

# Desarrollo Basado en Componente y Servicios

## Servicios Web basados en SOAP

### Introducción

Los conceptos teóricos son vistos en teoría, por lo que aquí nos vamos a centrar en cuestiones prácticas de implementación usando JEE y Netbeans, sobre Glassfish. Se mostrará una guía sencilla de cómo hacerlo, y una serie de ejercicios básicos que permitan abordar la práctica.

Se mostrará cómo crear y acceder a los servicios web basados en SOAP

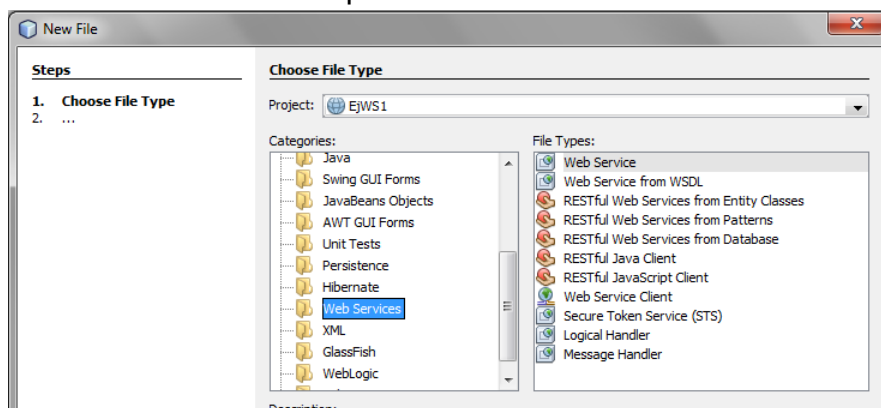
Vamos a empezar mostrando de manera resumida cómo crear y acceder a un servicio web basado en SOAP. Más información en <https://netbeans.org/kb/docs/websvc/jax-ws.html>.

### 1. Creando el contenedor

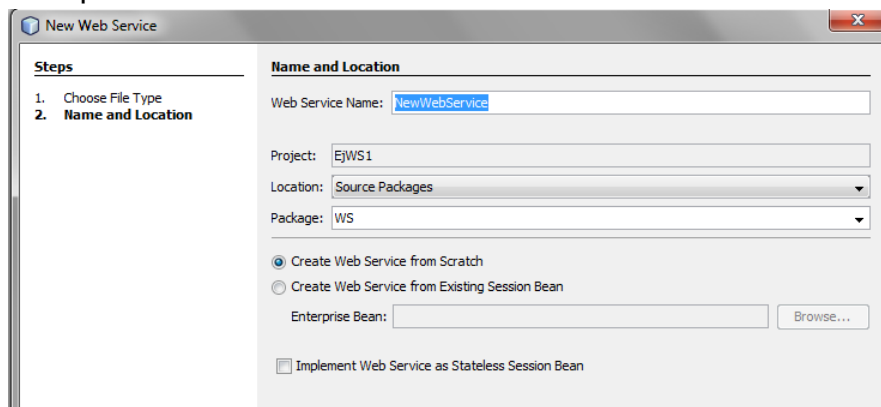
Un servicio web se puede desplegar contenido tanto en una aplicación web como en un módulo EJB, por lo que lo primero que habrá que hacer es crear un proyecto de cualquiera de estos tipos.

### 2. Creando un servicio web desde cero (“scratch”)

Añadir al proyecto un nuevo fichero de tipo “Web Service”

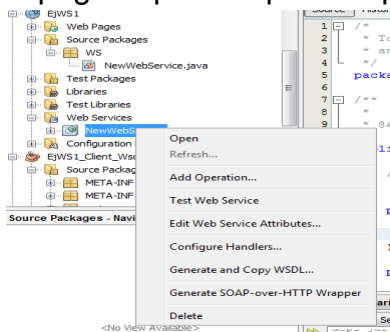


En el formulario siguiente indicar el nombre que le queremos dar, el paquete al que le queremos añadir y seleccionar la opción “Create Web Service from Scratch”



Al finalizar se habrá creado un .java con el típico método hola mundo. De momento el único de nuestro servicio web

Tras compilar el proyecto y desplegarlo, con Netbeans se puede probar sin necesidad de usar un cliente. Para ello en el menú contextual asociado al servicio web escoger la opción “Test Web Service”. Se abre en el navegador una página que nos permite probar el servicio web creado.



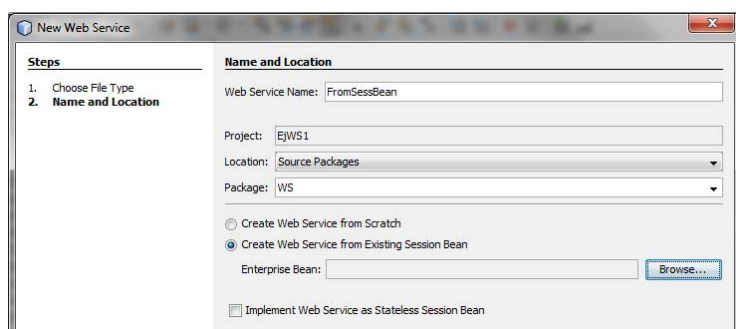
En el mismo menú contextual tenemos la opción de añadir operaciones al servicio web.

### 3. Creando un servicio web desde un EJB de sesión sin estado

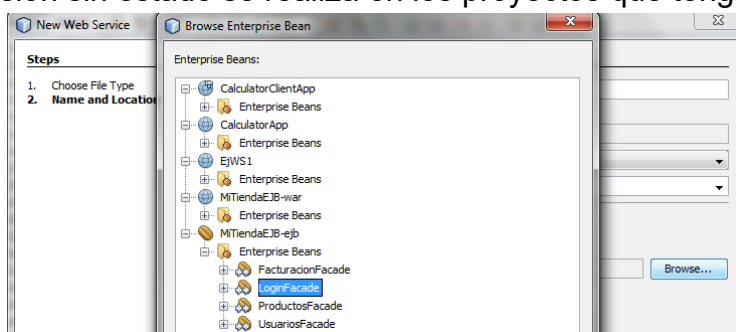
La primera opción es “hacerlo a mano”, a partir del EJB; se copia al nuevo proyecto (si se ha creado) y se añaden las anotaciones vistas en teoría. Si no existe una interface, el servidor la creará automáticamente al desplegar el servicio web; esta creación se realiza durante las operaciones de procesamiento de la anotación `@WebService` por parte del servidor. Esta interface, llamada Service Endpoint Interface (SEI), se usará, como en el caso de los EJB, para inyectar la referencia al servicio web, cuando el cliente se ejecute dentro del contenedor (anotación `@WebServiceRef`). El nombre de la SEI es dependiente del servidor usado, en caso de que sea creada por éste.

La segunda opción es usar Netbeans, aunque en este caso, lo que vamos a crear realmente es un servicio web que va a usar las operaciones ya implementadas en el EJB de sesión sin estado. Veamos cómo hacerlo.

En el formulario que aparece al insertar “New→Web Service” seleccionar la opción de crearlo desde un “Session Bean”.



Seleccionar el Bean que queremos convertir en servicio web y habremos acabado. AVISO: la búsqueda de EJB de sesión sin estado se realiza en los proyectos que tengamos abiertos.



**IMPORTANTE:** SI QUEREMOS QUE SE CREEN COMO SERVICIOS TODAS LAS OPERACIONES CONTENIDAS EN EL BEAN, LA INTERFACE DEBE ESTAR ACCESIBLE AL PROYECTO DONDE ESTAMOS CREANDO EL SERVICIO WEB, ANTES DE CREAR ÉSTE. CASOS:

- Si creamos el servicio web en el mismo proyecto que está el Bean no habrá problema, ya está la interface.
- Si no es así y la interface es remota, habrá que añadir a las bibliotecas, ANTES DE CREAR EL SERVICIO WEB, el proyecto que contenga la interface remota
- Si la interface es local, habrá que copiar, ANTES DE CREAR EL SERVICIO WEB, el fichero con la interface local.

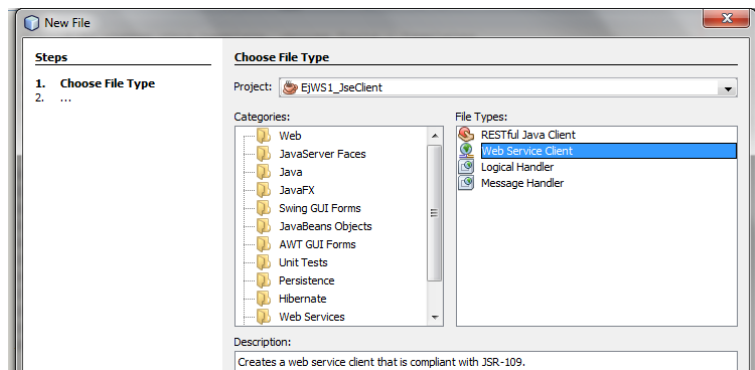
SI LA INTERFACE NO ESTÁ SE CREA EL ESQUELETO DEL SERVICIO WEB, PERO SIN OPERACIONES. HABRÁ QUE AÑADIRLAS A MANO.

#### 4. Accediendo desde un cliente

A un servicio web se puede acceder desde cualquier aplicación independientemente del lenguaje en que esté desarrollada. Aquí, como en la URL indicada al iniciar esta parte (<https://netbeans.org/kb/docs/websvc/jax-ws.html>), se muestra cómo hacerlo en los casos más típicos usando Java: cliente JSE, servlet o página JSP y cliente JEE. La forma de operar desde Netbeans es similar en todos los casos.

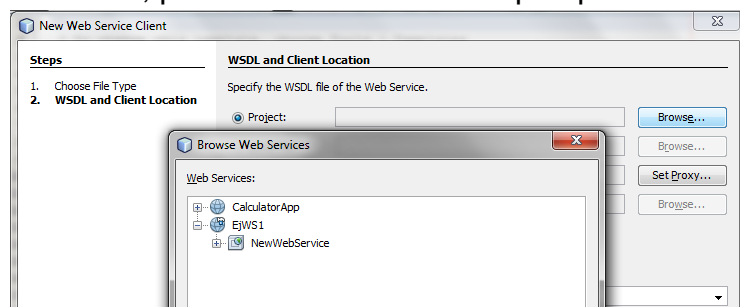
Lo primero será crear el contenedor: proyecto JSE, proyecto Web o cliente de aplicación enterprise.

A continuación hay que acceder al WSDL del servicio web a usar. Para ello seleccionar “New→Web Service Client”

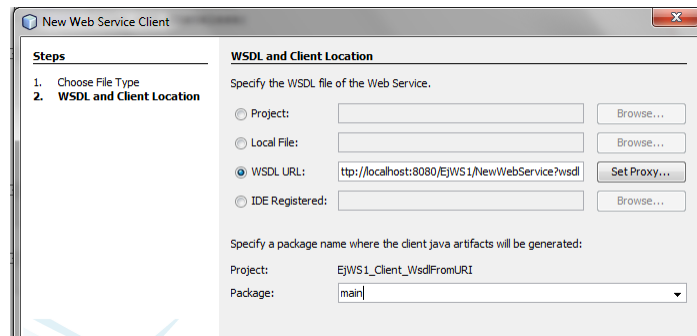


A continuación aparecerá el formulario donde especificar el WSDL correspondiente al servicio web a acceder. Hay dos opciones básicas de hacerlo:

- Si el servicio web se ha creado en un proyecto Java que todavía tenemos abierto, podemos seleccionar la opción “Project”. Al hacer clic sobre “Browse” se nos mostrarán los servicios web asociados a proyectos abiertos, pudiendo seleccionar el que queramos.



- II. Mediante la URL del WSDL. Esta segunda forma es más general y nos permite crear un cliente para cualquier servicio web.



Si queremos acceder de esta manera a un servicio web creado por nosotros y ya desplegado en el servidor, la URL de acceso al servicio web es mostrada en los mensajes de la consola del servidor durante el despliegue. Aparece un mensaje como el siguiente:

*NewWebService listening at address at http://Carlos-HP:8080/EjWS1/NewWebService*

Para acceder al fichero WSDL basta con añadir a esa URL “?wsdl”. Es decir, en el caso anterior la URL del WSDL será:

*NewWebService listening at address at http://Carlos-HP:8080/EjWS1/NewWebService?wsdl*

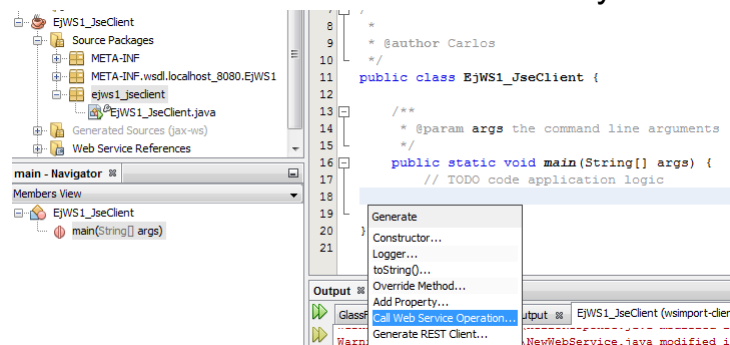
Se puede probar a poner esa URL en el navegador y ver como se nos muestra, efectivamente, el fichero WSDL.

Se puede o no crear un paquete donde los ficheros del cliente serán generados (ver la ayuda que aparece del formulario para más información). Esto siempre se puede modificar posteriormente en los ficheros xml de configuración. Por defecto, mejor dejarlo vacío.

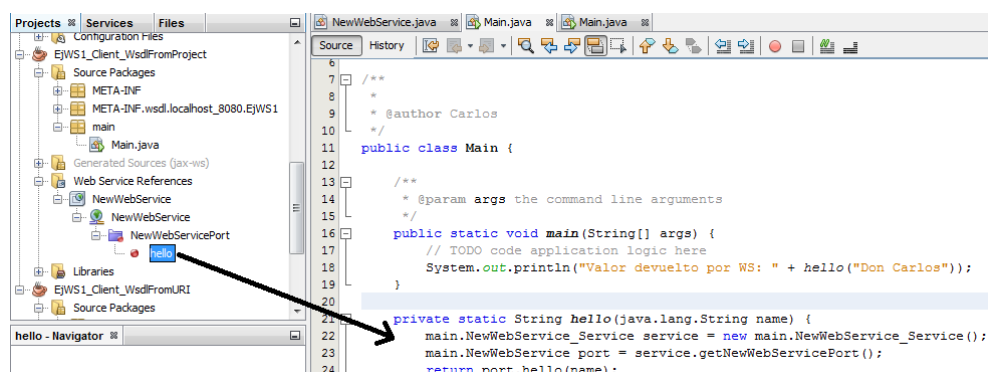
Al hacer clic en finalizar ya tenemos en nuestro cliente una referencia al servicio web, que vamos a usar como se indica a continuación para llamar a sus servicios.

Para invocar estos servicios desde el main() del cliente JSE, el servlet o el main() del cliente para aplicación enterprise tenemos dos opciones:

- Clic con el botón derecho dentro de la ventana de edición y seleccionar añadir código.

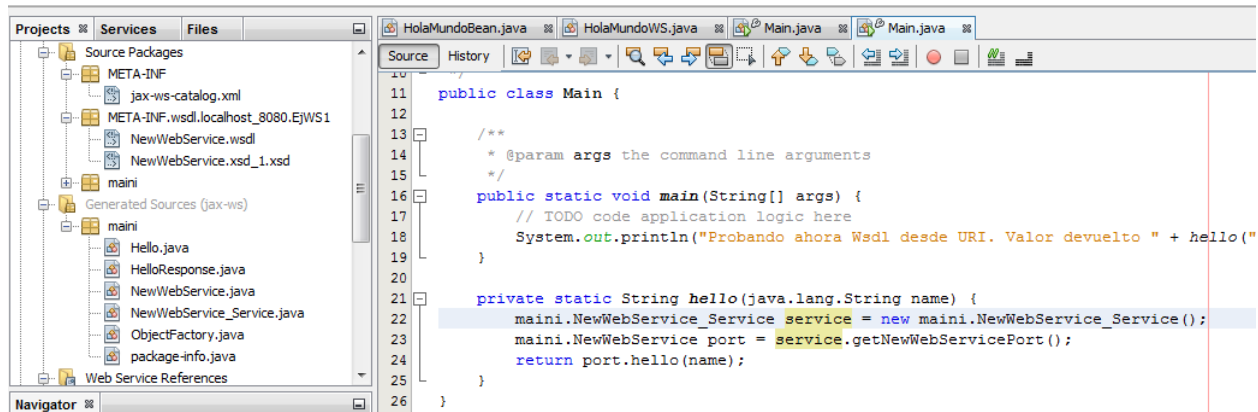


- Abrir el menú contextual asociado al servicio web, seleccionar el servicio que queremos llamar y arrastrar hasta la ventana de edición.

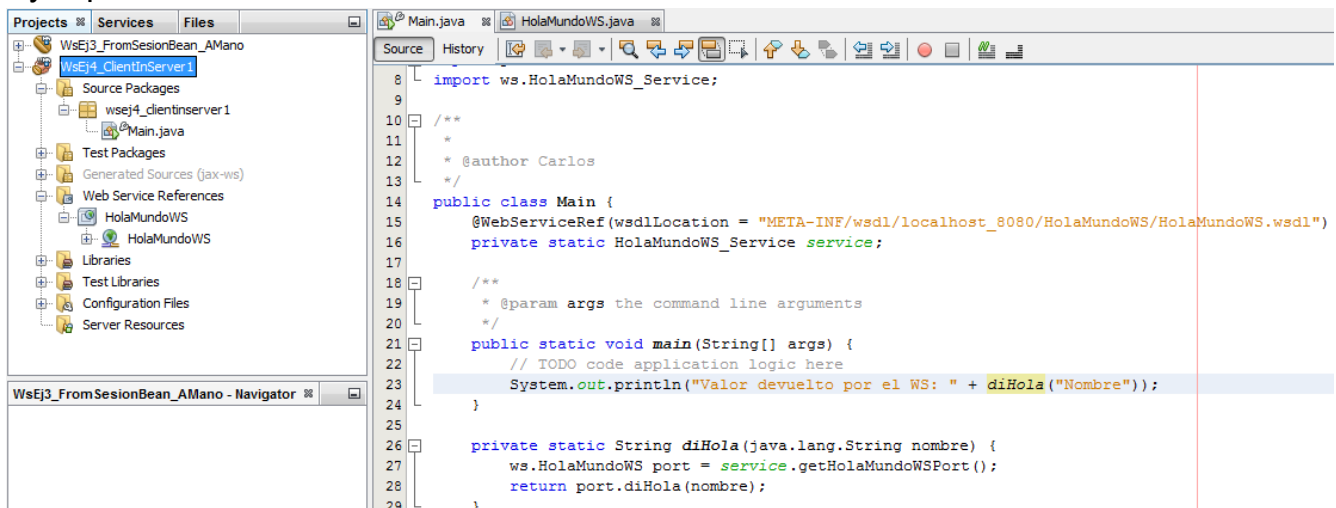


El código añadido será diferente en cada caso, ya que mientras el cliente JSE se ejecuta fuera del servidor (contenedor) los otros dos clientes son ejecutados por este, pudiendo usar anotaciones (`@WebServiceRef`) para inyectar una referencia al servicio web. Veamos algunos ejemplos:

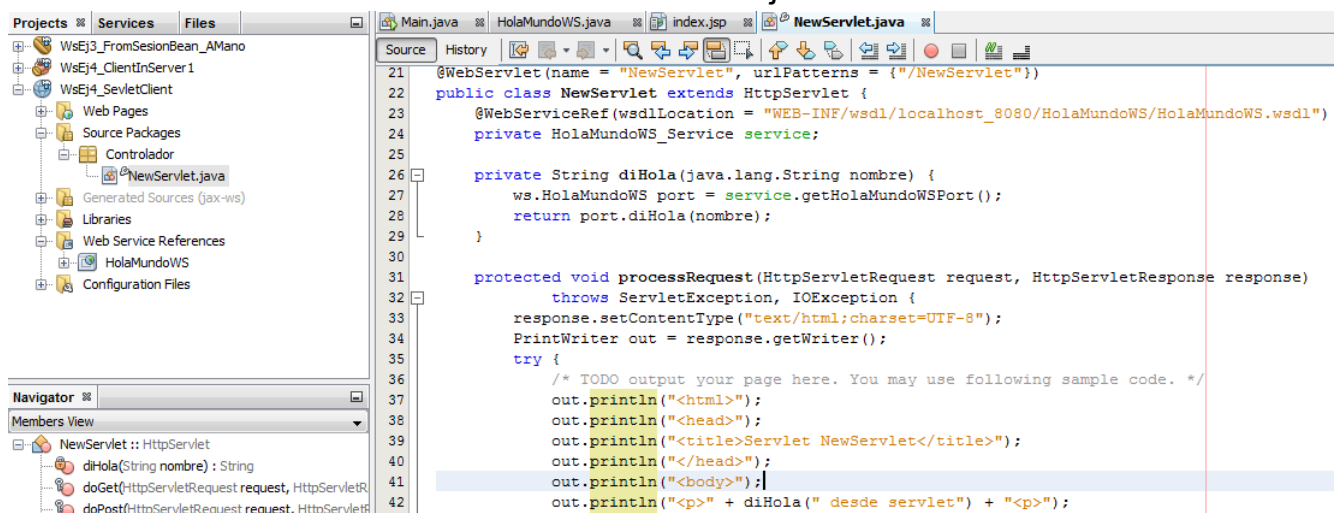
- Fuentes añadidos al proyecto y código insertado en `main()` en el caso cliente JSE



- Partes añadidas al proyecto y código en `main()` en caso de cliente para aplicación Enterprise. Como este cliente se ejecuta dentro del servidor, ahora se usa la anotación `@WebServiceRef`. Como se puede ver el cliente es más simple que en el caso anterior, ya que se usan los servicios del contenedor.



- Caso servlet. Es similar al caso anterior al ser ejecutado desde dentro del contenedor



### **Ejercicio 1. Creando un SW desde cero. Conversor Kilómetros – Millas**

Crear un servicio web que convierta kilómetros en millas, es decir, se le pasarán una cantidad que en kilómetros y devolverá el número de millas correspondiente. Usar como contenedor una aplicación web. Compilar el proyecto, desplegarlo y probarlo mediante la prueba del menú contextual asociada al servicio web, para así evitar de momento crear un cliente.

### **Ejercicio 2. Creando un SW desde un Bean de sesión “a mano”. Conversor Kilómetros – Millas**

Recuperar el proyecto realizado en el ejercicio 3 de la hoja de prácticas sobre EJB de sesión. Vamos a usar ahora como contenedor un proyecto tipo módulo EJB. Copiar el EJB de sesión creado en el ejercicio 3 indicado en el nuevo proyecto y añadir “a mano” las anotaciones `@WebService`, `@WebMethod` y `@WebParam` como se ha visto en teoría para convertir el EJB de sesión sin estado en un servicio web. Compilar el proyecto, desplegarlo y probarlo mediante la prueba del menú contextual asociada al servicio web.

### **Ejercicio 3. Creando un SW desde un Bean de sesión. Conversor Kilómetros – Millas**

Repetir el ejercicio anterior pero ahora usando la opción de crear servicio web desde Bean de sesión existente. Compara con la alternativa del ejercicio anterior.

### **Ejercicio 4. Accediendo desde cliente**

Acceder a cualquiera de los servicios web creados en los ejercicios anteriores mediante cada uno de los clientes especificados en el apartado 4