



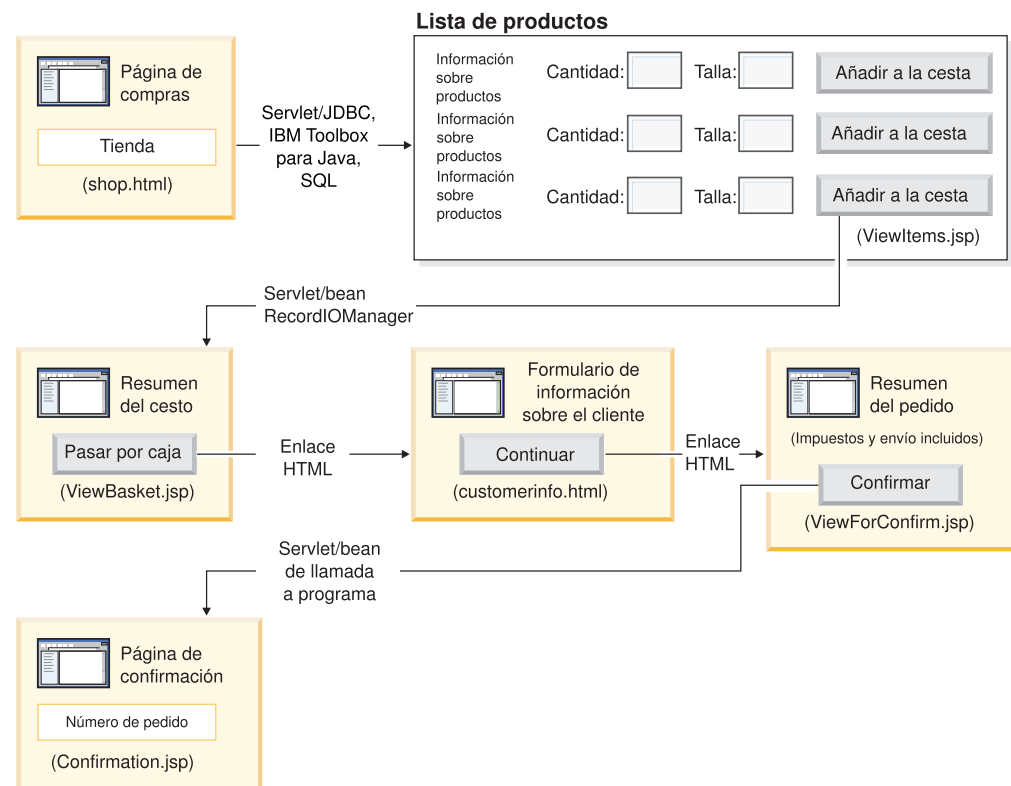
---

# Capítulo 6. Módulo 1 avanzado: Crear código HTML, servlets y archivos JSP que formulan pedidos de cliente en un servidor iSeries (SV001585)

---

## Introducción

Este proyecto está dirigido a los usuarios que conocen bien la programación Java y saben algo sobre la gestión de datos de iSeries y sobre RPG. El proyecto muestra cómo trabajar con las clases de acceso a datos de IBM Toolbox para iSeries, con el bean RecordIOManager y con el bean de llamada a programa para crear código HTML, servlets y archivos JSP que formulen pedidos de cliente en un servidor iSeries. Como usuario, su punto de partida será la página de compras, que le permite ver los productos disponibles y añadir artículos a la cesta de la compra. Cuando esté de acuerdo con todos los artículos seleccionados (en el resumen de la cesta de compra), pulsará un botón para pasar por caja y se le dirigirá a un formulario de información sobre el cliente. Una vez cumplimentado el formulario, el proyecto devuelve un resumen del pedido con un botón de confirmación; pasará a una página de confirmación que visualiza su pedido y su número de pedido.



---

## Resumen de los pasos de alto nivel

Como este es un módulo avanzado, las instrucciones no le orientan paso a paso en el proceso de crear el proyecto, sino que indican los pasos de desarrollo específicos del iSeries para crear dicho proyecto. Los pasos de alto nivel son:

1. Escribir una página Web de compras HTML.

2. Escribir un servlet que puebla un bean Java (mediante JDBC y SQL) con los artículos disponibles que el cliente puede comprar.
3. Escribir un archivo JSP que permita a los clientes ver los artículos a la venta, entrar la cantidad y la talla del artículo que desean, y seleccionar el artículo pulsando un botón **Añadir a la cesta**.
4. Mediante el bean RecordIOManager, escribir un servlet que se invoca al pulsar el botón **Añadir a la cesta** y que actualiza la base de datos INVENTORY de iSeries, restando la cantidad y la talla requerida del artículo, y añadiendo esta selección a un bean Java que se llama "cesta". Luego, el servlet redirige la respuesta al archivo ViewBasket.jsp, si la operación es satisfactoria. Si no lo es, el servlet visualiza una página de error.
5. Escribir un formulario HTML para que el cliente entre información personal.
6. Escribir un archivo JSP que sea una página de confirmación de compra y que visualice el contenido de la cesta, los impuestos y los costes de envío y de manejo. En la página también debe haber un botón de confirmación para el cliente.
7. Escribir un servlet al que se llama cuando el cliente pulsa el botón de confirmación, y que utiliza un bean Java creado con el asistente de llamada a programa de iSeries. Uno de los métodos del bean Java llama a un programa RPG para crear en la base de datos ORDERS del servidor iSeries un pedido nuevo que se corresponde con el contenido de la cesta del cliente. Luego el servlet devuelve un número de pedido, coloca el bean Java en la sesión de la aplicación Web y carga un archivo JSP de confirmación de pedido que contiene el número del pedido.

---

## Antes de empezar

Solo podrá llevar a cabo los ejercicios si se cumplen los siguientes prerequisites. En el tema Capítulo 3, "Ejecutar el caso práctico", en la página 9 hallará más detalles sobre los prerequisites.

- Tiene acceso TCP/IP a un servidor iSeries.
- Ha iniciado los servidores iSeries con el mandato STRTCPSVR \*ALL.
- Ha iniciado el servidor WebFacing con el mandato STRTCPSVR \*WEBFACING.
- Ha restaurado las bibliotecas WHOLESale y RETAILSTOR en el servidor iSeries.
- Las bibliotecas WHOLESale, RETAILSTOR y QGPL figuran en la lista de bibliotecas.
- Ha llevado a cabo todas las tareas del Capítulo 3, "Ejecutar el caso práctico", en la página 9 (salvo las tareas de WebSphere Application Server opcionales, que no son necesarias para probar la aplicación en el entorno de trabajo).

---

## Crear la página Web, los servlets y los archivos JSP

Para construir los componentes del proyecto SV001585:

1. Cree un proyecto Web.
2. Con Page Designer, escriba una página shop.html que contenga un enlace que invoca un servlet GetItems.
3. Importe el archivo jt400.jar para las clases de iSeries Toolbox para Java a la carpeta *lib* del proyecto Web. Este archivo jar está en `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, siendo *x* el directorio en el que ha instalado Development Studio Client.

**Nota:** Consulte los archivos `GetItems.java` y `ViewItems.jsp` del proyecto SV001585 para ver los componentes JDBC y SQL principales relacionados con iSeries Toolbox para Java, para el servlet y el archivo JSP. En la vista Navegador de la perspectiva Web, para acceder a `GetItems.java`, expanda **SV001585 > source**, y para acceder a `ViewItems.jsp`, expanda **SV001585 > webApplication**.

4. Escriba un servlet `GetItems` que utilice los componentes JDBC y SQL de iSeries Toolbox para Java para recuperar los artículos de ropa de la base de datos INVENTORY de iSeries y que:
  - a. Coloque en la sesión el bean `ResultSet` que contiene el resultado de la consulta SQL
  - b. Redirija la petición al archivo `ViewItems.jsp`

Ejemplo de código para `GetItems.jsp`:

```
public void init() {
    .
    .
    .
    // Cargar el controlador JDBC de IBM Toolbox para Java.
    DriverManager.registerDriver(new com.ibm.as400.access.AS400JDBCdriver());
    // Observe que hemos recuperado el nombre de as400, el ID de usuario y la contraseña
    // a partir del archivo web.xml mediante un analizador xml.
    as400conn =
    DriverManager.getConnection(
        "jdbc:as400://" + as400 + ";naming=sql;errors=full",
        userid,
        password);

    dmd = as400conn.getMetaData();
    .
    .
    .
}

public void service(HttpServletRequest request, HttpServletResponse response){
    .
    .
    .
    Statement select =
    as400conn.createStatement(
        ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);
    ResultSet rs =
    select.executeQuery(
        "SELECT PRODNO, PRODNAME, RPRICE, IMAGE FROM "
        + retailLibrary
        + dmd.getCatalogSeparator()
        + inventoryFile);

    HttpSession session = request.getSession(true);
    session.setAttribute("resultset", rs);

    response.sendRedirect("/ViewItems.jsp");

    .
    .
    .
}
```

5. Escriba un archivo JSP `ViewItems` que recupere los artículos de ropa a partir del bean `ResultSet` obtenido en el paso anterior, para visualizar dichos artículos en formato de tabla. El archivo JSP también debe incluir, para cada artículo, un formulario que permita seleccionar la talla y la cantidad y luego

añadir el artículo a la cesta de la compra. Para escribir el archivo JSP, puede utilizar Page Designer en las herramientas de desarrollo Web para iSeries. Concretamente, puede dibujar la página en la vista Diseño, y añadir el correspondiente código en la vista Fuente. Ejemplo de código para ViewItems.jsp:

```

<!--Obtener el objeto ResultSet a partir de la sesión--><%
    int columnCount = 0;
    ResultSet rs = (ResultSet)session.getAttribute("resultset");
    if(rs !=null)
%>
<%
{
    rs.beforeFirst();
        ResultSetMetaData rsmd = rs.getMetaData ();
        columnCount = rsmd.getColumnCount ();

%>
<TABLE border="1">
<TBODY>
<TR>
    <TD>ID de producto</TD>
    <TD>Nombre</TD>
    <TD width="551">Precio</TD>
    <TD colspan="2"></TD>
</TR>
<%while (rs.next ()) {
<TR>
<!--Crear un formulario para esta fila (o este artículo)-->
<FORM name="myform" action="/SV001585/AddtoBasket" onsubmit="return errorChecking(this);">

<!--Obtener los datos de cada columna a partir de esta fila del objeto ResultSet-->
<!--Procesar datos es un método definido por usuario para modificar los datos para
visualizar, si es necesario-->
        <%
            for (int i = 1; i <= columnCount; ++i){
                String value = rs.getString(i);
                if (rs.wasNull ())
                    value = "<null>";
                else{
                    if(i==1)
                        prodID=value;
                    value = processData(i,value);

                }
            }
        <%
        <TD><%=value%></TD>
        <%
    }
%>

<!--Crear un campo de entrada para la cantidad y un menú desplegable para la talla-->
<!--Observe que estamos utilizando el ID del producto como nombre del campo-->
    <TD width="290">Cantidad
    <INPUT size="5" type="text" name='<%=prodID+"Q"%>' ><BR>
    Talla <SELECT name='<%=prodID+"S"%>'>
        <OPTION value="s" selected>Pequeña</OPTION>
        <OPTION value="m" selected>Mediana</OPTION>
        <OPTION value="l" selected>Grande</OPTION>
        <OPTION value="XL" selected>Muy grande</OPTION>
        <OPTION value="XXL" selected>Super grande</OPTION>
    </SELECT>
</TD>
<TD><INPUT type="image" name="submit" src="images/Add_to_basket.gif"></TD>
</FORM>
</TR>
<%

```

```

}
%>
</TBODY>
</TABLE>
<%
}

```

6. Importe el archivo iseriesut.jar de las herramientas de desarrollo Java para iSeries a la carpeta *lib* para su proyecto Web. Este archivo JAR está en `x:\wdsc\wssd\plugins\com.ibm.etools.iseries.toolbox\runtime`, siendo *x* el directorio en el que ha instalado el producto. Consulte los archivos `AddtoBasket.java` y `ViewBasket.jsp` del proyecto SV001585 para ver la implementación. En la vista Navegador de la perspectiva Web, para acceder a `AddtoBasket.java`, expanda **SV001585 > source**, y para acceder a `ViewBasket.jsp`, expanda **SV001585 > webApplication**.
7. Utilice el bean `RecordIOManager` de las herramientas de desarrollo Java para iSeries para escribir un servlet `AddtoBasket` al que debe llamar el botón **Añadir a la cesta** y que actualiza la base de datos `INVENTORY` de iSeries restando la cantidad solicitada por el cliente y añadiendo los artículos a un bean Java que representa la cesta de la compra en la sesión. Ejemplo de código de `AddtoBasket.jsp`:

```

public class AddtoBasket extends HttpServlet {

    //Clase interna de AddtoBasket
    public class MyRecordIOManager extends RecordIOManager {
        .
        .
        .
        public MyRecordIOManager(
            String hostInfo1,
            String hostInfo2,
            String hostInfo3,
            String file,
            String lib)throws Exception{
            super(hostInfo1, hostInfo2,hostInfo3,file,lib);
            setFileType(RecordIOManager.FILEACCESS_KEYED);
            setCommitLockLevel(RecordIOManager.COMMITLOCKLEVEL_ALL);
            //el diario tiene el mismo nombre que el archivo de base de datos
            setJournal(file);
            //el diario está en la misma biblioteca que el archivo de base de datos
            setJournalLibrary(lib);
        }
        .
        .
        .
        public synchronized String updateDBFile(
            String id,
            String size,
            String quantity
        ) {
            .
            .
            .
            //Abrir el archivo
            try {
                if (openFile()) {
                    record = readRecord(key);
                    quantityAvailable = ((BigDecimal)
                        record.getValueAt(0,sizeColumn)).intValue();
                    totalQuantityAvailable = ((BigDecimal)
                        record.getValueAt(0, 8)).intValue();
                    if (quantityRequested <= quantityAvailable) {
                        newQuantity =
                            new BigDecimal(quantityAvailable - quantityRequested);

```

```

        totalNewQuantity =
            new BigDecimal
                (totalQuantityAvailable - quantityRequested);
        record.setValueAt(newQuantity, 0, sizeColumn);
        record.setValueAt(totalNewQuantity, 0, 8);
        // Observe que actualizamos el registro, pero no comprometemos
        // si el cliente decide
        // vaciar la cesta, en cuyo caso
        // llamamos al método rollBack para restrotraer
        updateRecord(record);
        status = success;
    } else {
        status = notEnough;
    }
} else
    status = accessError;
} catch (Exception e) {
    e.printStackTrace();
    status = accessError;
}

//Cerrar el archivo y añadir
try {
    closeFile();
} catch (Exception e) {
    //En caso de error, retrotraer (rollback)
    try {
        rollback();
    } catch (Exception e1) {
        e1.printStackTrace();
    }
}
status = accessError;
}

return status;
}
}

//Método init del servlet AddtoBasket
public void init() {
    hostInfo = GetItems.getHostInfo();
}

public void doGet(HttpServletRequest req, HttpServletResponse res) {
    .
    .
    .
    Basket basket = (Basket) session.getAttribute("basket");
    MyRecordIOManager recIO =
        (MyRecordIOManager) session.getAttribute("recIO");
    if (basket == null) {
        basket = new Basket();
        session.setAttribute("basket", basket);
    }

    if(recIO == null){
        if (recIO == null) {
            try {
                recIO =
                    new MyRecordIOManager(
                        hostInfo[0],
                        hostInfo[1],
                        hostInfo[2],
                        GetItems.getInventoryFile(),
                        GetItems.getRetailLibrary());
            } catch (Exception e) {

```

```

        try {
            res.sendRedirect("errorPage.html");
            return;
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
}
id = req.getParameter("id");
size = req.getParameter(id + "S");
quantity = req.getParameter(id + "Q");

status = recIO.updateDBFile(id, size, quantity);
session.setAttribute("recIO", recIO);

if (status.equals("SUCCESS")) {
    basket.addItem(id, quantity, size);
    try {
        res.sendRedirect("ViewBasket.jsp");
        return;
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    if (status.equals("NOT_ENOUGH")) {
        try {
            res.sendRedirect("insufficient.html");
            return;
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else
        if (status.equals("ACCESS_ERROR")) {
            try {
                res.sendRedirect("errorPage.html");
                return;
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
}
}

```

8. Escriba un archivo JSP ViewBasket que visualice el contenido de la cesta de la compra.
9. Escriba un formulario customerinfo.html para entrar la información de pago, con un botón **Continuar** que invoque el archivo ViewForConfirm.jsp.
10. Escriba un archivo ViewForConfirm.jsp que muestre el contenido de toda la cesta más el saldo total. Puede desarrollar el archivo ViewForConfirm.jsp de la misma manera que el archivo ViewBasket.jsp, con la adición de un gasto de envío calculado y de un botón **Confirmar** que coloque un pedido en la base de datos ORDERS de iSeries.
11. Utilice el asistente de llamada a programa de iSeries, de las herramientas de desarrollo Java para iSeries, para crear un bean PLACEORD.java que acceda al programa de servicio PLACEODR de la biblioteca RETAILSTOR. El asistente crea beans que se utilizarán en las aplicaciones Java o en el asistente Servicios Web para acceder al programa ILE de iSeries.
  - a. Para abrir el asistente, pulse SV001585 con el botón derecho del ratón en la vista Navegador y seleccione **Nuevo > Otros**.
  - b. En la ventana **Nuevo**, pulse **iSeries > Java > Bean de llamada a programa**.

- c. En el asistente de llamada a programa, entrará información sobre el programa ILE de iSeries, como el nombre de programa, la biblioteca, el tipo de programa y los parámetros de entrada y de salida.
- d. En la última ventana del asistente, tendrá la opción de crear un bean para una aplicación Java, para un servicio Web o para ambos. En este proyecto, tan solo tiene que crear un bean para una aplicación Java.

**Nota:** El programa de servicio PLACEODR toma una matriz de estructuras y coloca cada elemento en un registro de una base de datos ORDERS, generando un número de pedido como salida para cada matriz.

12. Escriba un servlet PlaceOrder, invocado por el botón **Confirmar** del archivo ViewForConfirm.jsp.
  - El servlet utiliza el bean generado por el asistente de llamada a programa de iSeries para acceder a un servidor iSeries y colocar los pedidos en la base de datos ORDERS de la biblioteca RETAILSTOR.
  - Los pedidos son los artículos que hay en la cesta de la compra, enviados al programa ILE en forma de matriz de estructuras.
  - Cada estructura de esta matriz es un artículo de la cesta.
  - El programa de servicio RPG PLACEORD al que llama el bean devuelve el número de pedido como parámetro de salida y lo coloca en la sesión.

El siguiente segmento de código muestra cómo utiliza el servlet PlaceOrder el bean PLACEORD:

```

.
.
.
public void init() throws ServletException {
    hostInfo = GetItems.getHostInfo();
    super.init();

    try {
        /* Crear una instancia del bean PLACEORD creado
           por el asistente del bean de llamada a programa de iSeries */
        orderBean = new PLACEORD();
        orderBean.setConnectionData(hostInfo[0],
            hostInfo[1], hostInfo[2]);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

.
.
.

public void doPost(HttpServletRequest request,
    HttpServletResponse response) {

    ...

    PLACEORD.Orditems_Struct inputStruct = null;

    // Recuperar los artículos del pedido a partir de la cesta
    Basket basket =
    (Basket) request.getSession().getAttribute("basket");
    AddtoBasket.MyRecordIOManager recIO =
    (AddtoBasket.MyRecordIOManager)
        request.getSession().getAttribute("recIO");
    if (basket == null || basket.size() == 0
        || recIO == null) {
        try {

```



```

        response.sendRedirect("errorPage.html");
    } catch (IOException e) {
        e.printStackTrace();
    }
} else {
    items = basket.elements();
    // Establecer la matriz de elementos de tipo estructura
    while (items.hasMoreElements()) {
        item = (String[]) items.nextElement();
        inputStruct = orderBean.getOrdItemAr(j);
        inputStruct.setItemNo(new BigDecimal(item[0]));
        inputStruct.setQuantity(new BigDecimal(item[1]));
        inputStruct.setSizeOrd(item[2]);
        j = j + 1;
    }
    // Establecer el resto de los elementos de la matriz en valores ficticios
    for (int i = j - 1; i < 100; i++) {
        inputStruct = orderBean.getOrdItemAr(i);
        inputStruct.setItemNo(new BigDecimal(0));
        inputStruct.setQuantity(new BigDecimal(0));
        inputStruct.setSizeOrd("s");
    }

    // Establecer los otros dos parámetros de entrada del bean
    orderBean.setNumOfItems(new BigDecimal(j));
    orderBean.setBalance((BigDecimal)
        request.getSession().getAttribute("balance"));
    try {
        // Invocar el programa iSeries
        orderBean.invoke();

        // Recuperar el número de pedido a partir del bean PLACEORD
        orderNumber = (orderBean.getRetCode()).toString();
        request.getSession().setAttribute("orderNumber",
            orderNumber);
        basket.empty();
        // Comprometer este pedido ahora
        recIO.commit();
        response.sendRedirect("orderNumber.jsp");
        return;
    } catch (Exception e) {
        response.sendRedirect("errorPage.html");
        e.printStackTrace();
    }
}
}
}

```

13. Escriba un servlet OrderNumber que recupere el número del pedido y lo visualice para el cliente, junto con un mensaje de confirmación. En cambio, asegúrese de que se devuelve una página de error si el cliente no ha añadido ningún artículo a su cesta de la compra.



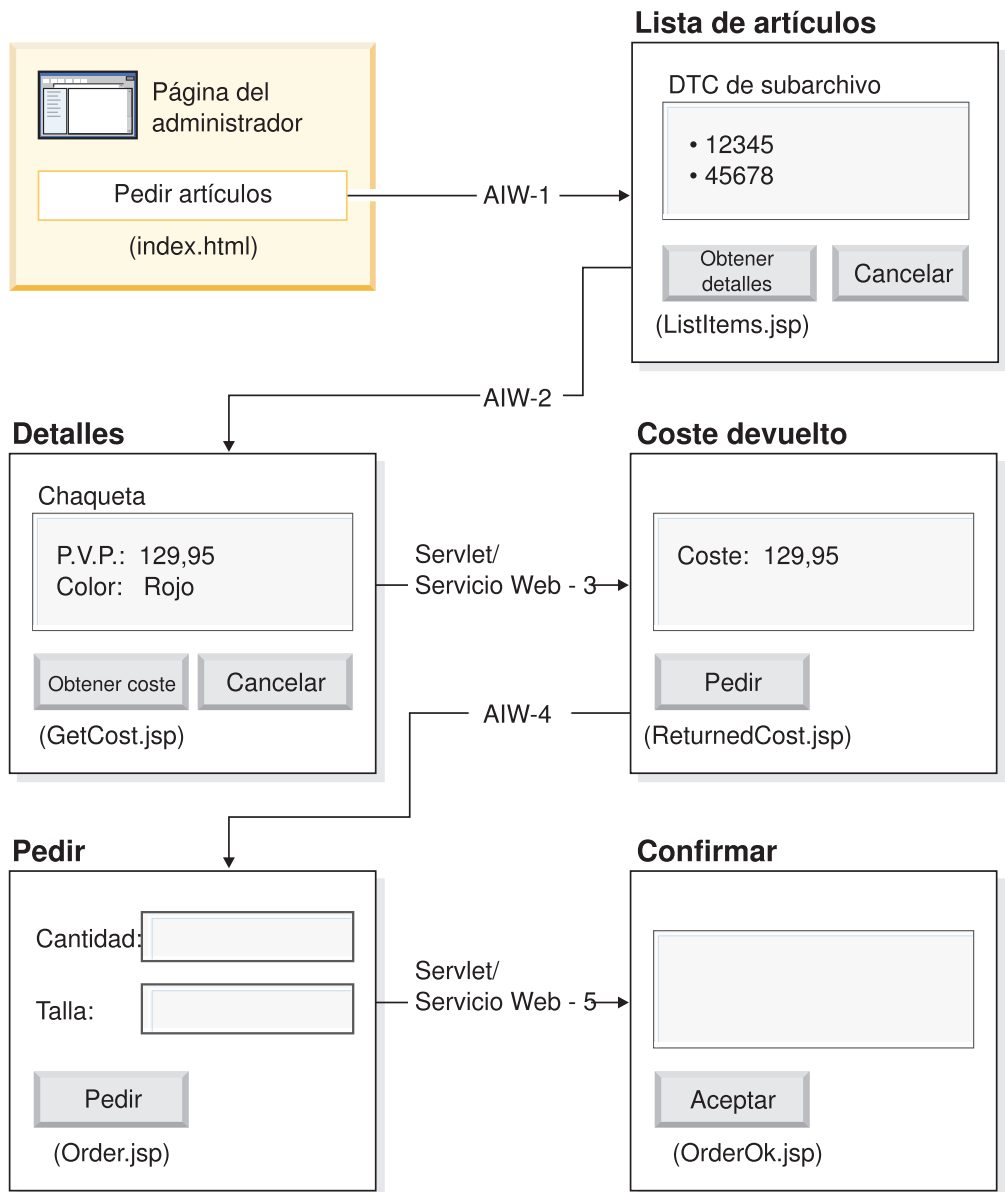
---

## **Capítulo 7. Módulo 2 avanzado: Crear el proyecto Web que utiliza los servicios Web SV000514 y SV001586 (SV000618)**

---

### **Introducción**

Este proyecto demuestra cómo puede utilizar sus conocimientos en programación RPG para crear clientes Web para programas RPG y servicios Web de iSeries. En este proyecto, desempeñará el rol de administrador, recorriendo una serie de páginas Web para determinar la cantidad que hay en el inventario y formular al suministrador al por mayor pedidos de inventario adicionales para su tienda al por menor. Entrará el número del artículo, verá los detalles del artículo, solicitará las cantidades y la talla, y aceptará la confirmación.



(AIW = Asistente de interacción Web)

Como este es un módulo avanzado, las instrucciones no orientan paso a paso en el proceso de crear el proyecto, sino que indican los pasos de desarrollo para crear dicho proyecto. En este proyecto se emplean los siguientes componentes de Development Studio Client:

- Las herramientas de desarrollo Web para iSeries, para crear las páginas Web con Page Designer, incorporando la salida a partir del asistente Interacción Web y diversos componentes Web
- El Explorador de Sistemas Remotos, para crear un programa de servicio RPG TNLSTITM que devuelve información sobre el artículo
- El asistente Servicios Web, para generar código de proxy de servlet, que localiza los precios de los artículos y formula los pedidos
- Las herramientas de desarrollo Java para iSeries, para crear los servlets necesarios

- El Entorno de prueba WebSphere, para verificar la aplicación antes de desplegarla en el servidor iSeries mediante WebSphere Application Server

---

## Antes de empezar

Solo podrá llevar a cabo los ejercicios si se cumplen los siguientes prerequisites. En el tema Capítulo 3, “Ejecutar el caso práctico”, en la página 9 hallará más detalles sobre los prerequisites.

- Tiene acceso TCP/IP a un servidor iSeries.
- Ha iniciado los servidores iSeries con el mandato STRTCPSVR \*ALL.
- Ha iniciado el servidor WebFacing con el mandato STRTCPSVR \*WEBFACING.
- Ha restaurado las bibliotecas WHOLESale y RETAILSTOR en el servidor iSeries.
- Las bibliotecas WHOLESale, RETAILSTOR y QGPL figuran en la lista de bibliotecas.
- Ha llevado a cabo todas las tareas del Capítulo 3, “Ejecutar el caso práctico”, en la página 9 (salvo las tareas de WebSphere Application Server opcionales, que no son necesarias para probar la aplicación en el entorno de trabajo).

---

## Crear las páginas Web, los servlets, los archivos JSP y el código RPG

Para construir los componentes de SV000618:

1. Cree un proyecto Web destinado a contener todos los archivos que va a crear.
2. Escriba un archivo JSP ListItems que proporcione una lista de los artículos existentes en la base de datos del inventario de iSeries. Para escribir el servlet, puede utilizar Page Designer en las herramientas de desarrollo Web para iSeries. Concretamente, puede dibujar la página en la vista Diseño, y añadir el correspondiente código en la vista Fuente. También tendrá que insertar un control de tiempo de diseño (DTC) de subarchivo para que interactúe con el programa de servicio RPG TNLSTITM, rellenando el subarchivo con los registros de base de datos. Puede especificar el programa de servicio en los valores del control DTC.

A continuación tendrá que utilizar el asistente Interacción Web para crear la página de entrada:

- Especifique ListItems.jsp como página de salida para listar los artículos del inventario; así se asegura de que el página Interacción Web cree un archivo ListItems.wit.
- Asegúrese de que no especifica llamadas a programa en el asistente Interacción Web, porque el DTC de subarchivo invoca automáticamente el programa de servicio RPG TNLSTITM. El asistente también genera un servlet ListItemsWitServlet que funciona a modo de enlace para invocar la página ListItems.jsp.
- Para revisar el archivo ListItemsWit.wit generado por el asistente Interacción Web:
  1. Expanda **SV001618** y pulse **ListItems.wit** dos veces para visualizar el asistente de interacción del archivo.
  2. Pulse **Siguiente** mediante el asistente para revisar los valores especificados para la interacción.

A continuación, debe escribir una página de salida JSP GetCost con Page Designer, que toma los datos de entrada a partir de la página de entrada ListItems.jsp.

Cuando un usuario pulsa un artículo en la página `ListItems.jsp`, aparece una página `GetCost.jsp` que visualiza los detalles del artículo.

Tras crear la página `GetCost.jsp`, utilice el asistente Interacción Web para crear una interacción `WitOrder` entre el archivo `ListItems.jsp` (seleccionado como página de entrada) y el archivo `GetCost.jsp` (seleccionado como página de salida):

- En el asistente, en la página Llamada a programa, especifique una invocación del procedimiento `GetDetail` y su parámetro a partir del programa de servicio `RPG TNLSTITM`.
- En el procedimiento, se incorporan las API del DTC de subarchivo para determinar qué registro de subarchivo se ha seleccionado. El procedimiento utiliza esta información para recuperar el registro seleccionado a partir de la base de datos `INVENTORY`, y visualiza los detalles, incluida la imagen, del artículo seleccionado en `GetCost.jsp`.
- Para revisar el archivo `WitOrderWit.wit` generado por el asistente Interacción Web:
  1. Expanda **SV001618** y pulse **WitOrder.wit** dos veces para visualizar el asistente de interacción del archivo.
  2. Pulse **Siguiente** mediante el asistente para revisar los valores especificados para la interacción.

Fíjese en que el parámetro *flow* se especifica como controlador de flujo en la página de salida. Esto hace que el valor del parámetro asegure la visualización de los archivos JSP adecuados.

A continuación, debe importar los archivos WSDL (lenguaje de descripción de servicios Web) a partir del proyecto `SV000514` para que el administrador pueda recuperar el coste actual de un artículo a partir del suministrador al por mayor pulsando el botón **Obtener coste**.

- El botón **Obtener coste** invoca el archivo `QryProdCostServlet.jsp` y el correspondiente servicio Web del proyecto `SV000514`.
- Utilice el asistente Servicios Web y los archivos WSDL importados para generar el código de proxy Java necesario para invocar el servicio Web.
- El archivo `QryProdCostServlet.jsp` toma los datos de entrada de la página `GetCost.jsp`, utiliza el código de proxy Java para invocar el servicio Web `SV000514` con el fin de localizar el coste del artículo seleccionado, y visualiza el coste en una página que se llama `ReturnedCost.jsp`.
- Para ver el código de `QryProdCostServicesProxy.java` y el archivo `QryProdCostServlet.java`:
  1. Expanda **SV001618** > **source** > **proxy** > **soap**.
  2. Pulse dos veces el archivo **QryProdCostServicesProxy.java**.
  3. Para `QryProdCostServlet.java`, pulse dos veces el archivo **QryProdCostServlet.java**, situado en **SV001618** > **source**, y observe cómo crea una instancia del código de proxy Java.

Ahora tendrá que utilizar el asistente Interacción Web para enlazar el archivo `ReturnCost.jsp` como entrada y el archivo `Order.jsp` como salida para que el administrador pueda pulsar un botón **Pedir** con objeto de pedir el artículo seleccionado al suministrador al por mayor.

- Con esta interacción, no hace falta que utilice una llamada a programa, porque el enlace de las dos páginas es suficiente para visualizar la información correcta.
- Para ver el archivo `WitPlaceOrder.wit`:
  1. Expanda **SV001618**.

2. Pulse dos veces el archivo **WitPlaceOrder.wit** para abrir la interacción.
3. Pulse **Siguiente** mediante el asistente para revisar los valores especificados.

A continuación, utilice el servicio Web SV001586 para que el administrador pueda especificar la talla y la cantidad de los artículos solicitados.

- Importe los archivos WSDL SV001586 a este proyecto, genere el código de proxy Java para invocar el servicio Web, y escriba un servlet OrderSupplyServlet que se invoca cuando el usuario pulsa el botón **Pedir** en el archivo Order.jsp.
- El servlet reúne información a partir del archivo Order.jsp, invoca el código de proxy Java del servicio Web, que invoca el servicio Web SV001586 y solicita el artículo.
- El servlet visualiza el archivo OrderOK.jsp si el pedido es satisfactorio; en caso contrario, visualiza una página de error.
- Para ver cómo se crea una instancia del proxy de servicio Web generado y cómo se le invoca para que utilice el servicio Web SV001586:
  1. Expanda **SV000618 > source**.
  2. Pulse dos veces el archivo **OrderSupplyServlet.java** y examine el contenido.

---

## Antes de desplegar en WebSphere Application Server

Antes de desplegar la aplicación en WebSphere Application Server, tendrá que cambiar un URL concreto en los archivos de clase Java de SV000514 y SV001586 para que la aplicación funcione como es debido.

Para SV000514:

- En la clase de proxy de servicios Web, QryProdCostServicesProxy, está definida una variable que contiene el URL del servicio Web que hay que invocar.
- Cuando el proxy se crea por primera vez, este URL se establece en `http://localhost:9080/SV000514/servlet/rpcrouter`.
- Con la variable establecida en este valor, se invoca el servicio Web del proyecto SV000514 en el IDE, tal como se ha demostrado en el tema “Ejecutar la aplicación en el entorno de trabajo” en la página 17.
- Antes de desplegar esta aplicación, debe cambiar el valor de este URL para que señale hacia donde ha desplegado el archivo EAR, SVWholeSale.ear, en el directorio IFS de iSeries.

Para el proyecto SV001586:

- En la clase de proxy de servicios Web, OrderSupplyServicesProxy, está definida una variable que contiene el URL del servicio Web que hay que invocar.
- Cuando el proxy se crea por primera vez, este URL se establece en `http://localhost:9080/SV001586/servlet/rpcrouter`.
- Con la variable establecida en este valor, se invoca el servicio Web del proyecto SV001586 en el IDE, tal como se ha demostrado en el tema “Ejecutar la aplicación en el entorno de trabajo” en la página 17.
- Antes de desplegar esta aplicación, debe cambiar el valor de este URL para que señale hacia donde ha desplegado el archivo SVWholeSaleEAR.ear en el directorio IFS de iSeries.





---

## Capítulo 8. Avisos

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos. Es posible que IBM no ofrezca los productos, servicios o características descritas en esta documentación en otros países. Consulte a su representante local de IBM para obtener información acerca de los productos y servicios disponibles actualmente en su área geográfica. Las referencias a productos, programas o servicios de IBM no pretenden afirmar ni implican que sólo puedan utilizarse dichos programas, productos o servicios. En su lugar, puede utilizarse cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar la operativa de los productos, programas y servicios que no pertenezcan a IBM.

IBM puede tener patentes o patentes pendientes de aplicación referentes a los temas descritos en esta documentación. La adquisición de esta documentación no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por correo, a:

*IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.*

Para formular consultas sobre licencias referentes a información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país, o envíe las consultas por correo a:

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

**El párrafo que sigue no se aplica en los Estados Unidos ni en ningún otro país en el que tales disposiciones sean incoherentes con las leyes**

**locales:**INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPRESA O IMPLÍCITA, INCLUIDAS LAS GARANTÍAS O CONDICIONES IMPLÍCITAS DE NO VULNERABILIDAD, COMERCIALIZACIÓN O ADECUACIÓN A UN PROPÓSITO DETERMINADO, PERO SIN LIMITARSE A ELLAS. Algunos estados no permiten la declaración de limitación de responsabilidad con respecto a las garantías expresas o implícitas en determinadas transacciones y, por tanto, esta afirmación puede no ser procedente en su caso.

Esta documentación puede incluir inexactitudes técnicas o errores tipográficos. La información que contiene está sujeta a cambios periódicos, que se incorporarán en nuevas ediciones de la publicación. IBM puede efectuar mejoras y/o cambios en los productos y programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias efectuadas en esta información a sitios Web que no son de IBM se suministran únicamente a efectos de comodidad, y no implican garantía alguna

acerca de dichos sitios Web. Los materiales de esos sitios Web no forman parte de los materiales de este producto IBM, y el usuario asume la responsabilidad de utilizarlos.

IBM puede utilizar o distribuir la información suministrada por el cliente de cualquier forma que considere adecuada sin contraer ninguna obligación con respecto a él.

Los licenciatarios de este programa que deseen obtener información acerca del mismo con la finalidad de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido este) y (ii) la utilización mutua de la información intercambiada, deben ponerse en contacto con:

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

Tal información puede estar disponible, sujeta a los términos y condiciones adecuados, incluyendo en algunos casos el pago de una tasa.

IBM suministra el programa bajo licencia descrito en esta documentación y todo el material bajo licencia disponible para el mismo bajo los términos del IBM Customer Agreement (Acuerdo de cliente de IBM), IBM International Program License Agreement (Acuerdo internacional de licencia de programa de IBM) o cualquier acuerdo equivalente entre las partes.

La información concerniente a los productos que no pertenecen a IBM se ha obtenido de los proveedores de dichos productos, de sus anuncios publicados y de otras fuentes disponibles públicamente. IBM no ha probado dichos productos y no puede confirmar la exactitud de las declaraciones sobre rendimiento, compatibilidad ni de cualquier otro tipo relacionadas con productos que no pertenecen a IBM. Las preguntas relativas a las posibilidades de productos no IBM deben dirigirse a los proveedores de dichos productos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones habituales de gestión. Para ilustrarlos de la forma más completa posible, los ejemplos pueden incluir los nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con nombres y direcciones utilizados por empresas reales es mera coincidencia.

---

## LICENCIA DE COPYRIGHT:

Esta información ejemplos de programas de aplicación en lenguaje fuente, que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pago a IBM, con la finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación compatibles con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que se han escrito los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por tanto, IBM no puede garantizar la fiabilidad, capacidad de servicio ni funcionamiento de estos programas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pago a IBM, con la finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación compatibles con las interfaces de programación de aplicaciones de IBM.

Cada una de las copias, cualquier parte de estos programas de ejemplo o cualquier producto derivado debe incluir una aviso de copyright como el siguiente:

(C) (nombre de la empresa) (año). Partes de este código se derivan de IBM Corp. Sample Programs. (C) Copyright IBM Corp. 1992, 2002. Reservados todos los derechos.

---

## Información acerca de las interfaces de programación

La información acerca de las interfaces de programación tiene por objeto facilitar la creación de software de aplicación mediante este programa.


Las interfaces de programación de uso general permiten escribir software de aplicación que obtenga los servicios de las herramientas de este programa.

Sin embargo, esta información también puede contener información de diagnóstico, modificación y ajuste. La información de diagnóstico, modificación y ajuste se suministra para facilitar la depuración del software de aplicación.

**Aviso:** No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación, ya que está sujeta a cambios.

---

## Marcas registradas y marcas de servicio

- 400
- AFP
- AIX
- AIX windows
- APPN
- Application System/400
- AS/400
- AS/400e
- BookManager
- C Set ++
- C/400
- CICS
- CICS/400
- CICS/ESA
- COBOL/2
- COBOL/400
- Common User Access
- CUA
- DB2
- DB2 Extenders
- DB2 Universal Database
-  **eServer**
- GDDM
- IBM
- IBMLink
- Integrated Language Environment
- iSeries
- Language Environment
- MQSeries
- Network Station
- Open Class
- Operating System/2
- Operating System/400
- OS/2
- OS/390
- OS/400
- POWER2
- PowerPC
- PROFS
- RPG/400
- RS/6000
- S/390
- SAA
- SQL/400
- System/36
- System/38
- VisualAge
- VTAM
- WebSphere

InstallShield es una marca registrada de InstallShield Corporation.

Intel y Pentium son marcas registradas de Intel Corporation en los Estados Unidos o en otros países.

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

Rational es una marca registrada de International Business Machines Corporation y Rational Software Corporation en los Estados Unidos o en otros países.

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

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos o en otros países.

Netscape Navigator es una marca registrada de Netscape Communications Corporation.

UNIX es una marca registrada de The Open Group.

Otros nombres de empresas, productos y servicios pueden ser marcas registradas de terceros.





Número de Programa: 5724-A81

Impreso en España