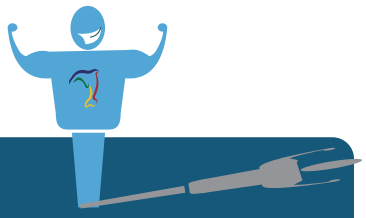
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

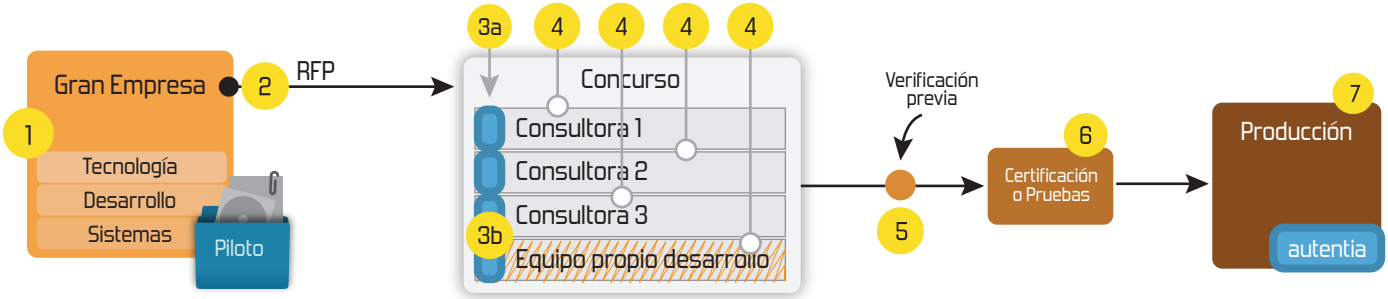
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

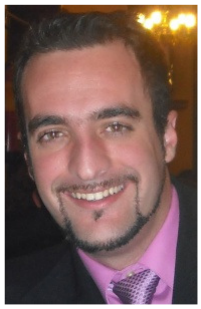
Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

E-mail:
Contraseña:
Entrar
Deseo registrarme
He olvidado mis datos de acceso

Inicio Quiénes somos Tutoriales Formación Comparador de salarios Nuestro libro Charlas Más

Estás en:
Inicio Tutoriales JavaBean en un servicio web Axis2



DESARROLLADO POR:
Víctor Javier Madrid

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática con especialidad en Ingeniería del Conocimiento *.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Fecha de publicación del tutorial: 2010-07-28 24

Share | Regístrate para votar

Crear una estructura compleja del tipo JavaBean en un servicio web Axis2

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Instalación.
- 4. Crear el Servidor.
- 5. Configuración en SOAP UI.
- 6. Crear el Cliente.
- 7. Conclusiones.

1. Introducción

En este tutorial se pretende enseñar a crear otra de las estructuras complejas que se pueden implementar en los servicios web : los tipos compuestos (En Java serán los JavaBeans)

Por regla general en la programación no solo se limita al uso de los elementos que facilita el lenguaje, sino que en muchos casos se permite la creación de elementos propios.

Un JavaBean es un componente Java que se utiliza para encapsular varios objetos en un único objeto, por asi decirlo haría las veces de una estructura de registro de C, es decir, un tipo que esta compuesto por otro tipos o por tipos simples.

Para que un objeto Java sea considerado un JavaBean requiere cumplir una serie de normas :

- Disponer de un contructor sin argumentos
- Sus atributos deberían de ser accesibles mediente métodos SET / GET
- Seguir las convenciones de nomenclatura estándar (nombre clase, nombres atributos, etc.)

Catálogo de servicios Autentia



Últimas Noticias

Corto sobre Metodologías Ágiles

Comentando el Libro: Piensa, es gratis de Joaquín Lorente.

Problemas de aspecto en AdictosAlTrabajo.com: Refresco de la hoja de estilo.

Comentando el libro: Lucro sucio de Joseph Heath

Nuevo formato de tutoriales podcast con bolígrafo LiveScribe

Histórico de NOTICIAS

Últimos Tutoriales

JEE6, haciéndolo fácil.

- Implementar de serializable

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G50Vseries (Core Duo P8600 2.4GHz, 4GB RAM,320 GB HD).
- Sistema operativo: Windows Vista Ultimate.
- Mozilla Firefox 2.0.0.20
- Apache Tomcat 7.0.0
- Axis2 1.5.1
- Soap UI 3.5

3. Instalacion.

Para la realización de este tutorial ser requiere :

1) Tener instalada servidor de aplicaciones Apache Tomcat (En mi caso uso la versión 7.0)

2) Tener la versión binaria de Axis2 (En mi caso uso la versión 1.5.1) descargada y descomprimida en un directorio

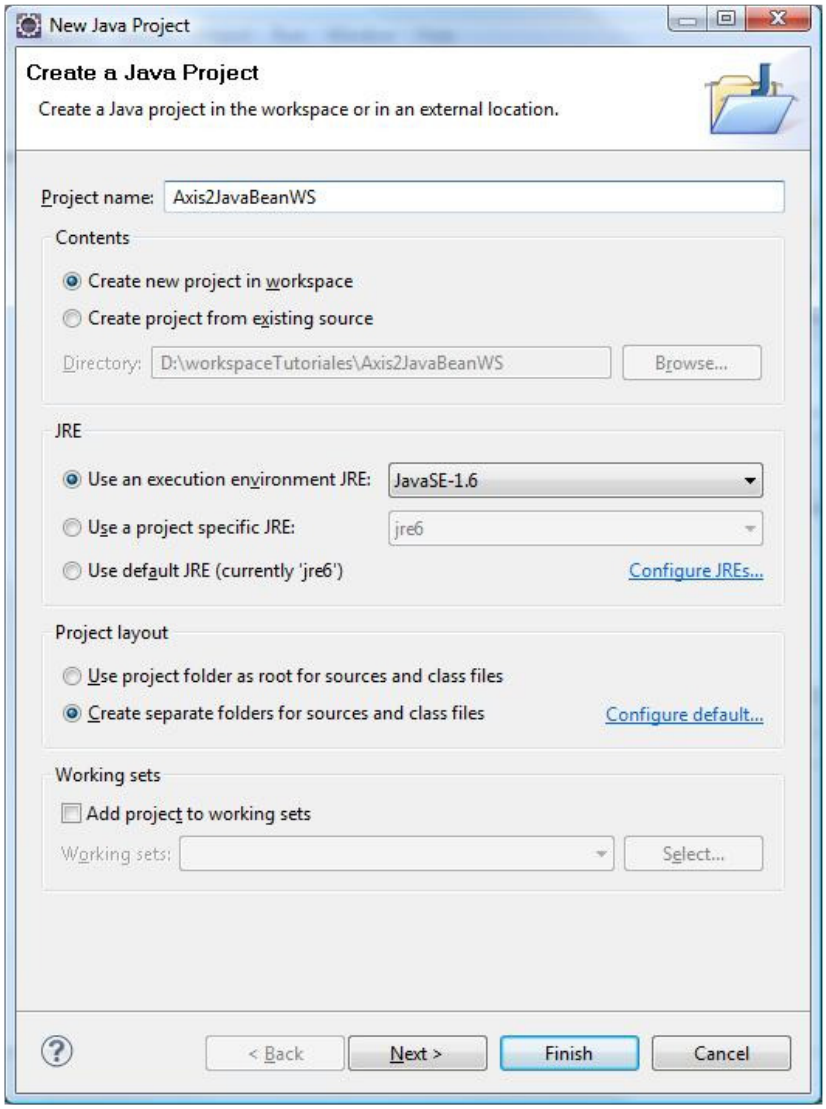
3) Tener la versión WAR de Axis2 (En mi caso uso la versión 1.5.1) desplegada en el Tomcat

4) Tener instalada la herramienta SOAP UI

Nota : Utilizaremos para su desarrollo la herramienta Eclipse

4. Crear el Servidor

PASO 1 : Crear un proyecto Java llamada Axis2JavaBeanWS



Securizar Hudson de una manera sencilla

Envío de correo electrónico con el soporte de Jboss Seam.

Ejemplo de arquitectura propuesta por Autentia

Creación de servicios web RESTful con el soporte de RESTeasy de Jboss Seam.

Últimos Tutoriales del Autor

Crear una estructura compleja del tipo Array en un servicio web Axis2

Apache TCPMON : El Sniffer de los Servicios Web

Introducción a Activiti

Introducción al gestor de base de datos H2

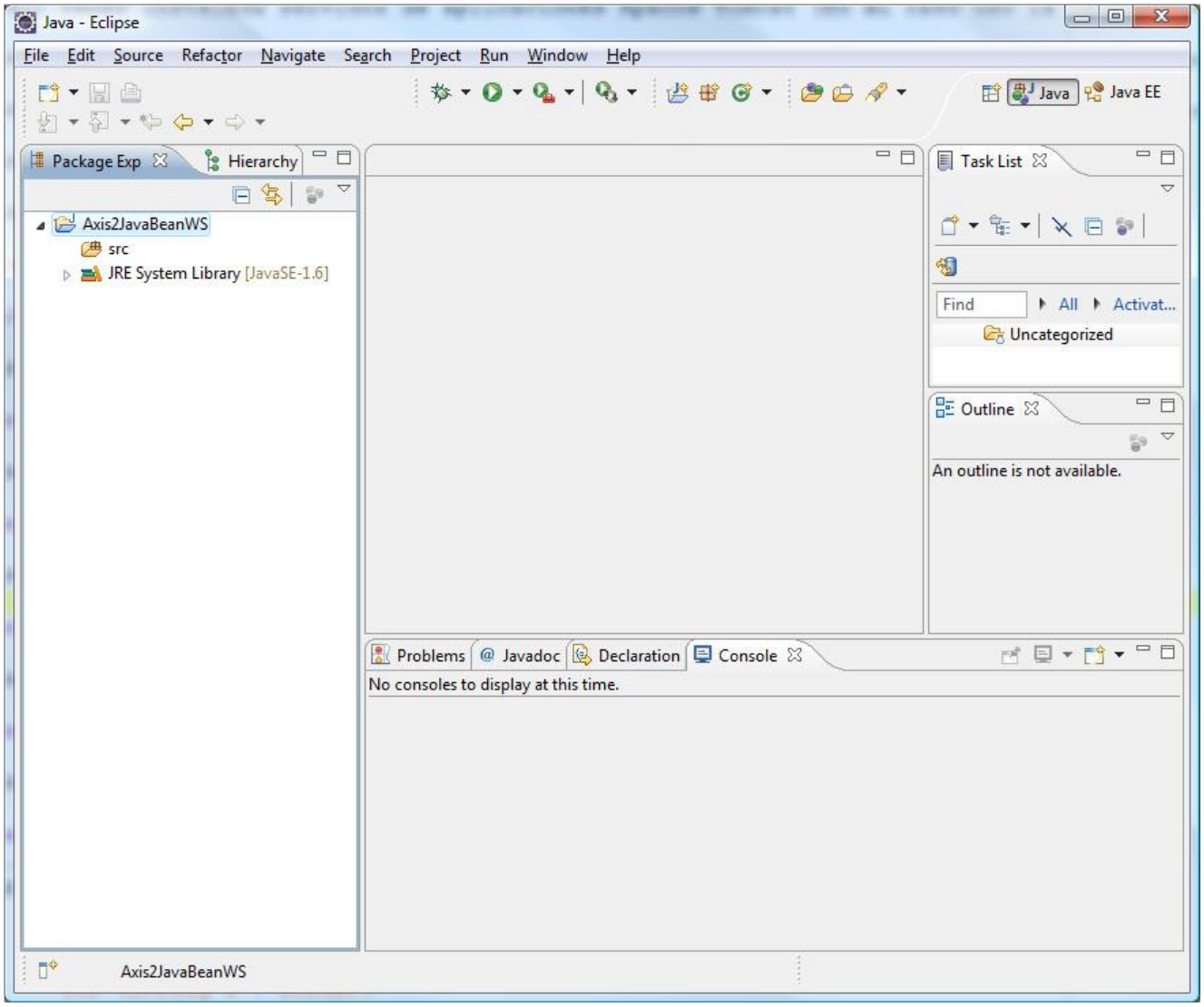
Exportación de trabajos en Talend

Síguenos a través de:

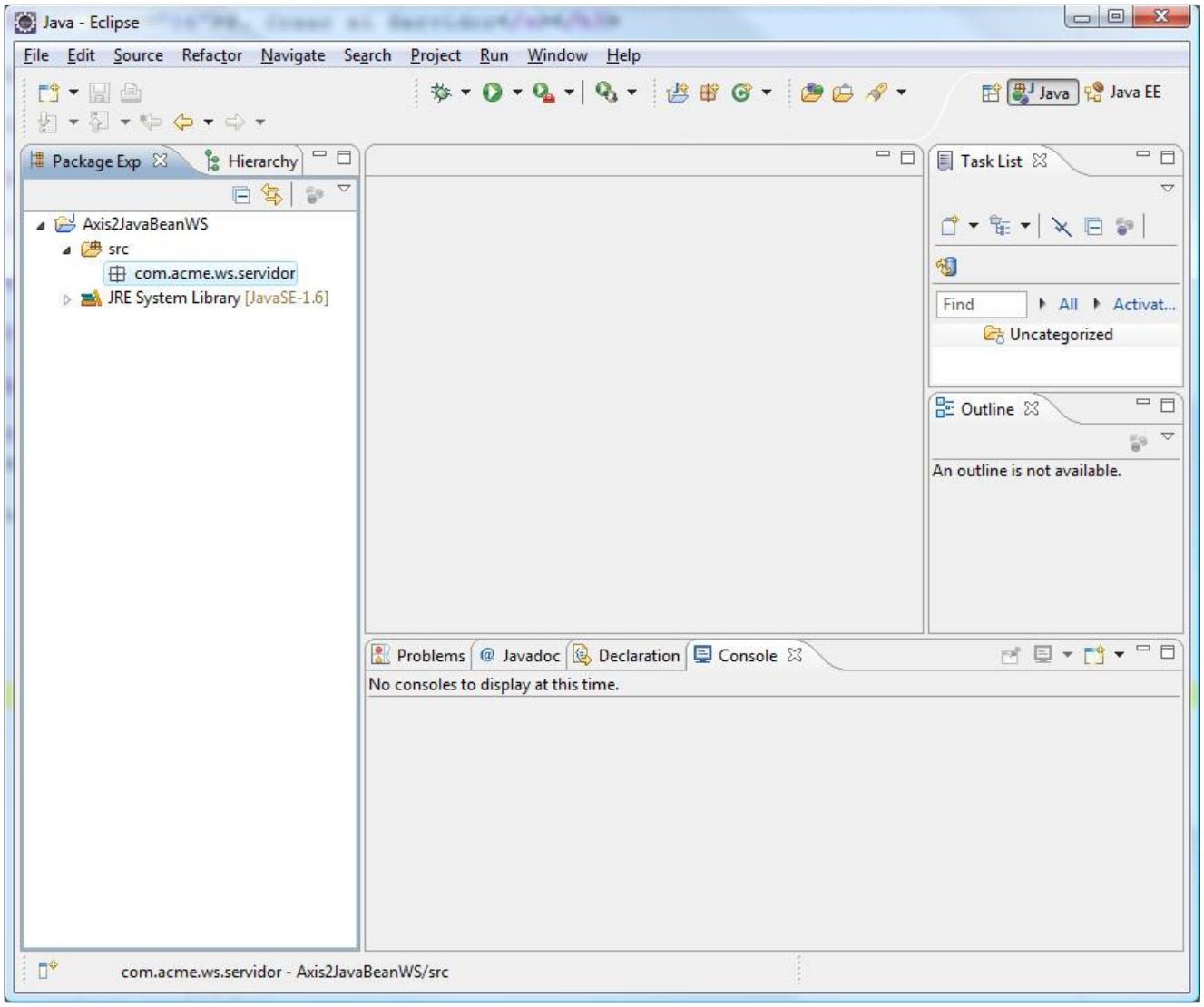
Últimas ofertas de empleo

2010-06-25
T. Información - Analista / Programador - BARCELONA.

Se creará una estructura del proyecto como la siguiente



PASO 2 : Crear el paquete "com.acme.ws.servidor" sobre la carpeta src

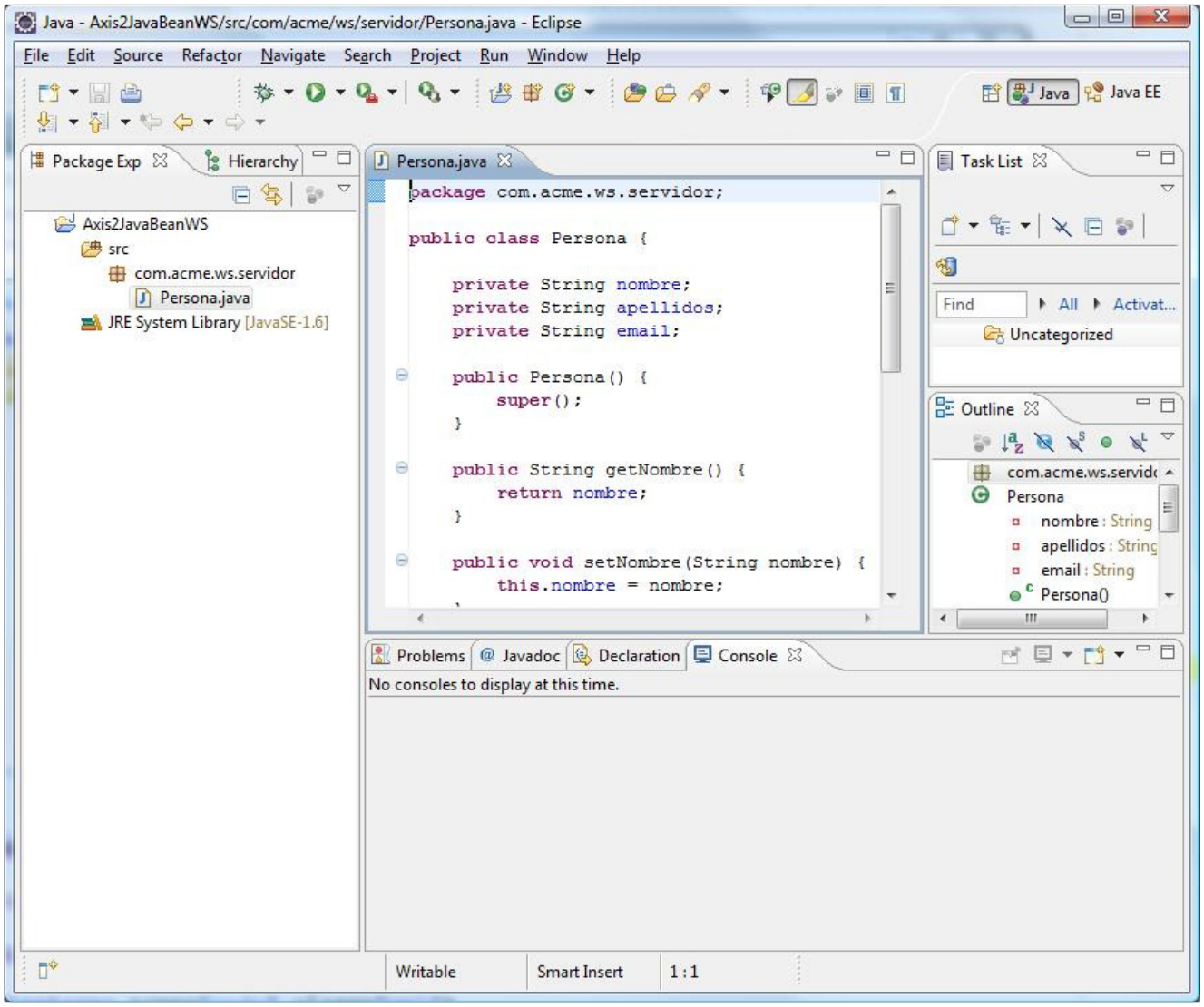


PASO 3 : Crear una clase Java en el paquete anterior llamada Persona que será el JavaBean

Crear una clase con 3 atributos : nombre, apellidos y email.

```
package
com.acme.ws.servidor
```

Nuestro proyecto quedaría asi :



PASO 4 : Crear una clase Java llamada BasicService en el paquete anterior

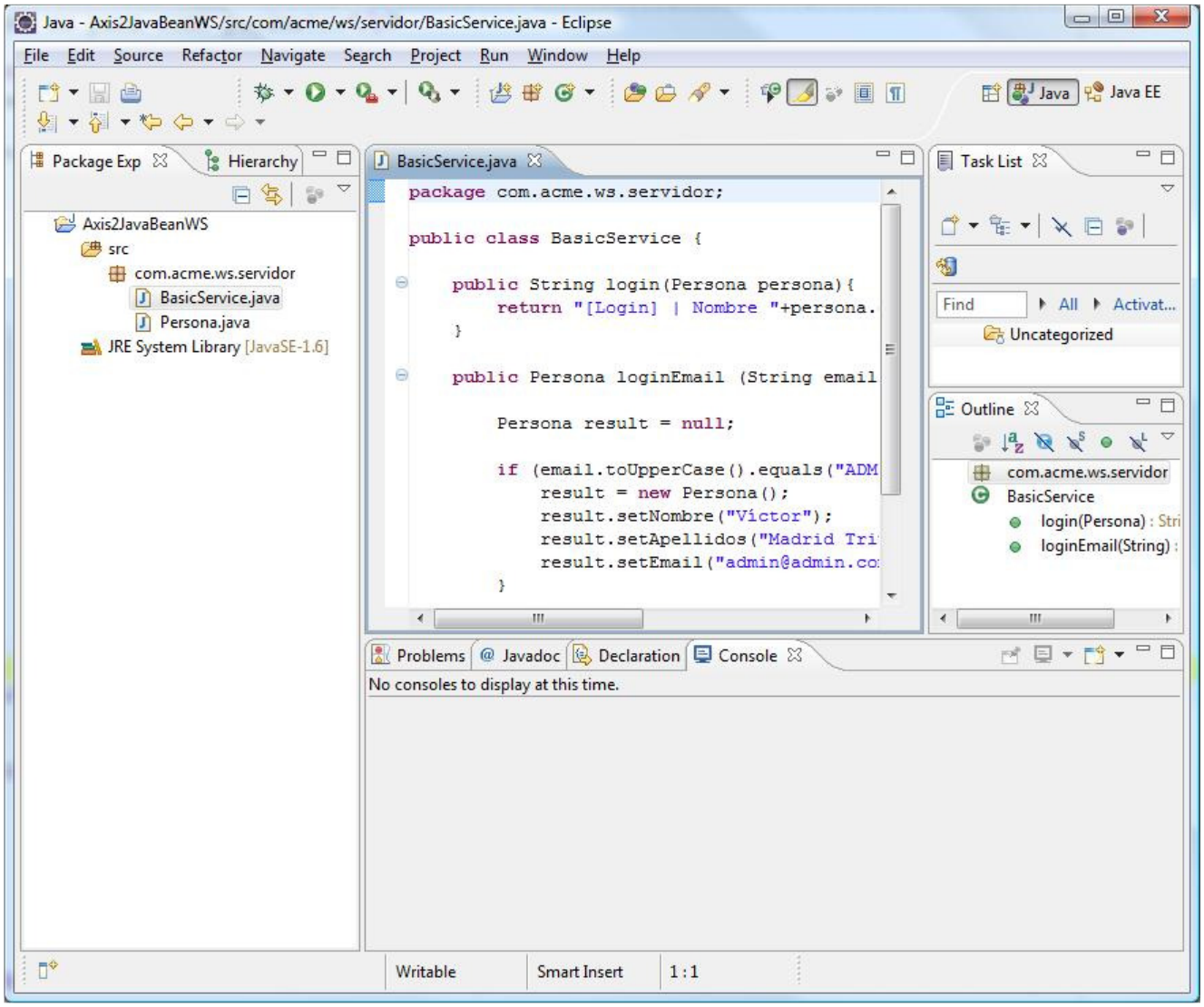
En este caso se definirán dos métodos que serán los servicios disponibles para los clientes.

En el método "login" se devolverá una cadena con los atributos de nombre y email indicando como que se ha autenticado sin probemas.

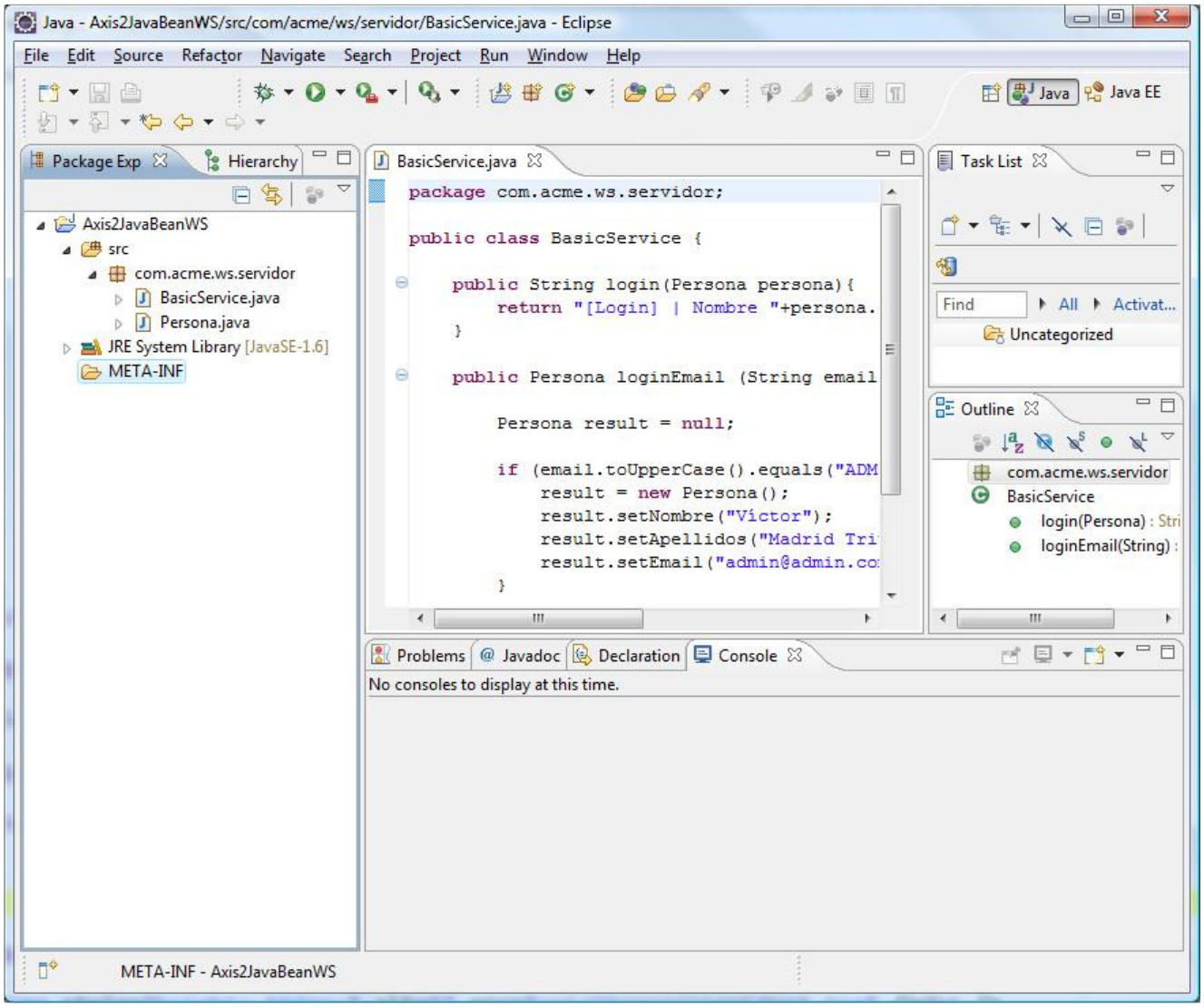
En el método "loginEmail" se devolverá un JavaBean que se inicializará si y solamente si coincide el email que se pasa como parámetro

```
package
com.acme.ws.servidor
```

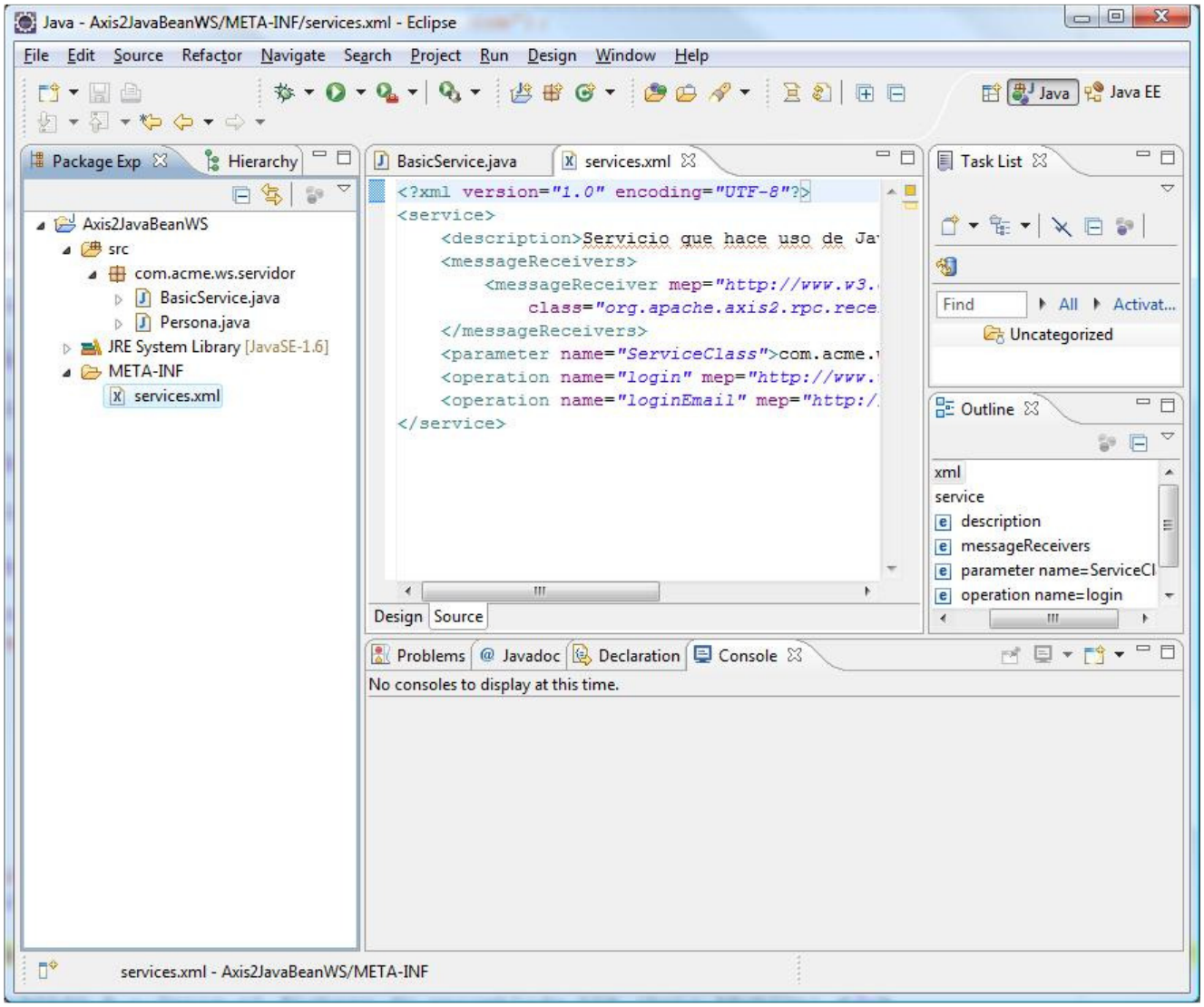
Nuestro proyecto quedaría asi :



PASO 5 : Crear la carpeta META-INF a nivel de la carpeta src



PASO 6 : Crear dentro de la carpeta META-INF el descriptor de servicio : services.xml



Introducimos el siguiente texto :

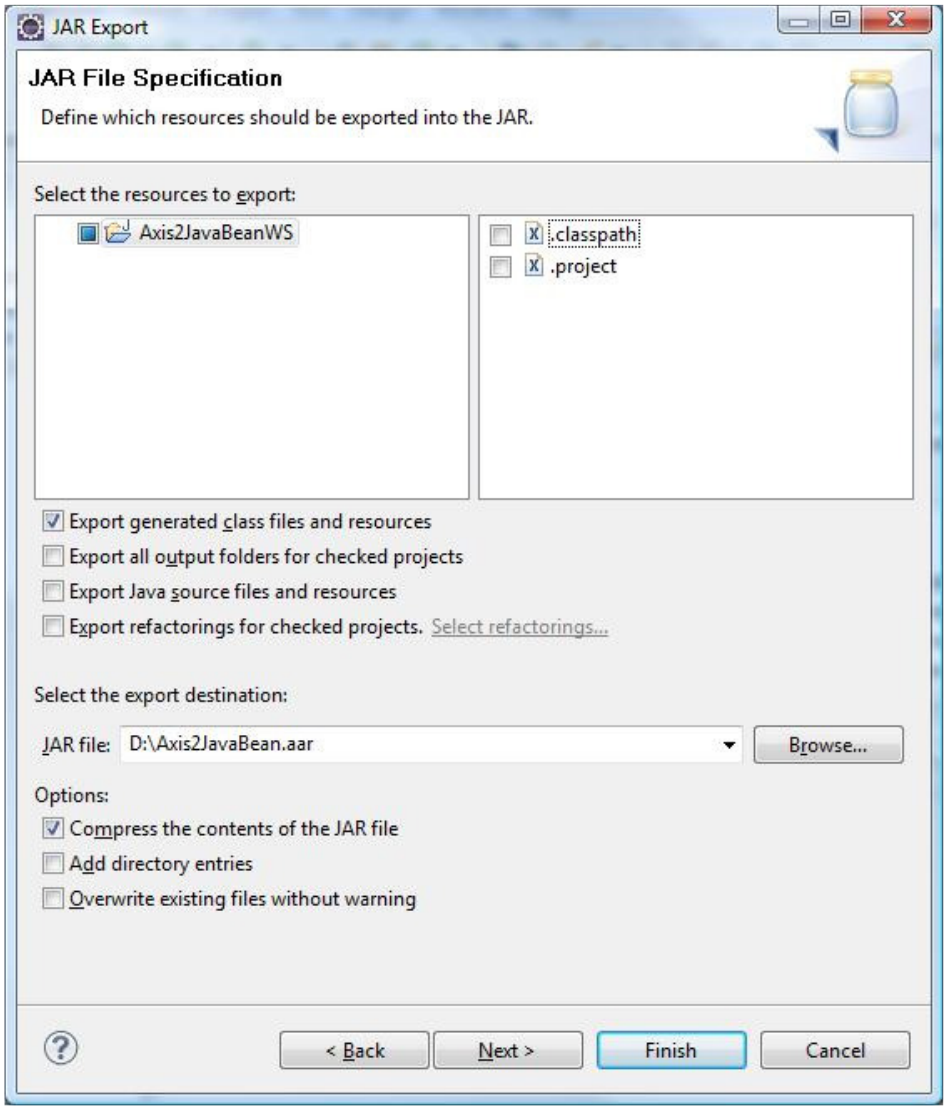
```
<service>
```

Nota : Recordar añadir la línea inicial : <?xml version="1.0" encoding="UTF-8"?>

PASO 8 : Crear el fichero de ensamblado AAR (Axis ARchive)

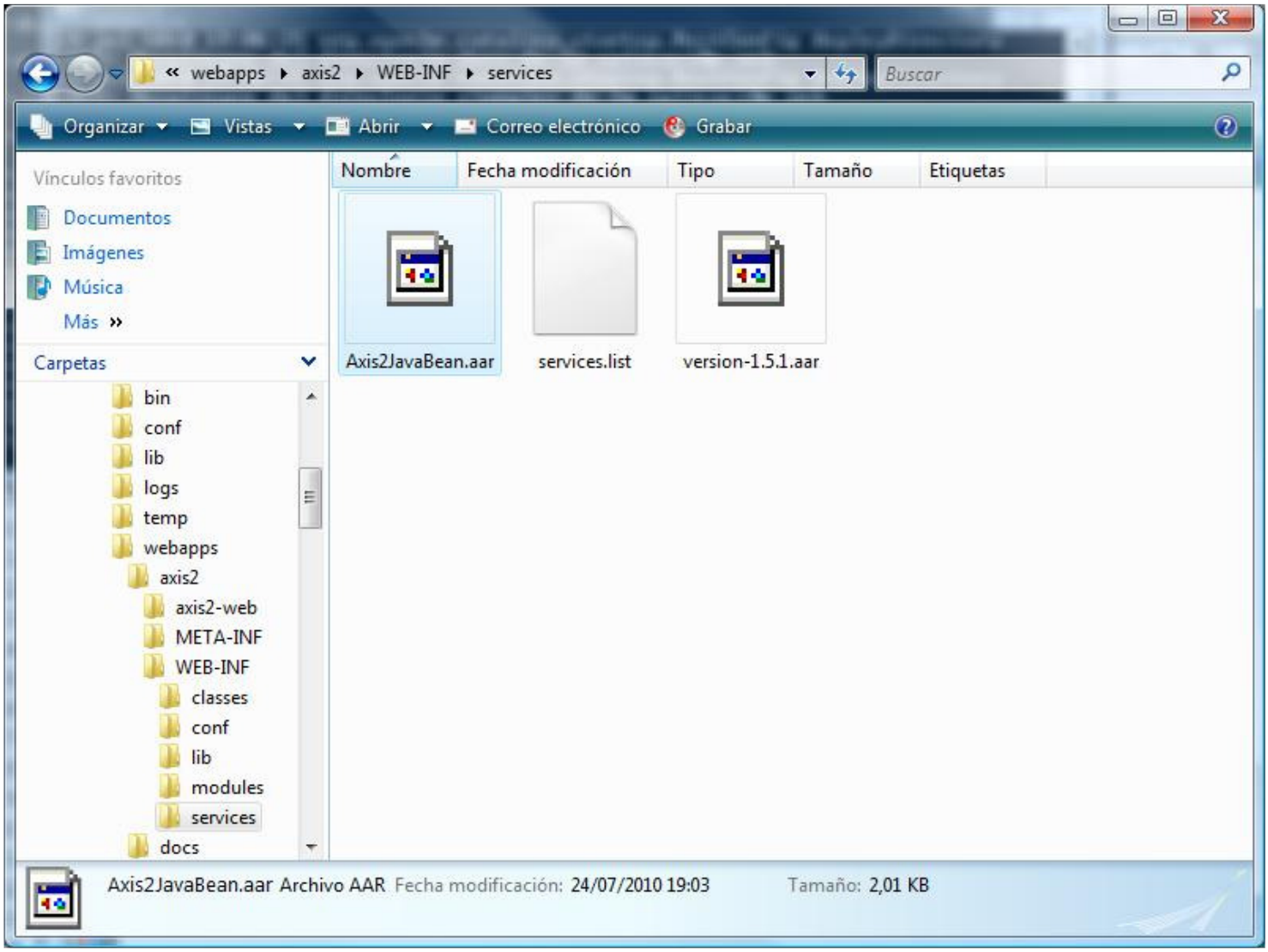
Para ello exportar el proyecto como si fuera un JAR pero poniendo de nombre Axis2JavaBean.aar

Nota : No incluir .classpath ni .project

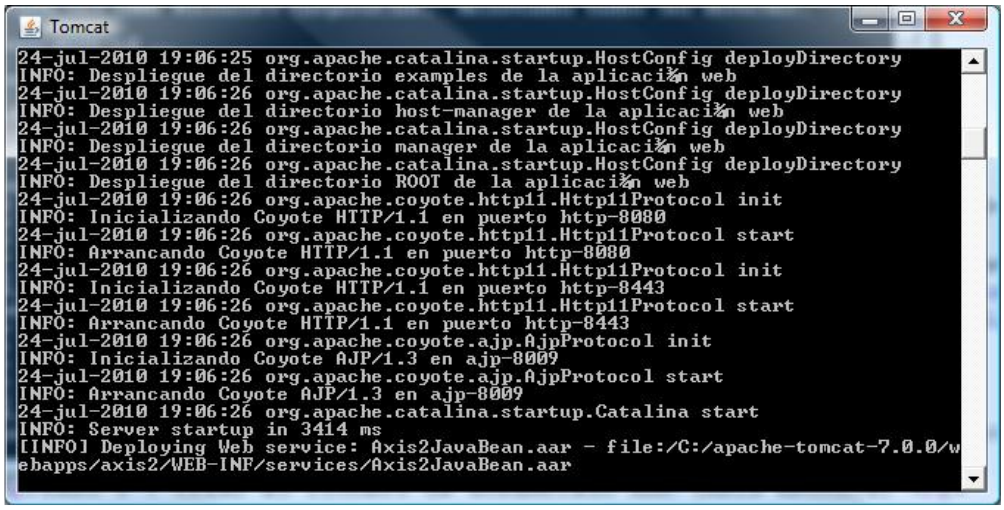


Arrancar Tomcat si esta parado

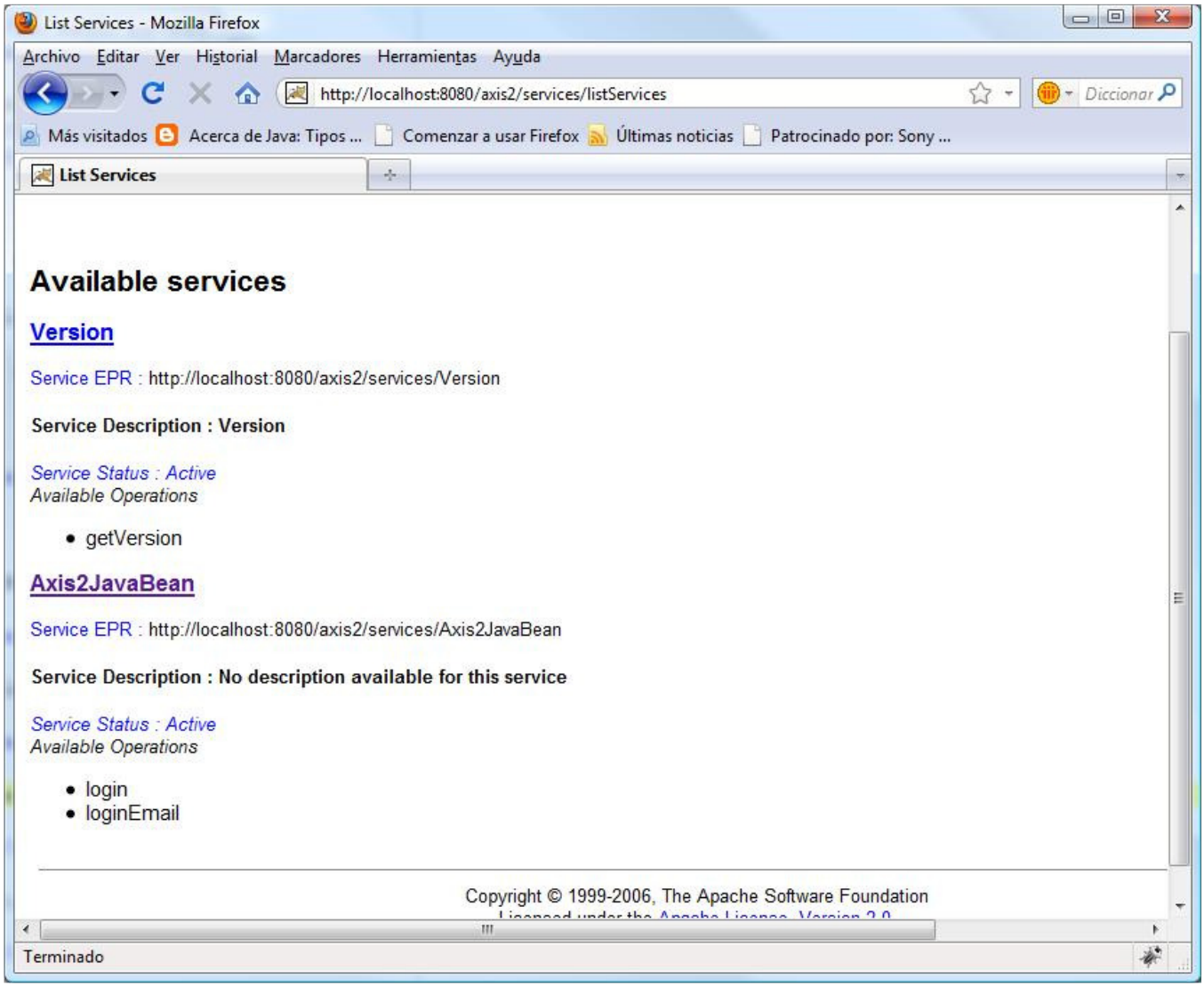
PASO 9 : Copiar el fichero AAR en <CATALINA_HOME>\webapps\axis2\WEB-INF\services



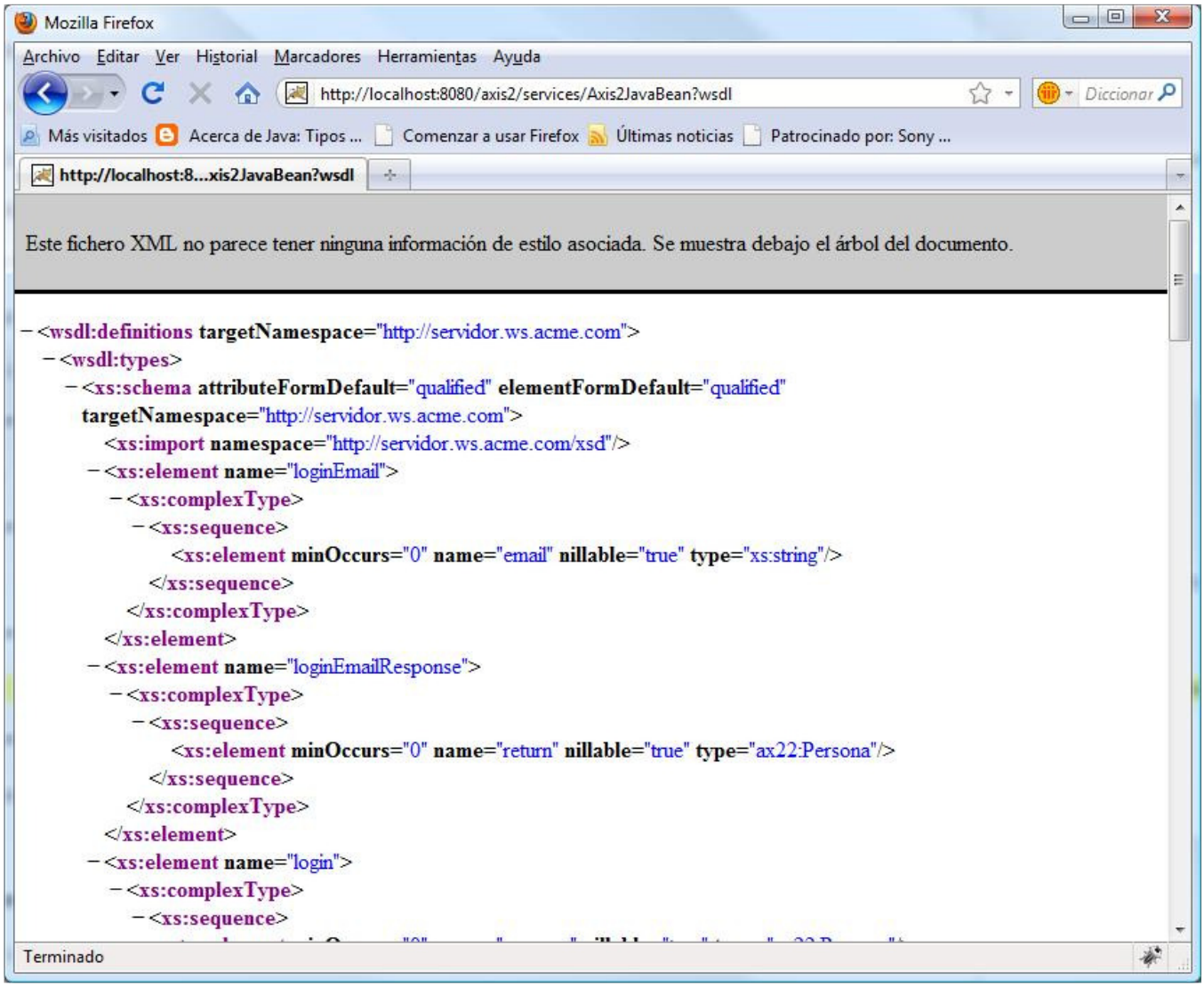
PASO 10 : Verificar en la consola de Tomcat la inclusión de este servicio



PASO 11 : Verificar el servicio desde la consola de administracion de Axis : http://localhost:8080/axis2/services/listServices



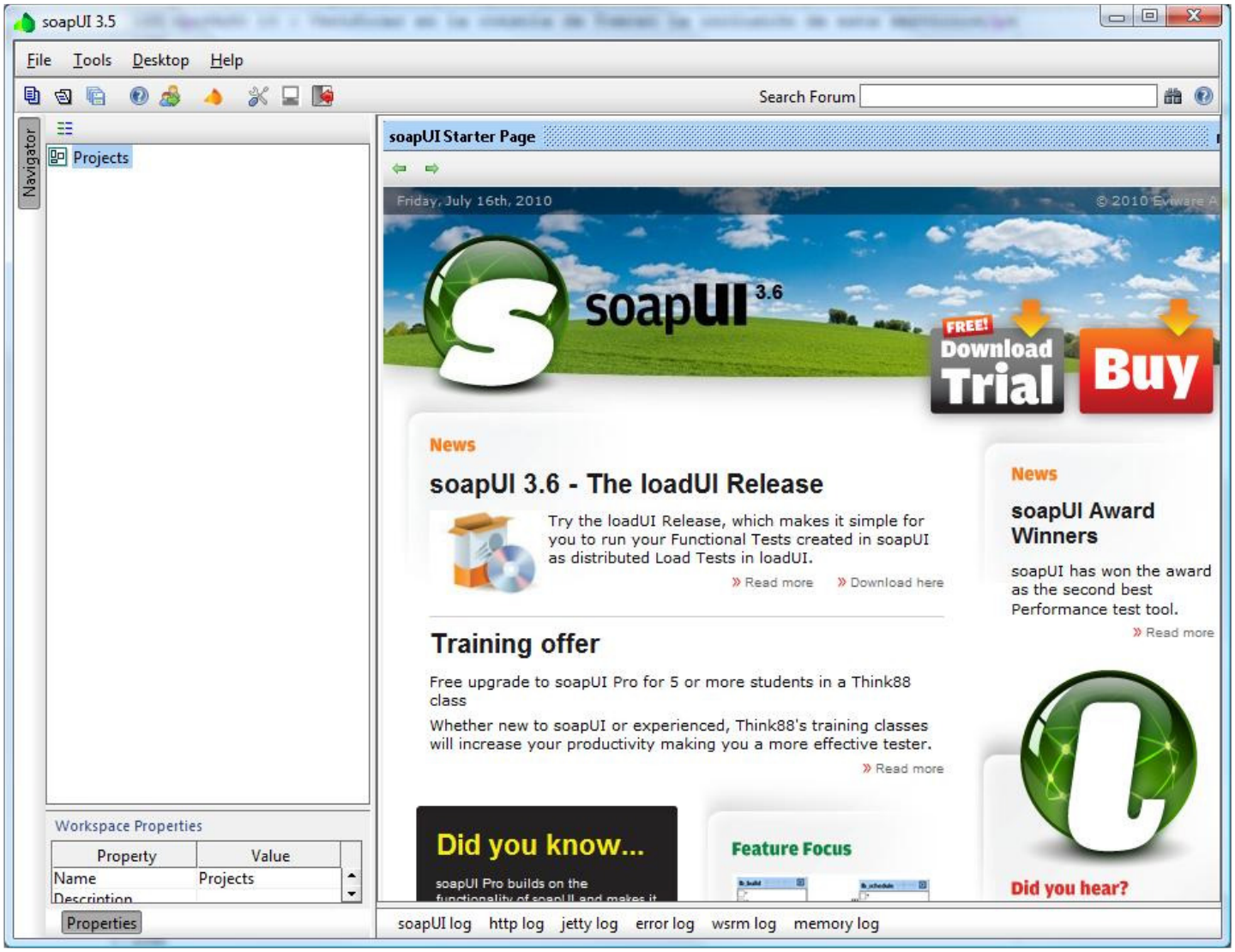
PASO 12 : Obtenemos el WSDL del servicio : http://localhost:8080/axis2/services/Axis2JavaBean?wsdl



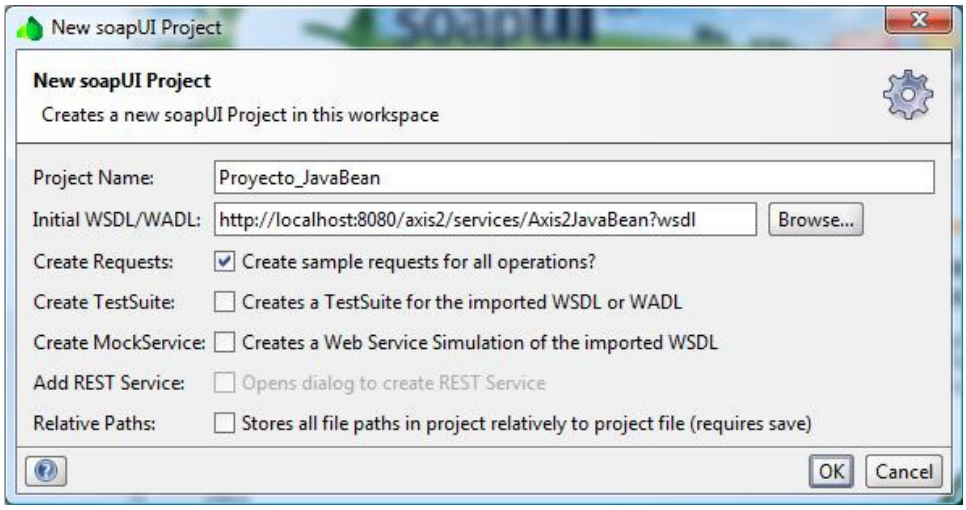
Guardamos el fichero anterior que vendrá dado en formato XML por formato WSDL

5. Probar el servidor con SOAP UI

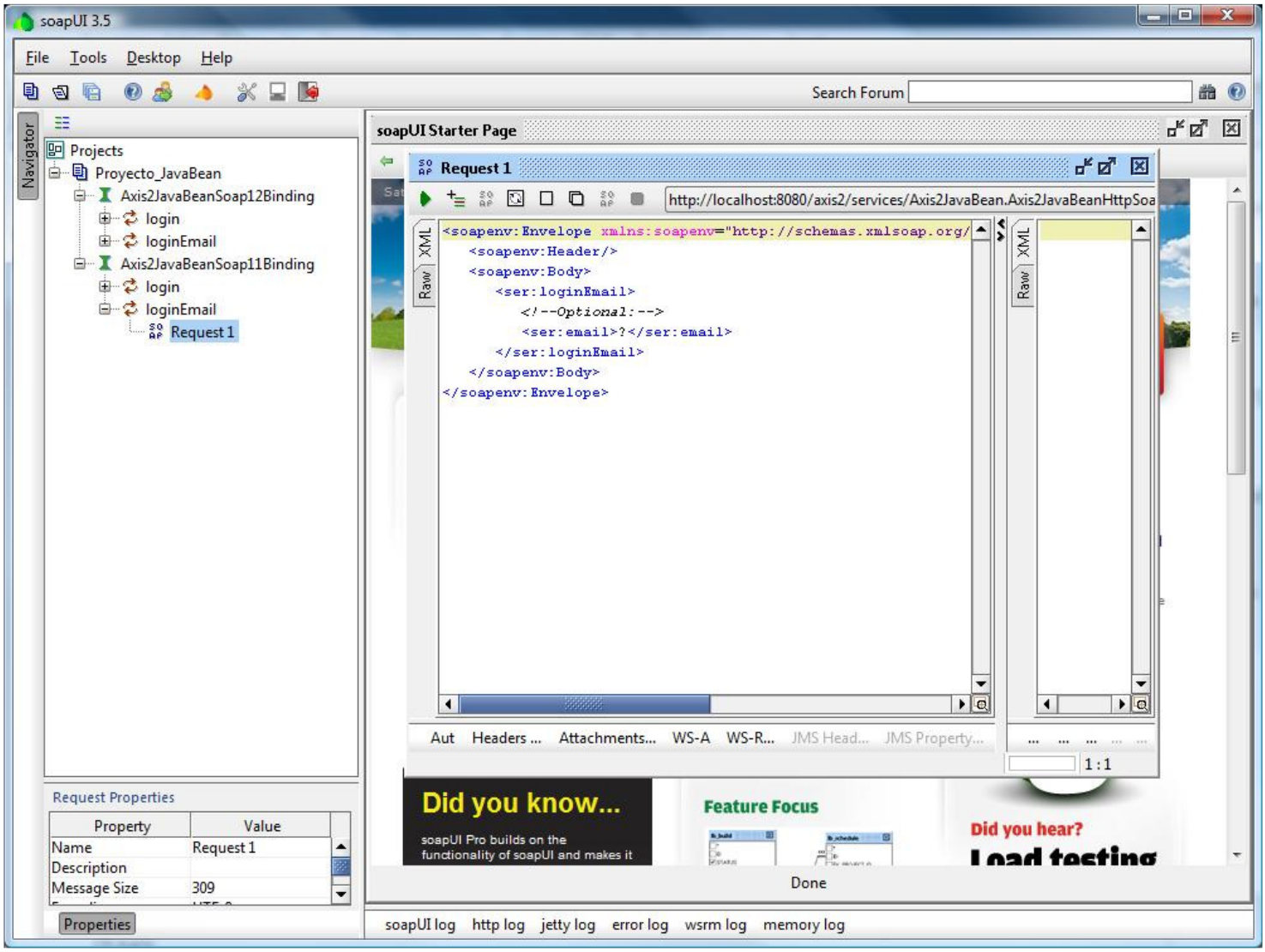
PASO 1 : Cargamos la herramienta



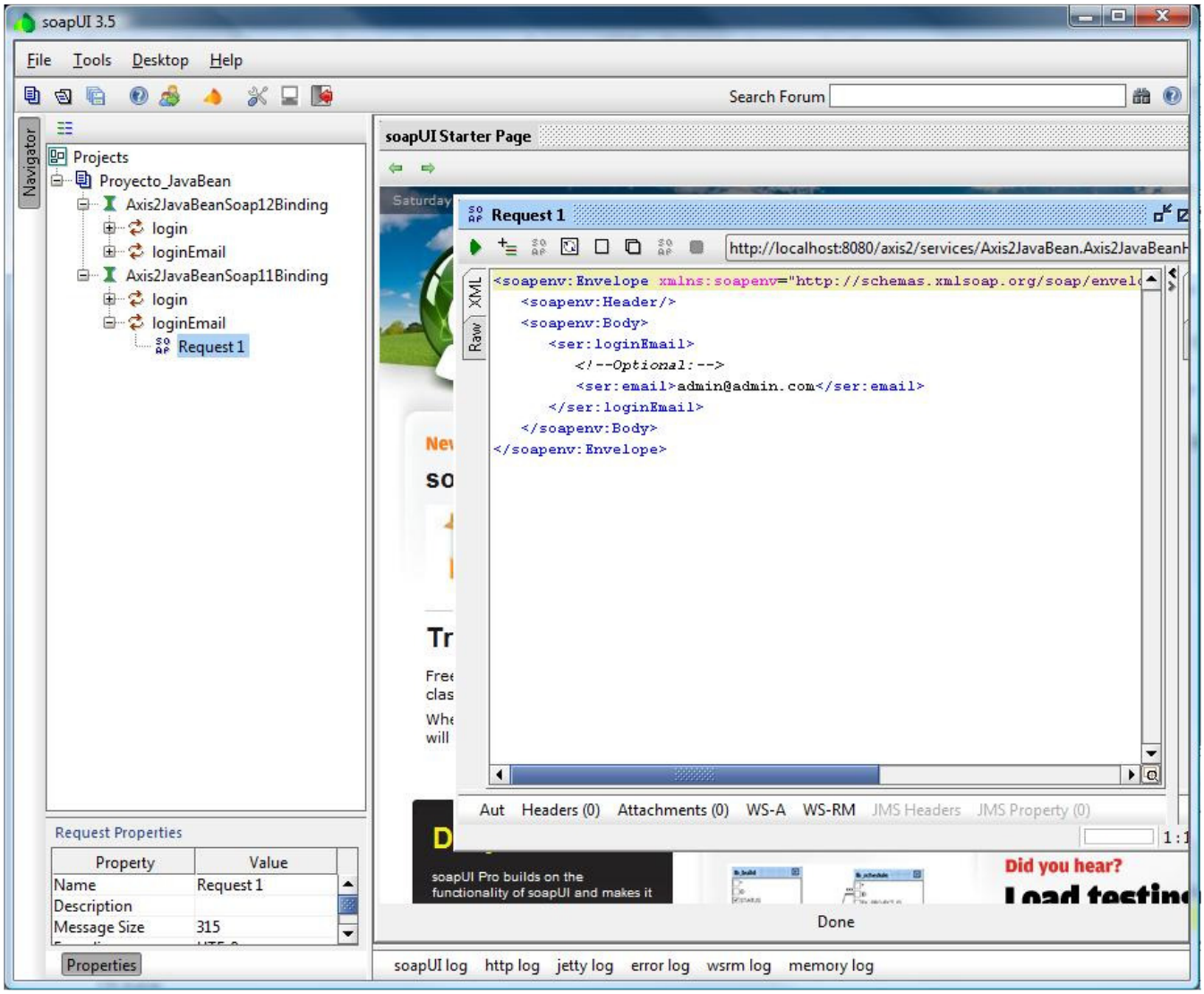
PASO 2 : Creamos un proyecto nuevo con la URL del WSDL anterior



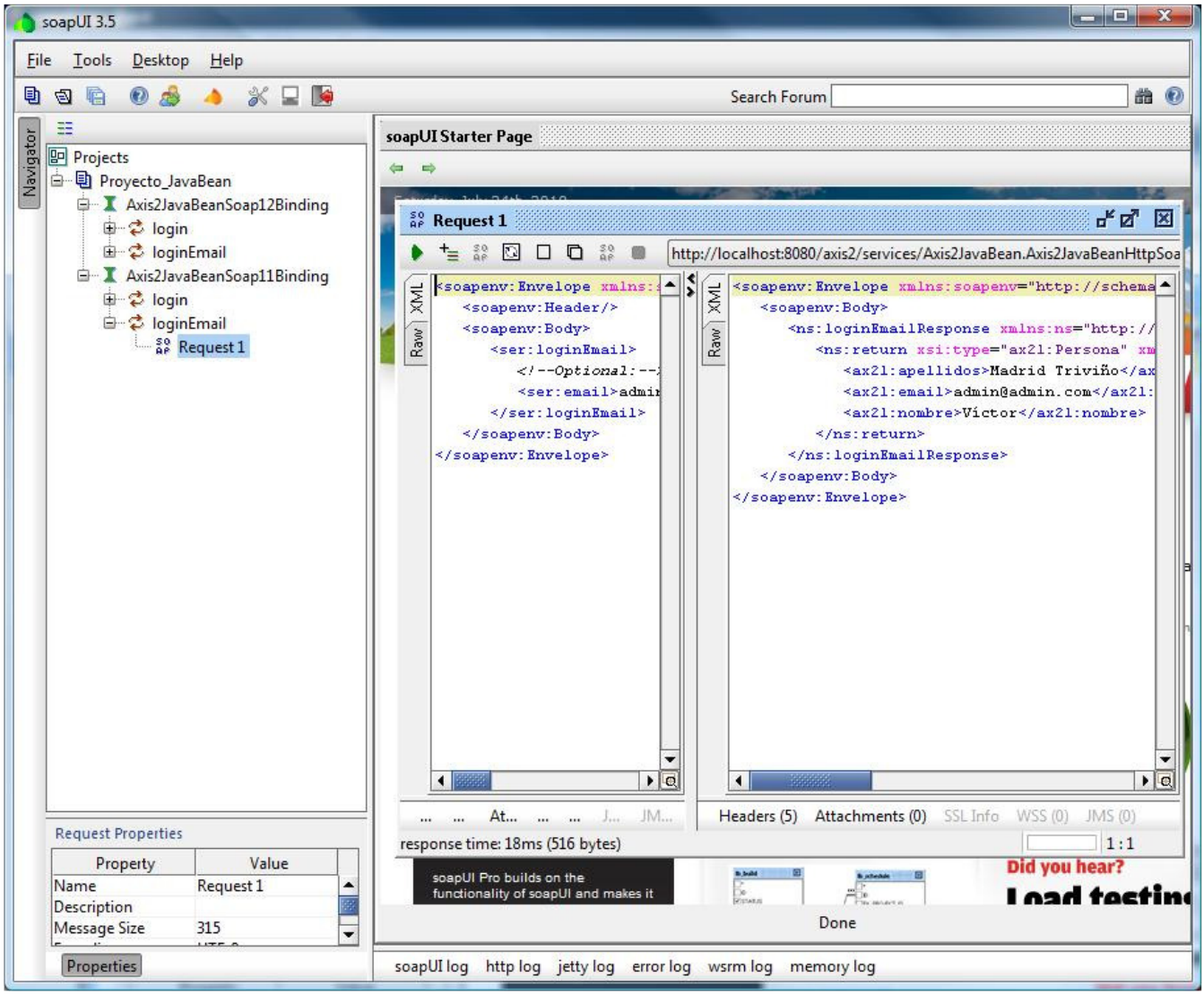
PASO 3 : Realizamos un test sobre el método "loginEmail"



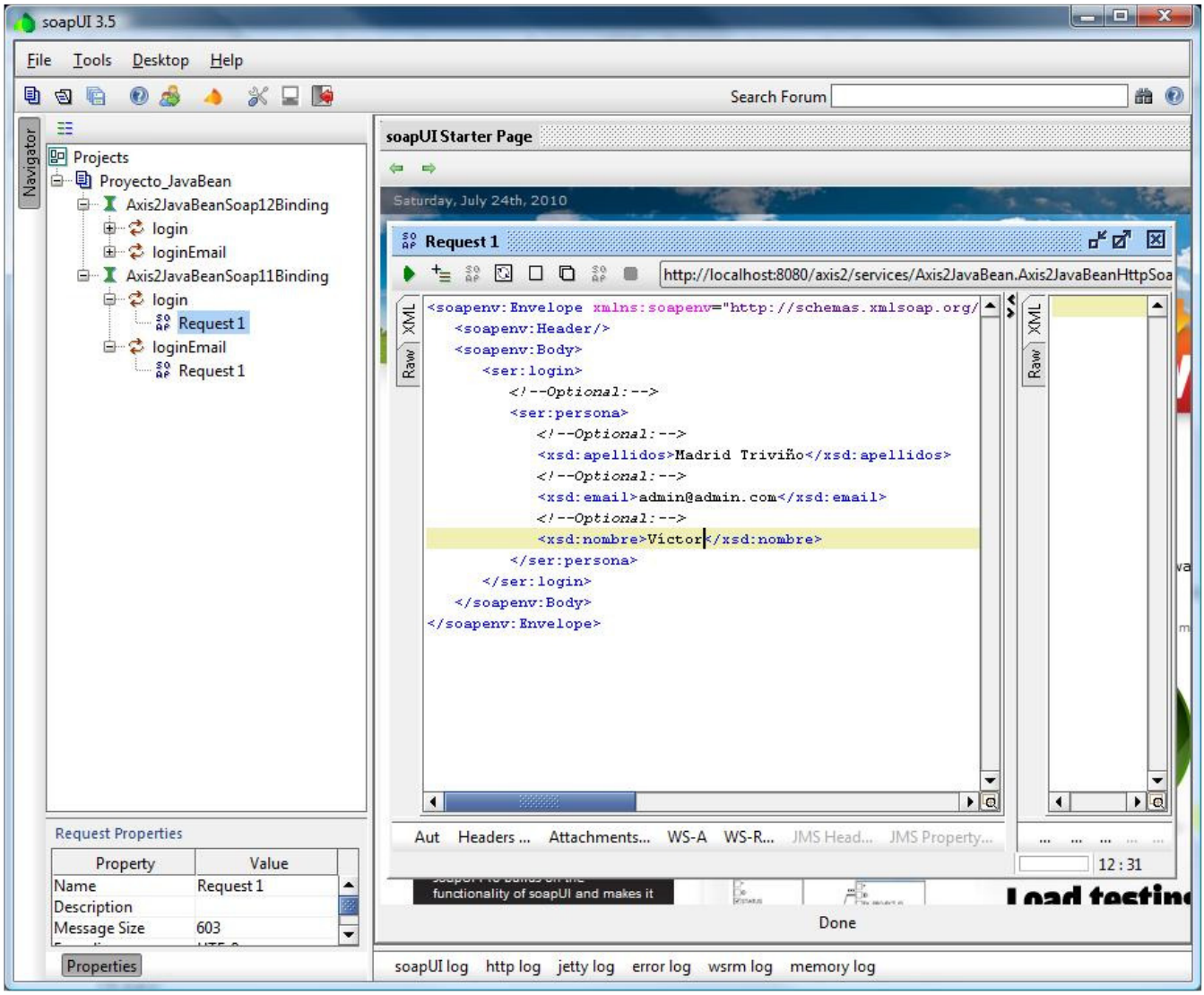
Para ello introducimos el siguiente email que será reconocido por el servidor :
<ser:email>admin@admin.com</ser:email>



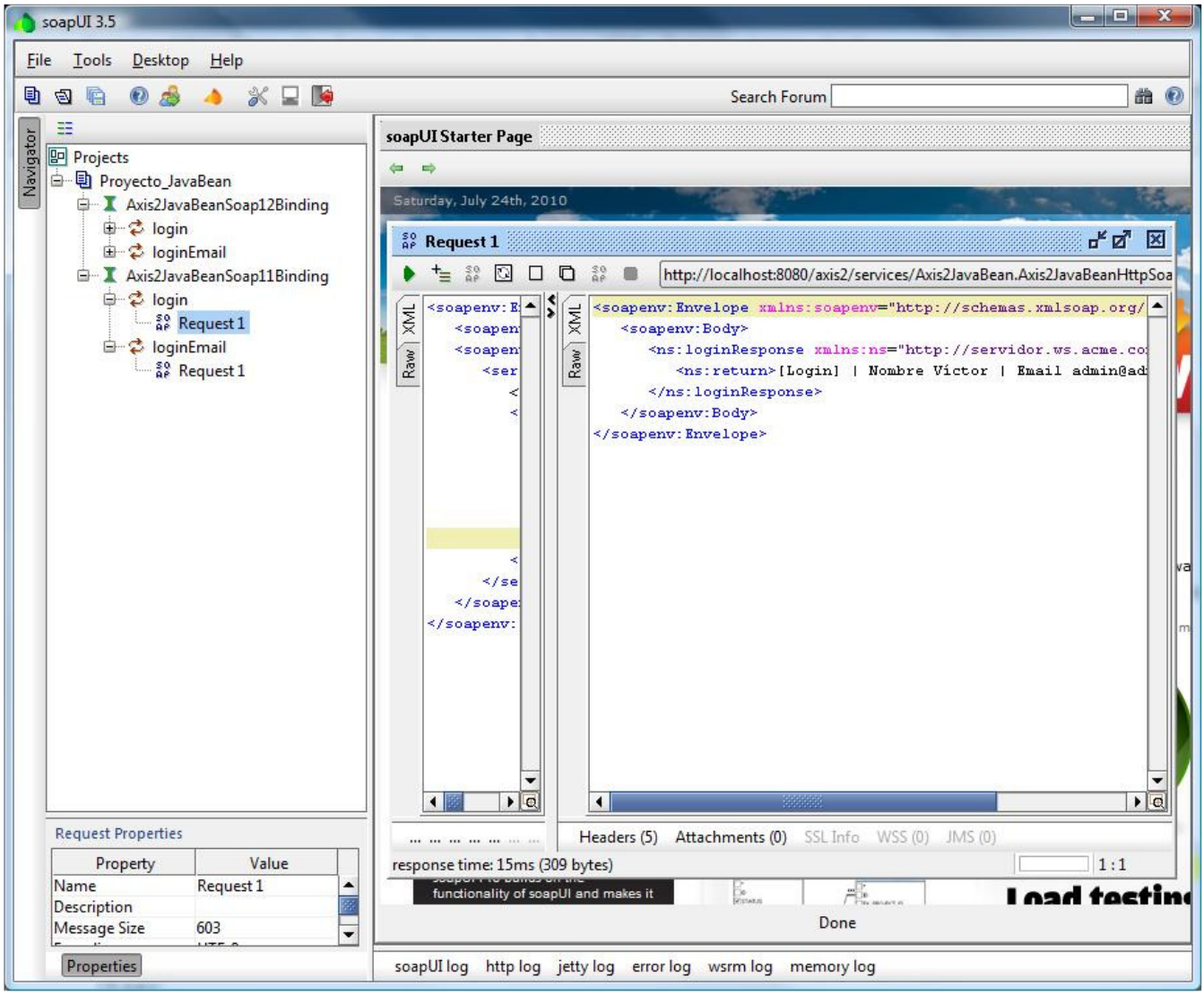
PASO 4 : Ejecutamos el test y verificamos la respuesta



PASO 5 : Realizamos un test sobre el método "login"



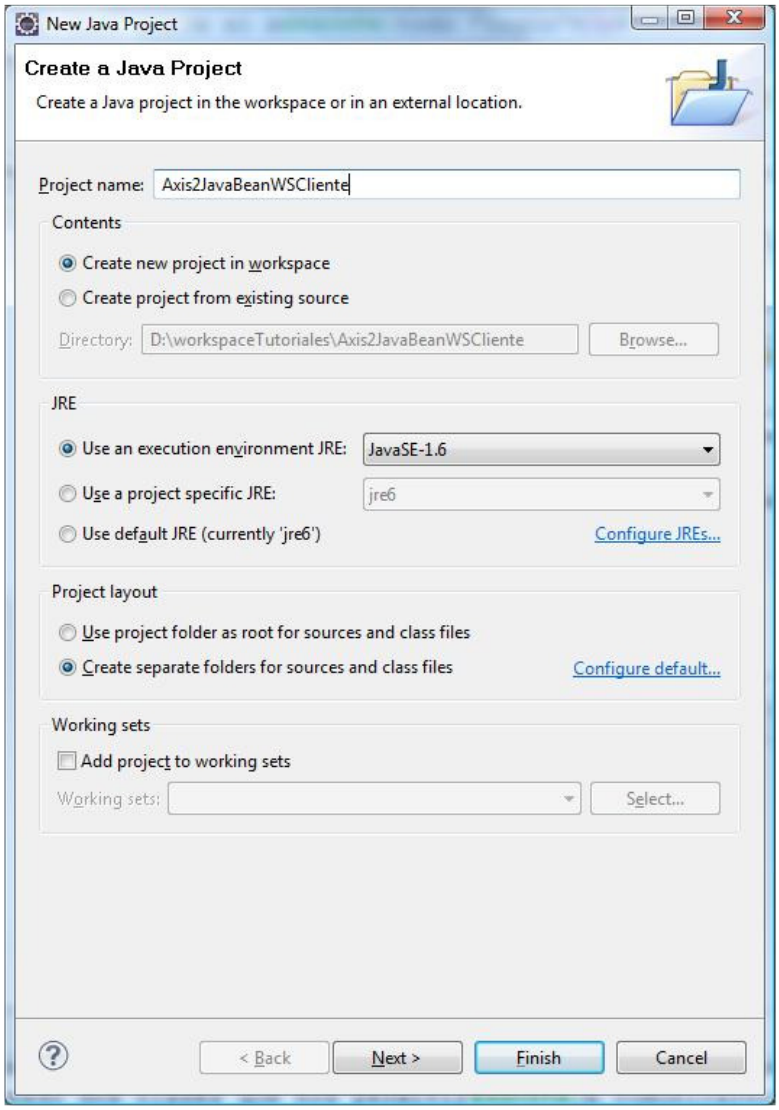
PASO 6 : Ejecutamos el test y verificamos la respuesta



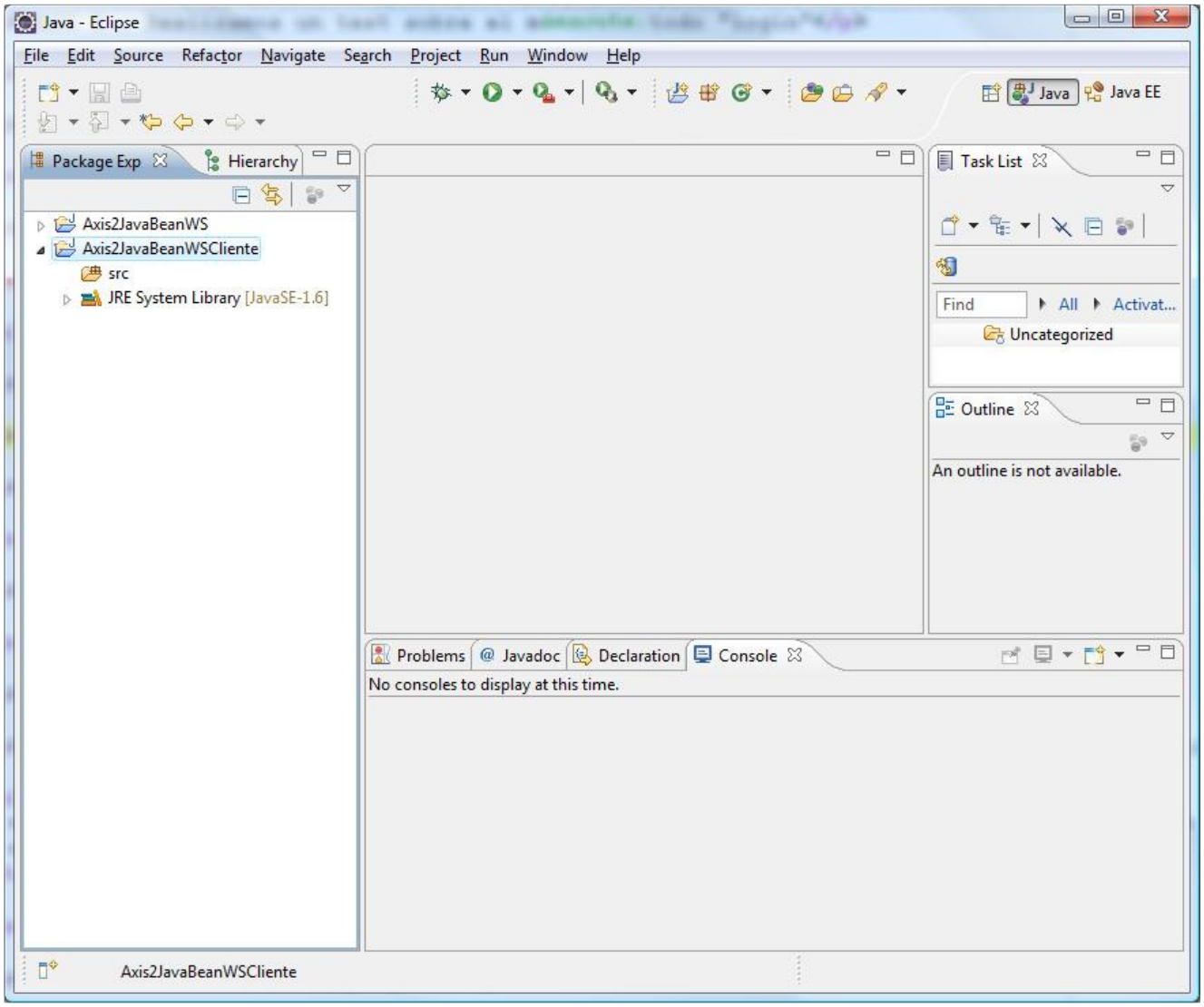
Con esto tendremos probado nuestro Servidor ;-)

6. Crear el cliente

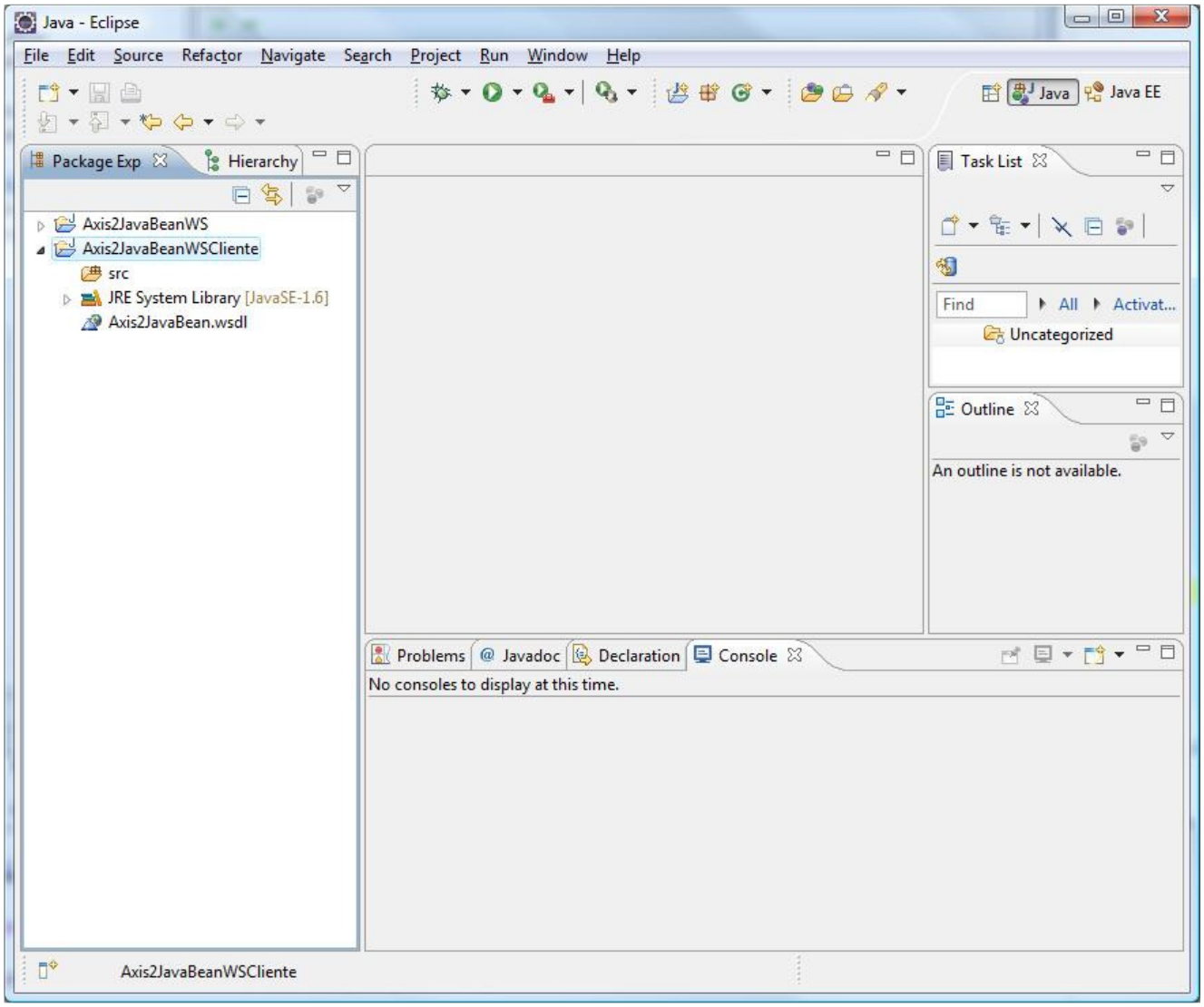
PASO 1 : Creamos un proyecto Java llamada Axis2JavaBeanWSCliente



Se generará una estructura similar a la siguiente



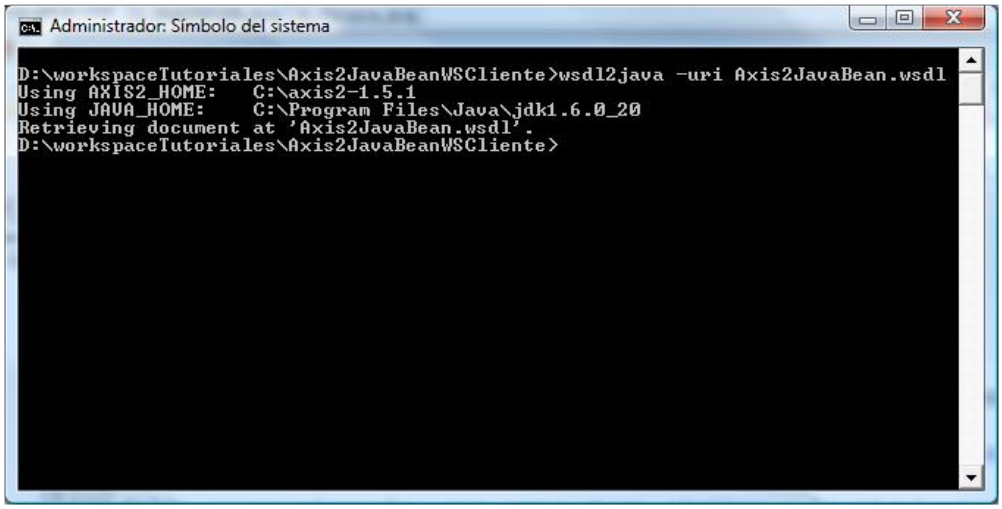
PASO 2 : Copiamos a nivel de raiz el fichero Axis2JavaBean.wsdl



PASO 3 : Generamos el Stub del cliente con la herramienta Axis2

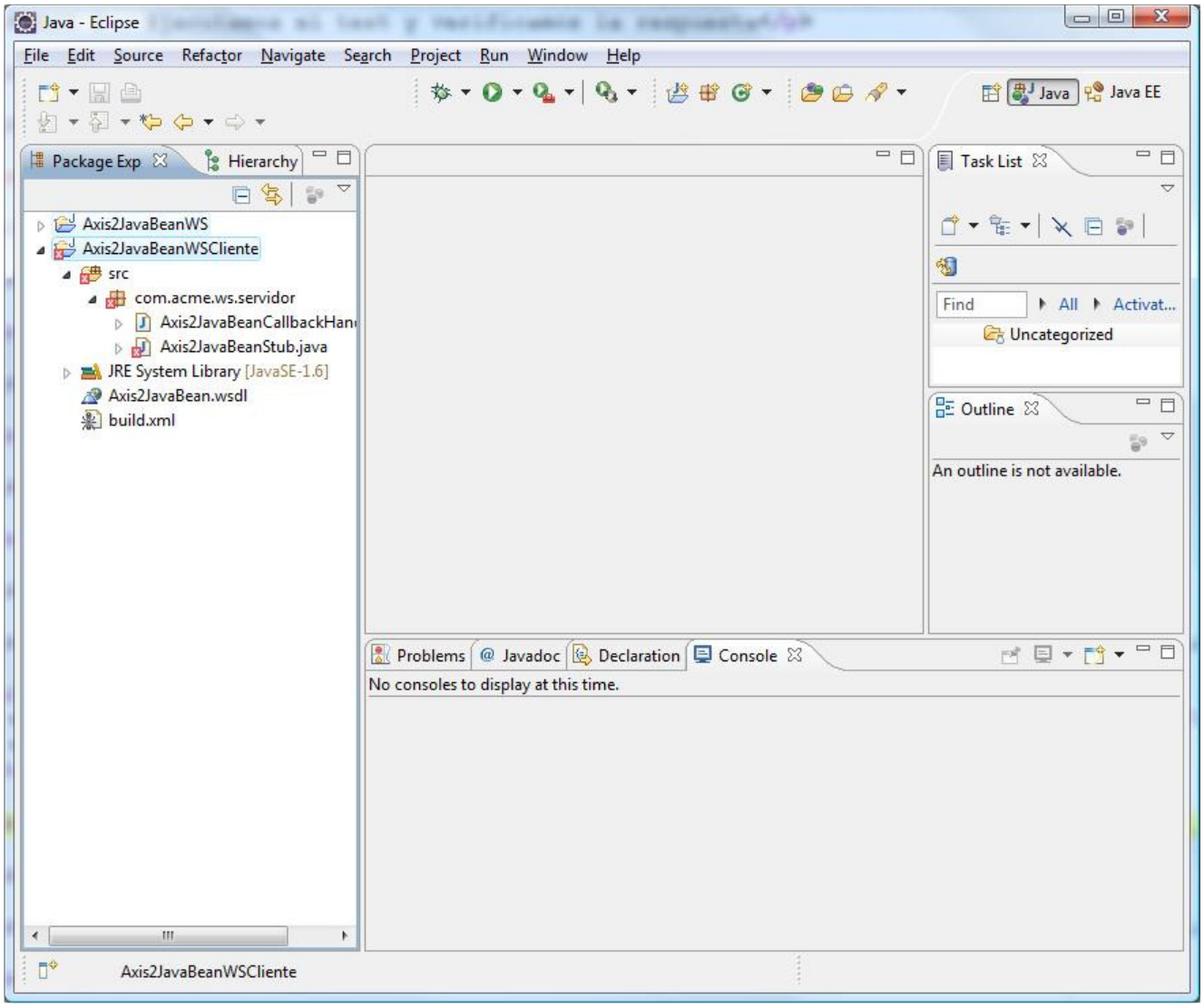
Para ellos nos situamos por consola en el directorio del proyecto y ejecutamos la siguiente instrucción

- Si Axis2 NO esta en el Path : %AXIS2_HOME%\bin\wsdl2java -uri Axis2JavaBean.wsdl
- Si Axis2 esta en el Path : wsdl2java -uri Axis2JavaBean.wsdl



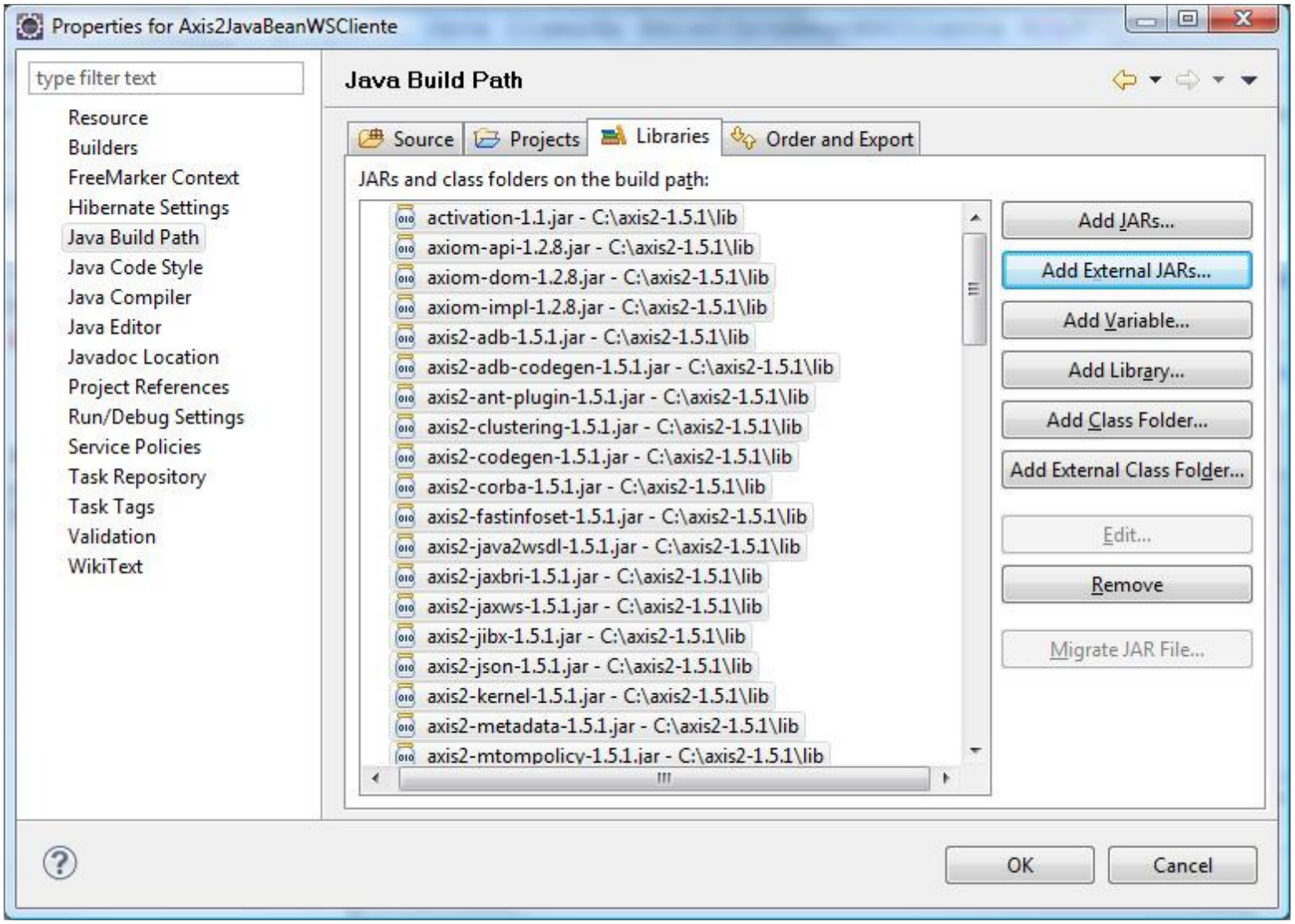
PASO 4 : Refrescamos el proyecto

Se habrán creado dos clases que nos permitirán comunicarnos con el servidor

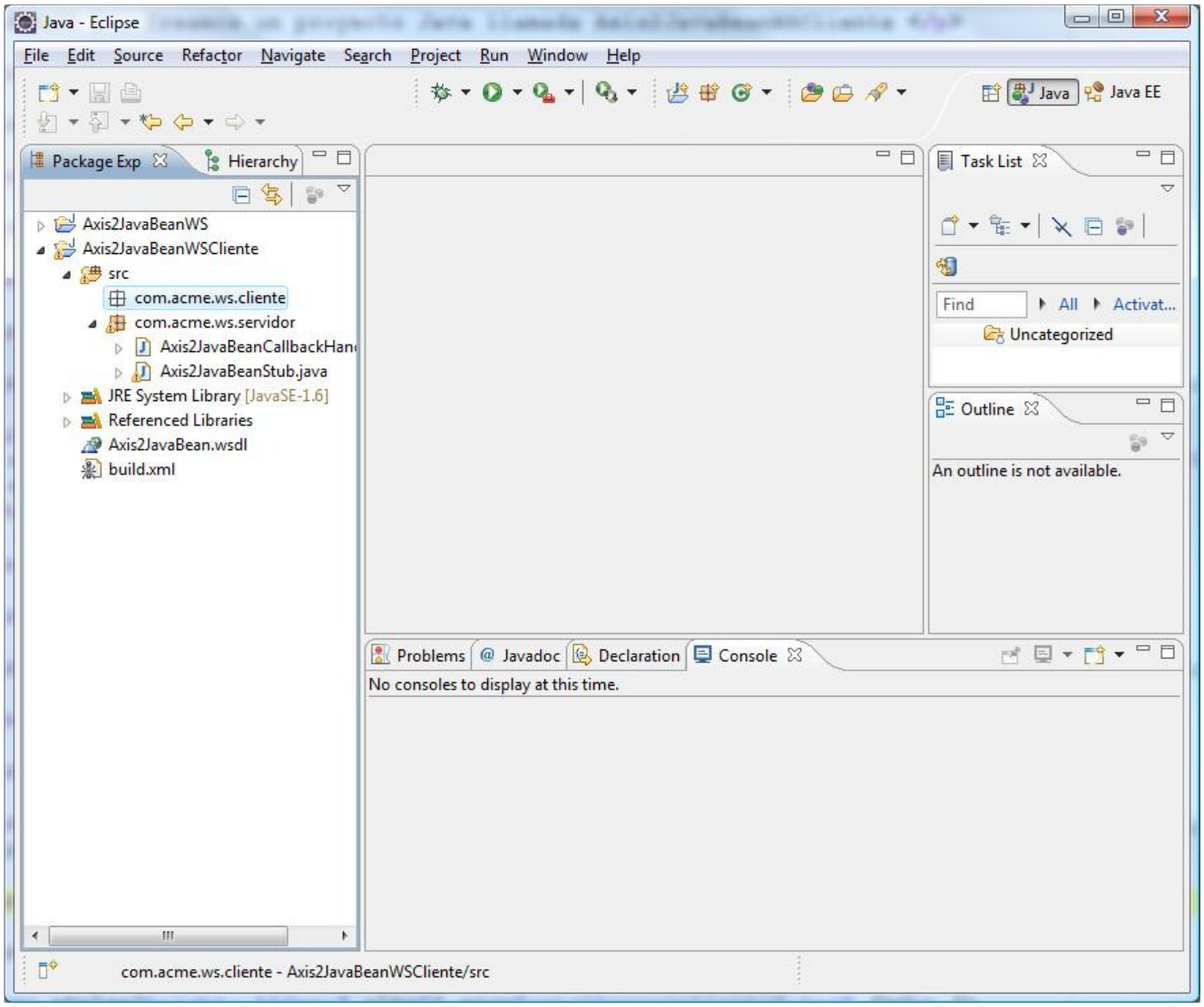


PASO 5 : Incluir las librerías de Axis2

Con el botón derecho del ratón sobre el proyecto Properties -> Java Build Path , pestaña de Libraries y pulsar sobre "Add Externar JARs"



PASO 6 : Crear un paquete "com.acme.ws.cliente"

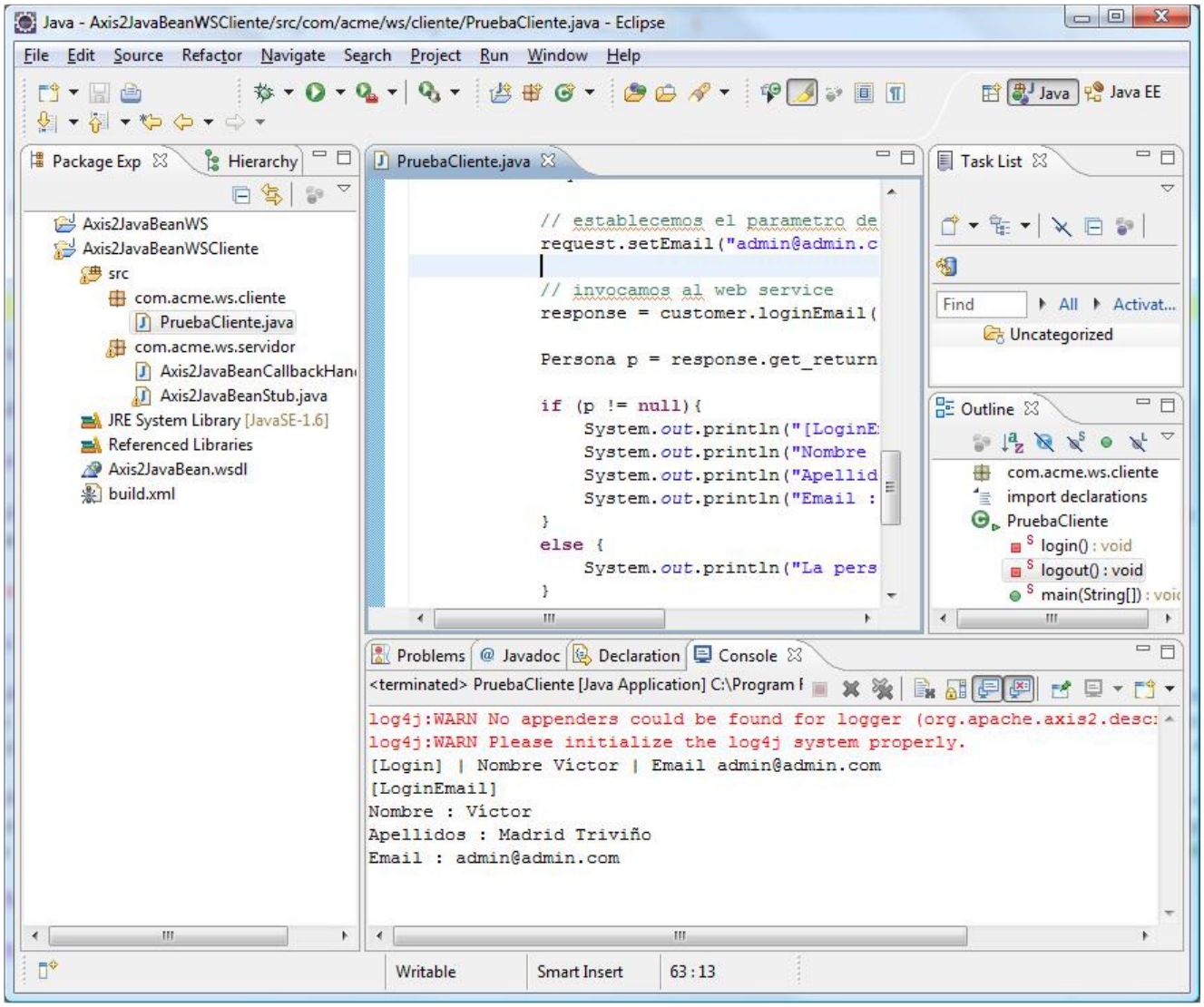


PASO 7 : Crear una clase PruebaCliente.java

Esta clase dipondrá de dos métodos para realizar la llamada a los dos servicios

```
package
com.acme.ws.cliente;
```

PASO 8 : Ejecutar la clase



Con esto habriamos conseguido tener un servicio web en el que se utiliza la estructura compleja del tipo JavaBean tanto de entrada como de salida.

7. Conclusiones.

Con estas pequeñas aportaciones espero estar ayudar a entender un poco mejor las estructuras complejas en SOAP. En el caso de este tutorial se ha utilizado algo muy habitual cuando uno envía una estructura propia usando SOAP .Se ha tratado de que el ejemplo sea lo más claro posible para facilitar su entendimiento sin entrar en ningúna complejidad a la hora de crear los servicios proporcionados.

Ya casi tenemos dominado esto de las comunicación SOAP ;-)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario
(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS

Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2010 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia |