

Creación de servicios Web con Apache Axis2 desde una clase Java

En este ejercicio se creará un Servicio Web desde la plataforma de desarrollo Eclipse con el plugin Web Tool Project (WTP) utilizando el framework ApacheAxis 2. Además, como contenedor que de soporte a la ejecución de Apache Axis2 usaremos el contenedor Web Tomcat.

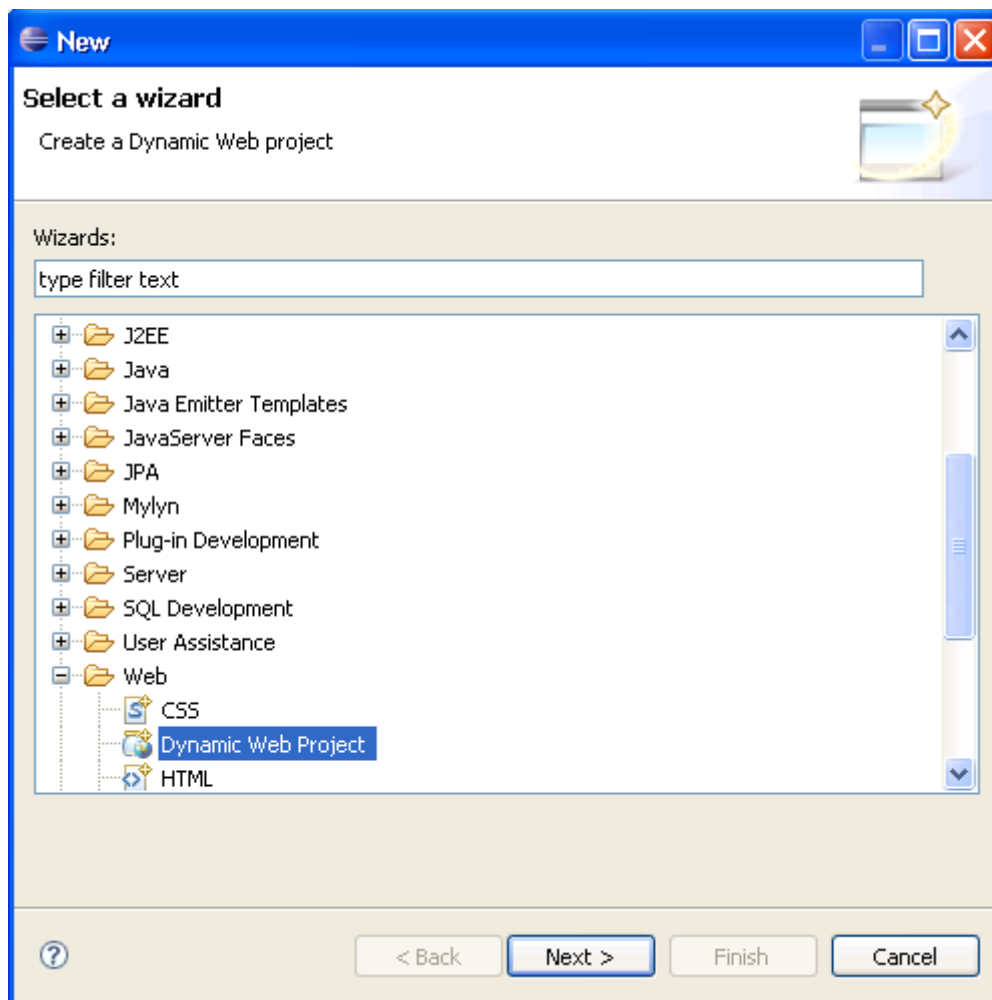
Prerrequisitos

1. Los mismos que en el ejercicio de la configuración de AXIS 2 en Eclipse.

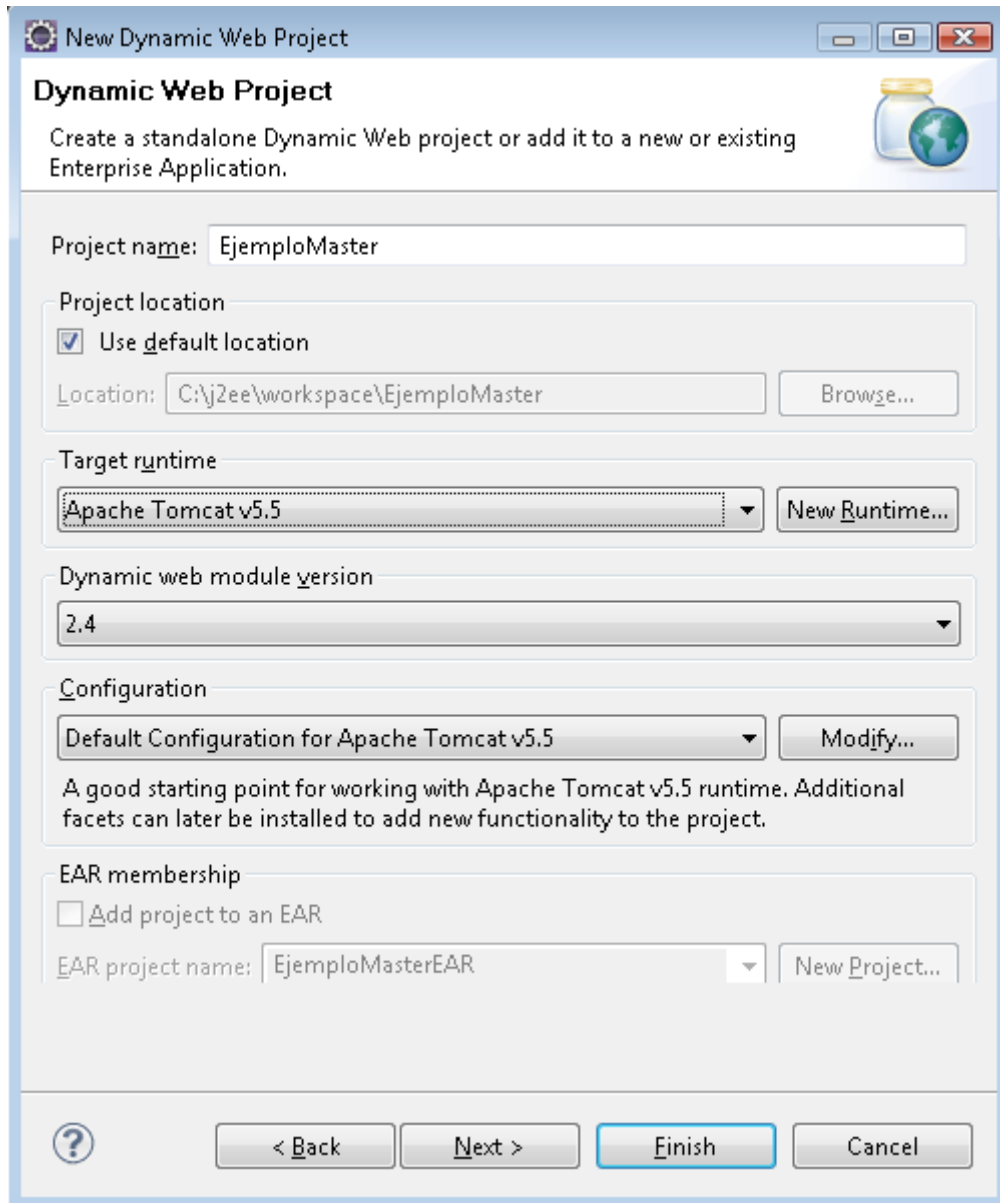
Pasos a seguir

1. Creación de proyecto Web dinámico

- 1.1. Iniciar Eclipse (si no ha sido ejecutado previamente)
- 1.2. Abrir *File* → *New* → *Other* → *Web* → *Dynamic Web Project*



- 1.3. Pulsar sobre *Next*
- 1.4. Completar los datos que se solicitan en el formulario (nombre del proyecto, entorno de ejecución, etc.).

The image shows a 'New Dynamic Web Project' dialog box in an IDE. It has a title bar with standard window controls. The main area is divided into several sections: 'Project name' with a text field containing 'EjemploMaster'; 'Project location' with a checked 'Use default location' checkbox and a text field showing 'C:\j2ee\workspace\EjemploMaster' next to a 'Browse...' button; 'Target runtime' with a dropdown menu set to 'Apache Tomcat v5.5' and a 'New Runtime...' button; 'Dynamic web module version' with a dropdown menu set to '2.4'; 'Configuration' with a dropdown menu set to 'Default Configuration for Apache Tomcat v5.5' and a 'Modify...' button, followed by a descriptive text block; and 'EAR membership' with an unchecked 'Add project to an EAR' checkbox, a text field for 'EAR project name' containing 'EjemploMasterEAR', and a 'New Project...' button. At the bottom, there is a help icon, and four buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: EjemploMaster

Project location
☒ Use default location
Location: C:\j2ee\workspace\EjemploMaster Browse...

Target runtime
Apache Tomcat v5.5 New Runtime...

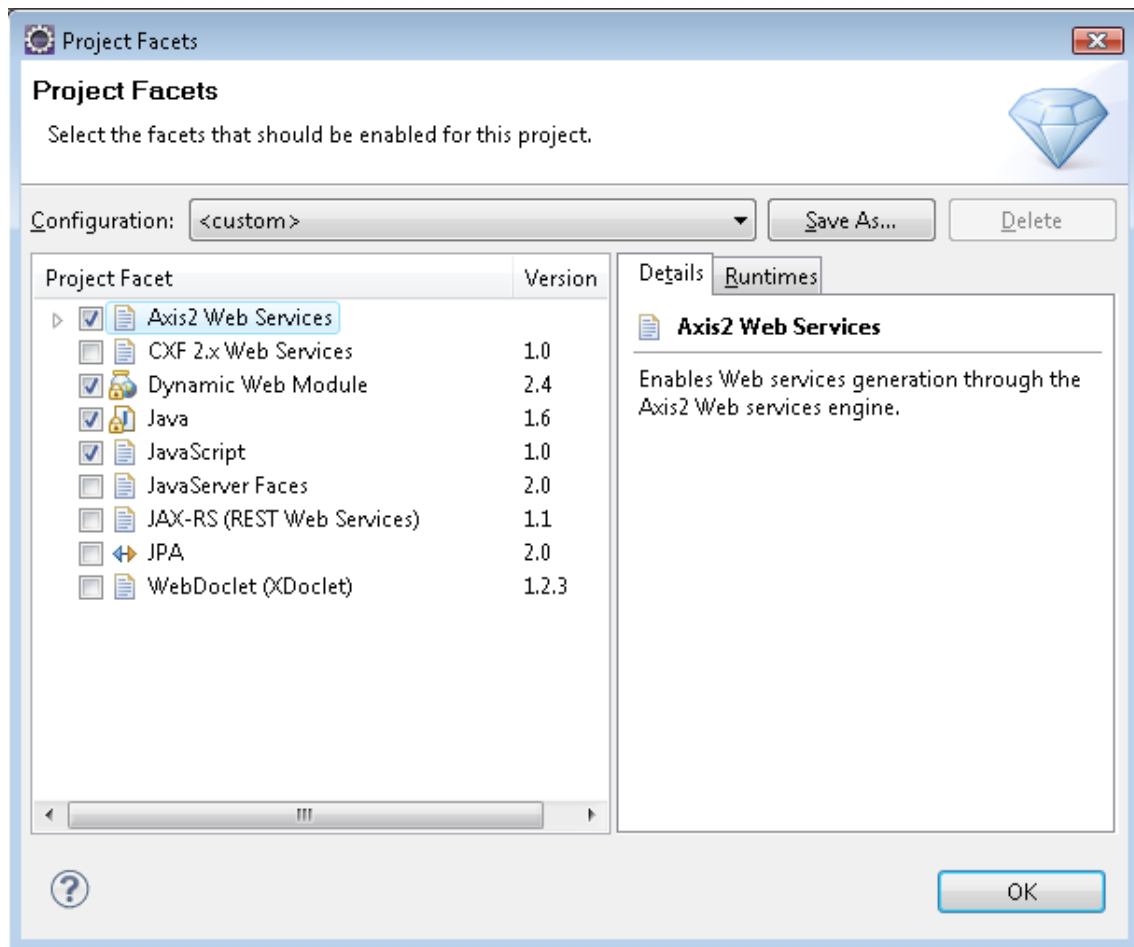
Dynamic web module version
2.4

Configuration
Default Configuration for Apache Tomcat v5.5 Modify...
A good starting point for working with Apache Tomcat v5.5 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership
☐ Add project to an EAR
EAR project name: EjemploMasterEAR New Project...

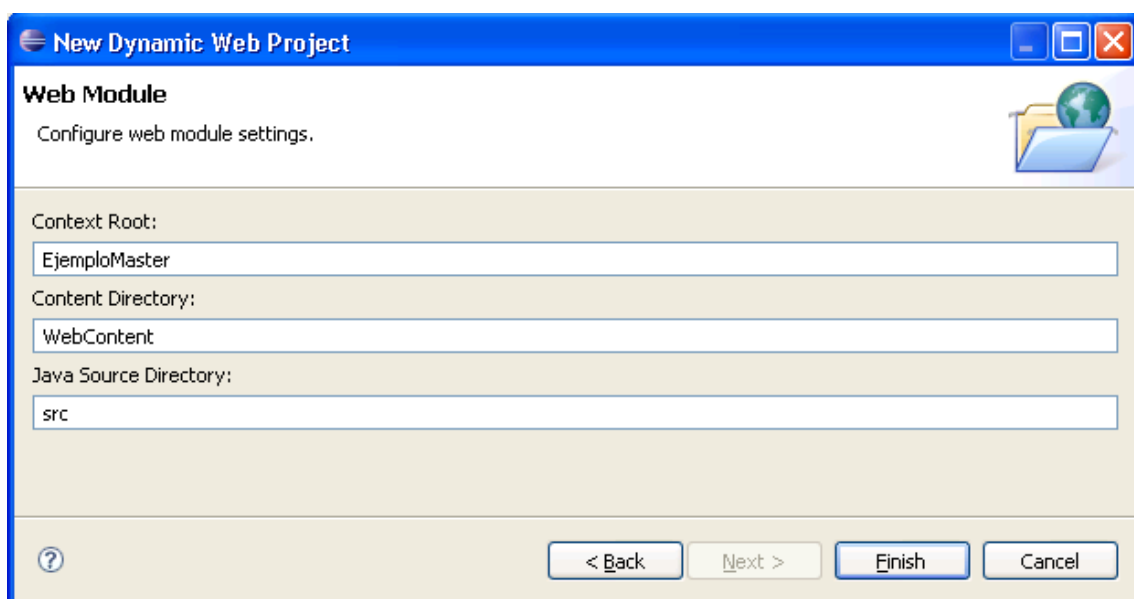
? < Back Next > Finish Cancel

- 1.5. Pulsamos en *Modify* y completamos los datos del siguiente formulario seleccionando la opción de Apache AXIS 2.

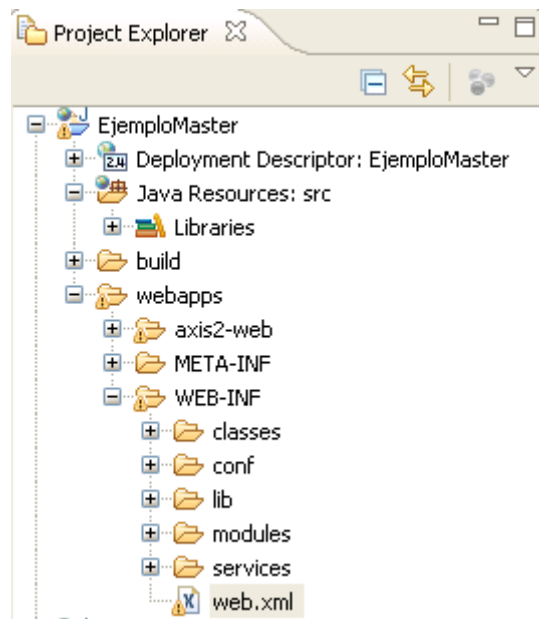


1.6. Pulsamos Ok y *Next*.

1.7. Para ser coherentes con el despliegue de la aplicación sobre el contenedor Apache Tomcat cambiamos el nombre del *Content Directory* y en lugar de *WebContent* le daremos el nombre del contexto con el cual lo vayamos a desplegar posteriormente en el servidor Tomcat (por ejemplo, EjemploMaster).



1.8. Para concluir pulsamos *Finish*.

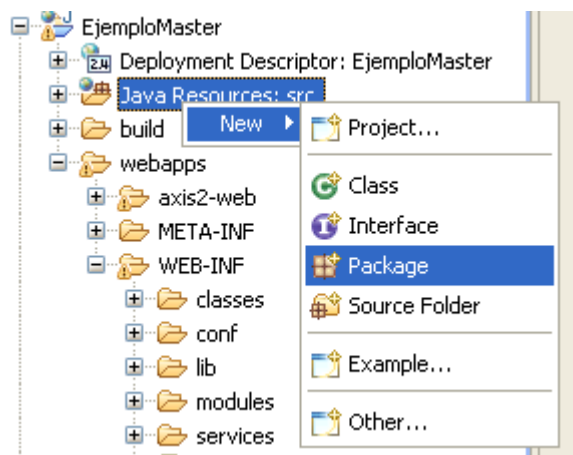


La estructura del proyecto se ha generado correctamente. Aunque aparezcan algunos avisos, no influyen sobre el funcionamiento de la aplicación.

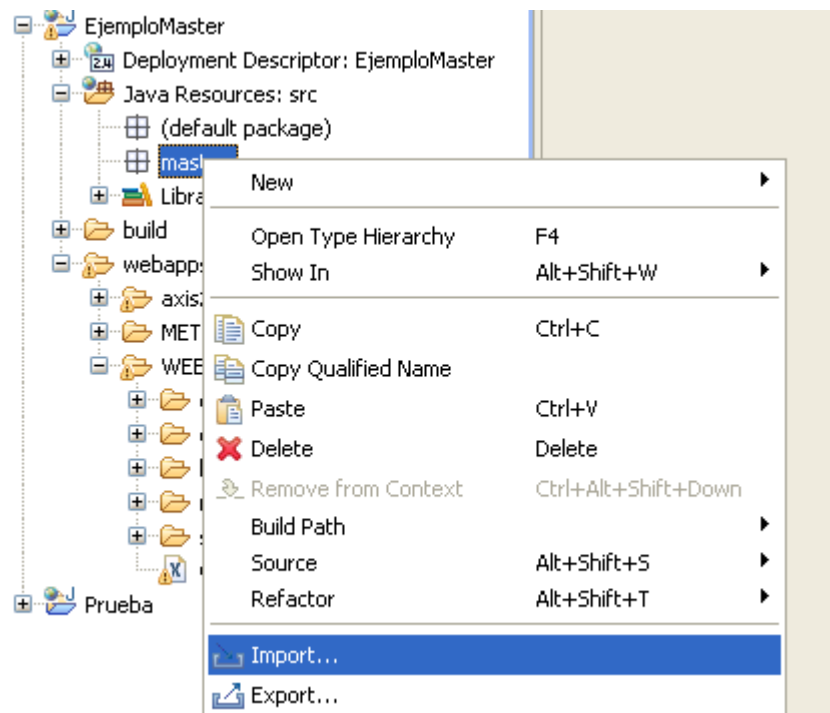
2. Generar Servicio Web desde una clase Java

2.1. Importar la clase Java de ejemplo *converter.java* al proyecto creado anteriormente. Esta clase simplemente convierte de grados Celsius a Fahrenheit y viceversa. Como ejercicio se puede añadir el convertir de los dos anteriores a Grados Centígrados y viceversa.

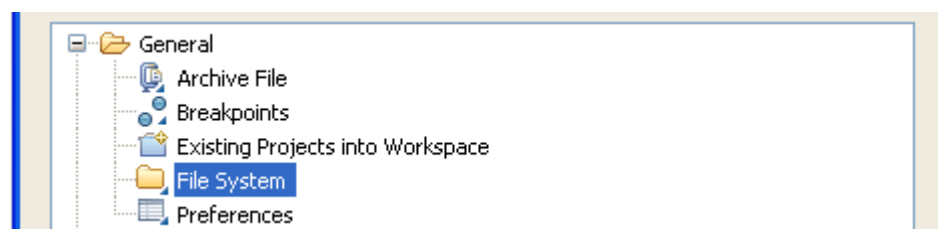
- Creamos primero un paquete llamado *master*.



- Y posteriormente importamos la clase *Converter.java* al proyecto dentro del paquete. Para ello pulsamos sobre el paquete derecho con el botón derecho y seleccionamos la opción *import...*



- Seleccionamos importar *General* → *File System* y elegimos el archivo que queremos importar *Converter.java* de donde lo hayamos ubicado.



NOTA: Para utilizar el generador automático de servicios Web de Eclipse para AXIS se debe seguir la convención de nombrar los paquetes en minúsculas y las clases Java deben comenzar por mayúscula. Si no el generador no podrá trabajar pero tampoco nos indicará cuál es el error.

- 2.2. Seleccionado sobre la clase *Converter.java* y pulsando el botón derecho aparece una opción llamada *Web Service* → *Create Web Service*. Otra forma de hacerlo es desde *File* → *New* → *Other...* → *Web Services* → *Web service*. La primera opción selecciona la clase automáticamente, en la segunda se debería especificar.

Web Service

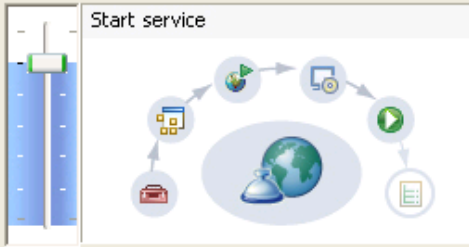
Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Web service type: Bottom up Java bean Web Service

Service implementation: master.Converter [Browse...](#)

Start service

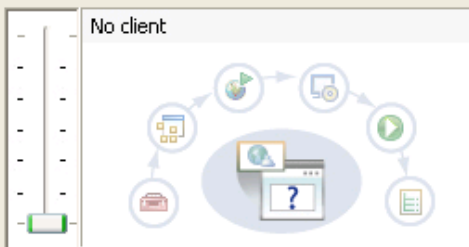


Configuration:

- [Server: Tomcat v5.5 Server](#)
- [Web service runtime: Apache Axis2](#)
- [Service project: EjemploMaster](#)

Client type: Java Proxy

No client



Configuration: No client generation.

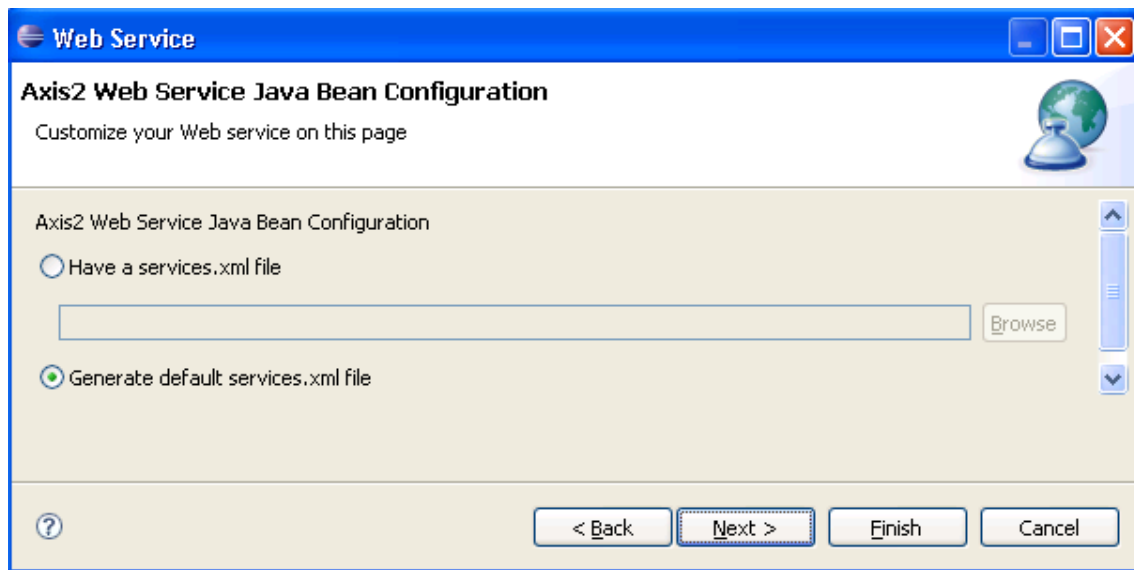
☐ Publish the Web service

☐ Monitor the Web service

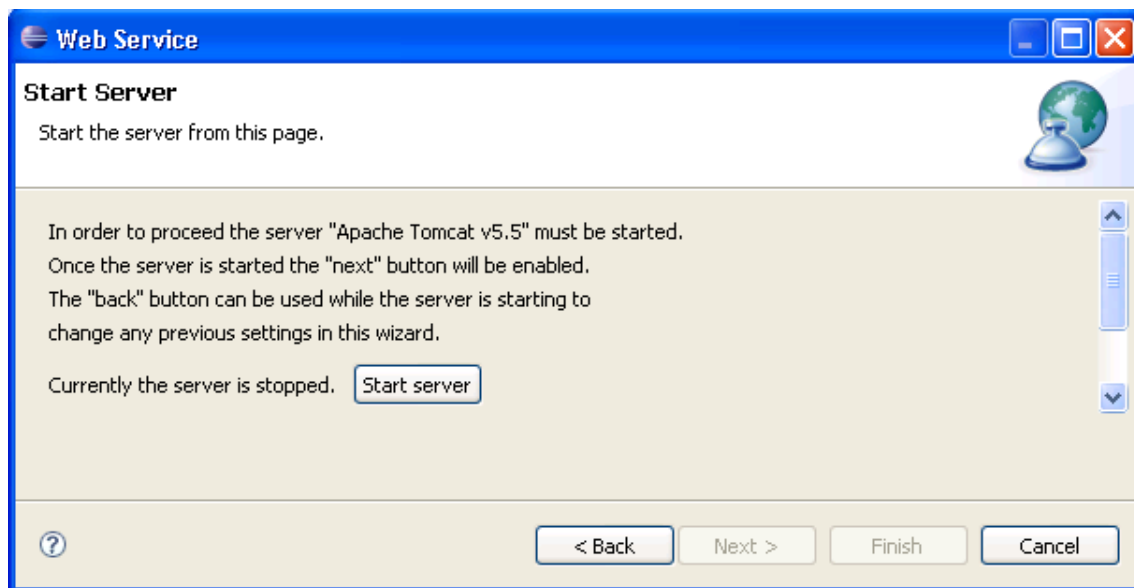
☐ Do not show me this dialog box again.

En el momento de generar el Servicio Web podemos elegir el nivel de generación del Servicio (Develop, Assemble, Deploy, Install, Stara y Test). También nos permite generar un cliente dentro del proyecto para probar el servicio con los mismos niveles de generación que el servicio. En este caso el cliente lo genera dentro de un proyecto Web (para el ejemplo el cliente lo generaremos más adelante). Podríamos modificar los entornos de ejecución que aparecen a la derecha, simplemente pulsando sobre el enlace y seleccionado el adecuado.

2.3. Pulsamos en *Next* >.



2.4. Vamos avanzando con las opciones que aparecen por defecto. Hasta que se nos indique que arranquemos el servidor Apache Tomcat. Esta opción aparece porque hemos indicado como opción despliegue y prueba. Eclipse lanza una instancia nueva de Apache Tomcat, por lo que **si tenemos arrancado el servidor previamente debemos pararlo**. Posteriormente pulsamos *Start Server*.



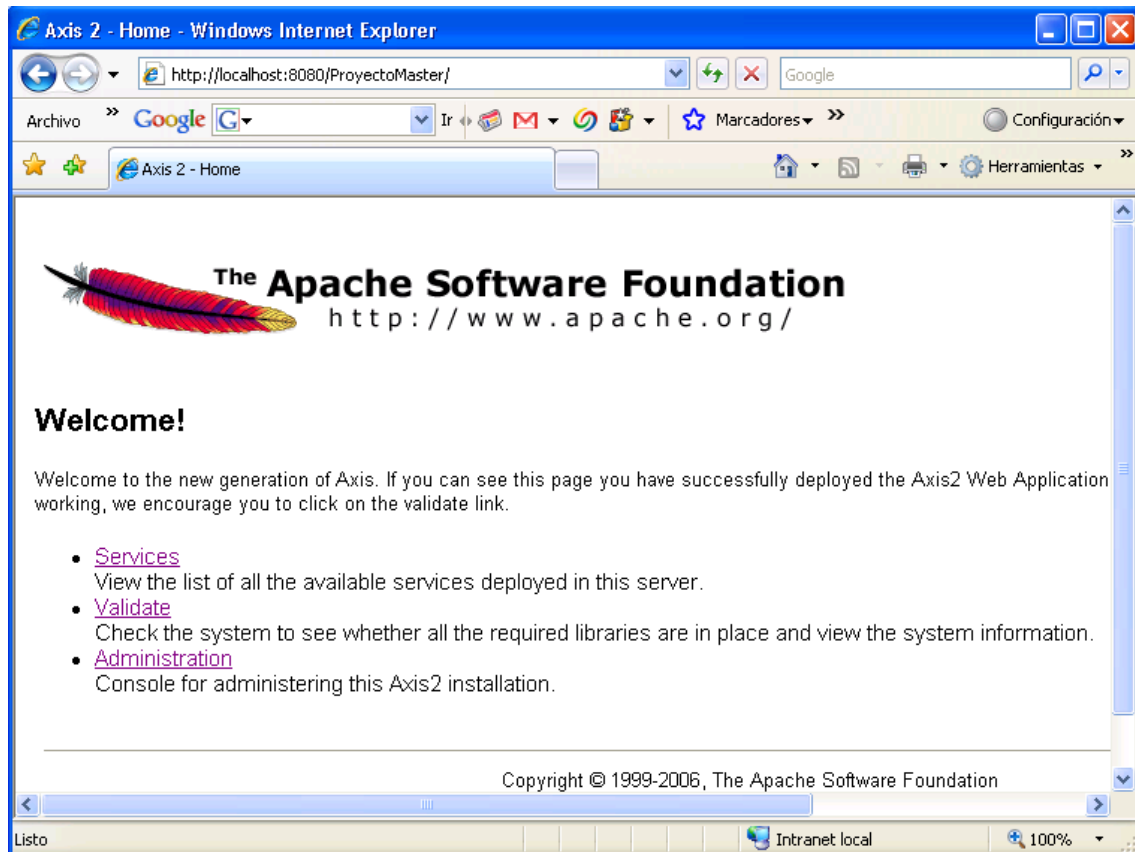
2.5. Una vez arrancado el servidor pulsaremos *Finish*.

3. Despliegue del Servicio Web en el servidor Apache Tomcat.

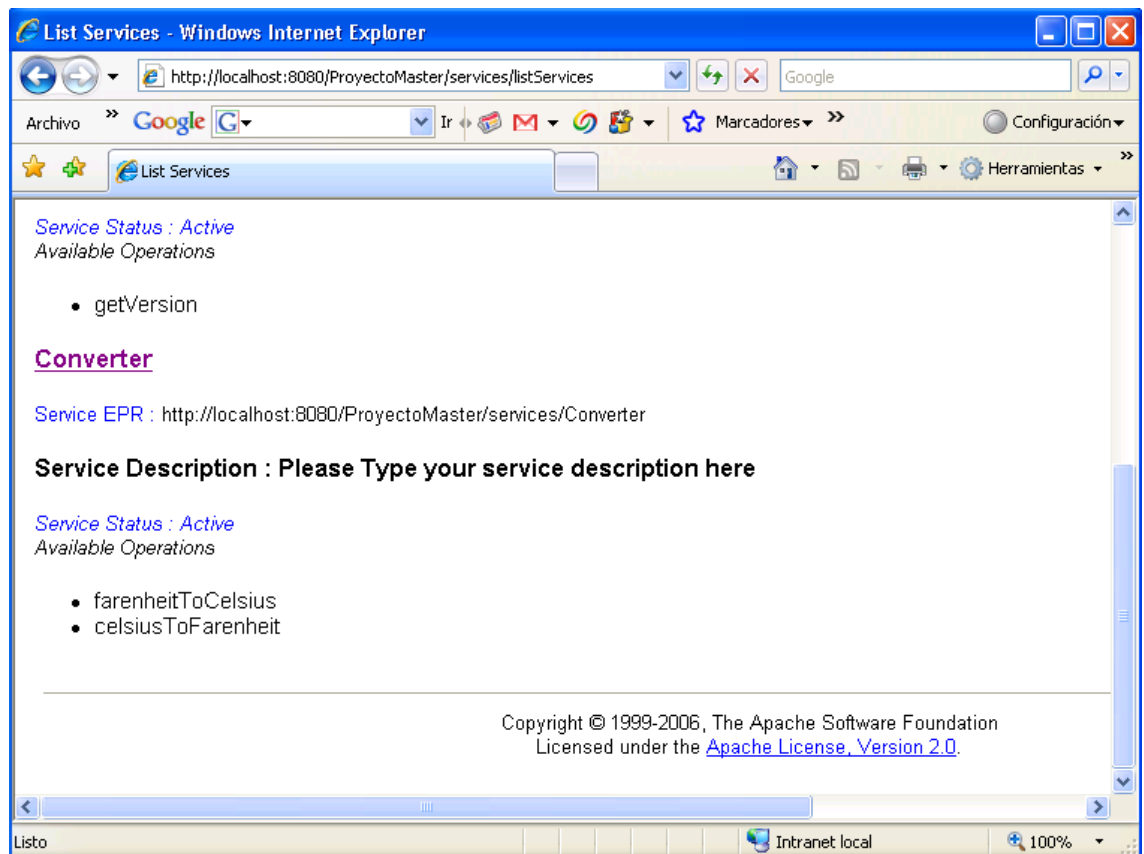
Para desplegar el Servicio Web en la distribución de apache Tomcat que tengamos instalada, sin que se lance desde el entorno Eclipse se deben seguir los siguientes pasos.

- 3.1. Crearemos una carpeta en el directorio *webapps* del servidor Apache Tomcat con el nombre del contexto que se considere apropiado (como comentamos anteriormente por ejemplo, ProyectoMaster).
- 3.2. Copiaremos en esta carpeta el contenido de la carpeta *webapps* generada en el proyecto.
- 3.3. Iniciamos el servidor Apache Tomcat y una vez arrancado comprobamos si el servicio ha sido desplegado.

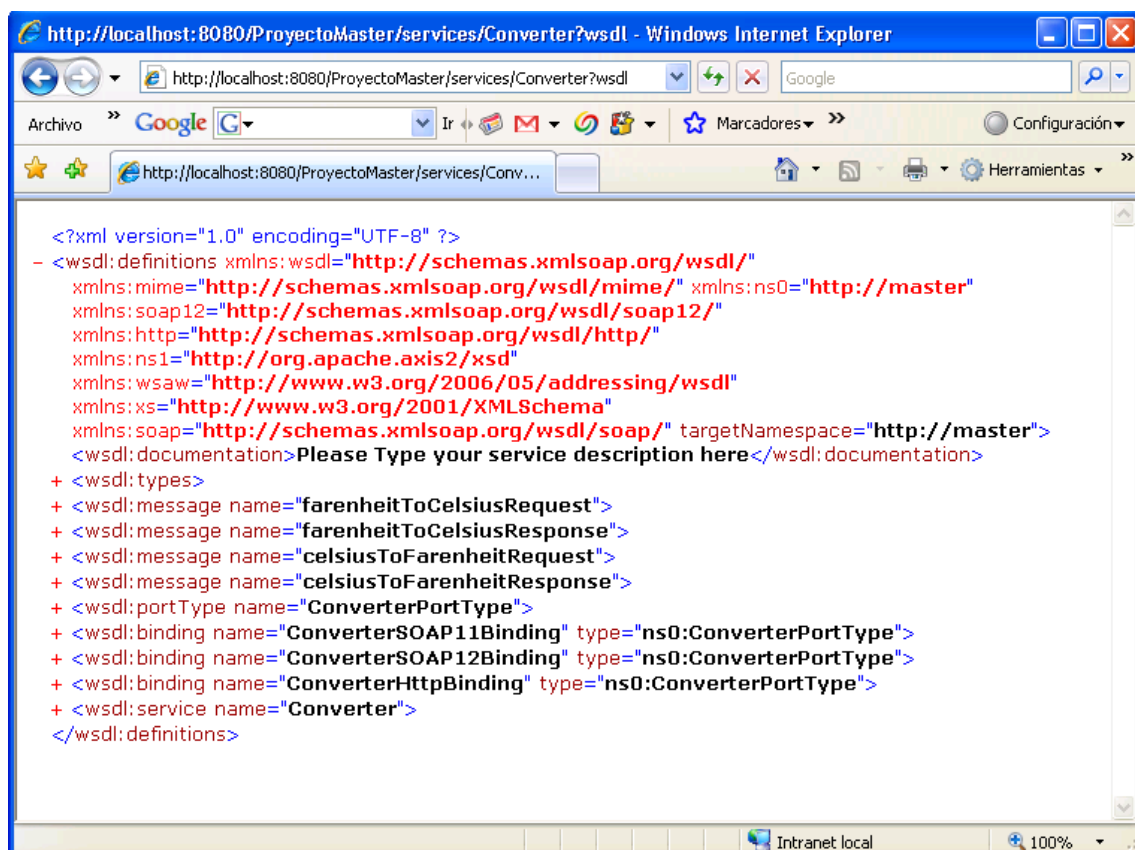
También se podría desplegar desde el entorno de Eclipse exportándolo como una aplicación de tipo WAR.



- 3.4. Accedemos al enlace *Services* para ver la lista de servicios y comprobar que el servicio *Converter* se encuentra disponibles, como se muestra en la imagen inferior.



3.5. Pulsando sobre el nombre del Servicio accederemos al documento WSDL que describe el servicio.



NOTA: Se ha configurado Eclipse para poder trabajar con Apache AXIS 2, pero por defecto Eclipse WTP incluye Apache AXIS. En principio, y como entrenamiento se puede utilizar Apache AXIS para generar los servicios Web, aunque el rendimiento en apache AXIS 2 se haya mejorado y se hayan añadido algunas características de WS-*.