

¿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

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

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

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



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

Tutorial desarrollado por: [Francisco Javier Martínez Páez](#)

**Puedes encontrarme en [Autentia](#)
Somos expertos en Java/J2EE
Contacta en info@autentia.com**



Descargar este documento en formato PDF [webservicesAxis.pdf](#)

[Firma en nuestro libro de Visitas](#)

Integrar SOA, WebServices

Sus datos 3270/5250 en J2EE & Java Integrar CICS/IMS con BEA, CRM

Cursos J2ee

La guía con la información que buscas para ampliar tu formación.

C++ y Visual C++

Programación avanzada C++ y Visual C++ Curso en Barcelona

ADSL + Llamadas gratis

Ahorro telefónico garantizado con el ADSL de Telefónica. Ahora online

Anuncios Goooooogle

Anunciarse en este sitio

WEBSERVICE CON JBOSS Y AXIS.

Todo el mundo sabe ya, o al menos entiende la potencia que supone el uso de los webservices. Poder hacer uso del concepto de RPC y que además sea independiente de la plataforma y del language de programación que se use, y que además, estos servicios puedan ser invocados a través de HTTP o HTTPS hace que se pueda decir de esta tecnología, que promete ser la próxima generación en el desarrollo de software. En pocas palabras, los webservices son llamadas a procedimientos remotos, usando como protocolo de comunicaciones SOAP sobre HTTP/s y como formato de los mensajes transmitidos XML, o mejor dicho SOAP-XML o envelopes. Además, incluye la publicación de la definición de los servicios en base al WSDL, de forma parecida a como los IDL definen los servicios en CORBA.

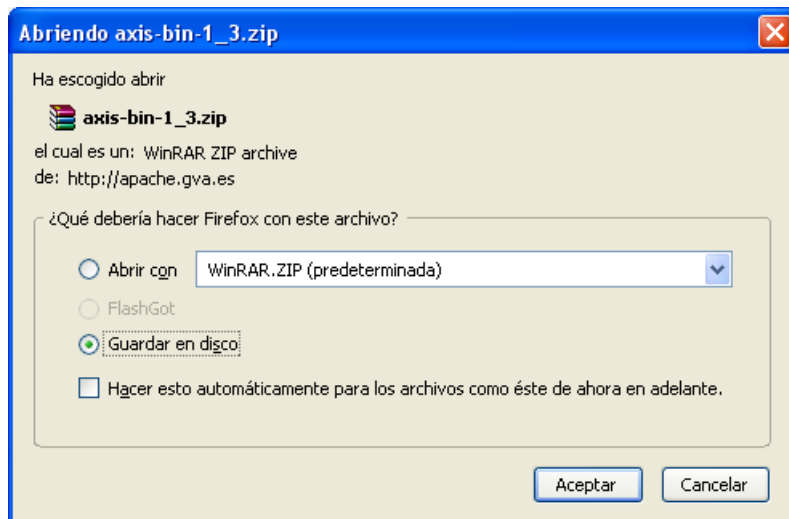
Sin más, porque no se pretende en este tutorial explicar la tecnología en profundidad, sino enseñar a crear un servicio WEB usando AXIS y Jboss.

Lo primero es instalar AXIS. ¿Pero que es AXIS ?. Axis no es más que una aplicación WEB del proyecto XML de apache que implementa el protocolo SOAP.

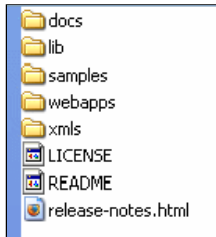
Por lo tanto, el primer paso será instalar Axis (suponemos que ya tenemos instalado el servidor Jboss, sino es así, en la dirección <http://labs.jboss.com/portal/download> podemos descargarlo. En el ejemplo estamos usando JBoss 4.0.2).

Nos vamos a la página de apache Axis <http://ws.apache.org/axis/> y seleccionamos la versión a descargar. Como vamos a trabajar con la versión 1.3, tenéis aquí el enlace directo a la descarga:

http://apache.gva.es/ws/axis/1_3/ y pulsamos sobre:



Guardamos el fichero en disco, y lo descomprimos en alguna carpeta del disco:



Nos fijamos en el directorio webapps. Dentro de este directorio, existe una carpeta llamada axis. Vamos a cambiar el nombre de esta carpeta y la vamos a llamar **axis.war**. Una vez que hayamos hecho esto, vamos a copiar la carpeta directamente al directorio de despliegue del jboss. Por defecto, el directorio será:

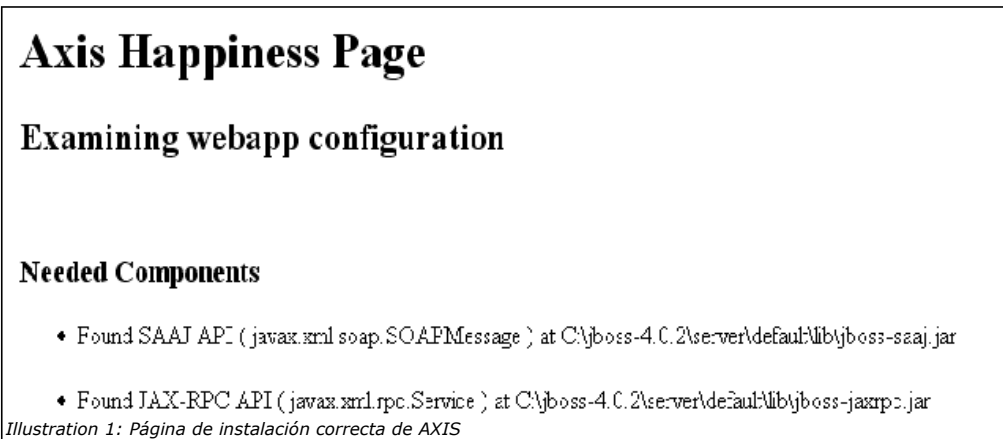
```
<RUTA_JBOSS>\server\default\deploy\webapps
```

Si no existe la carpeta webapps, la creamos.

Probablemente, si tenemos arrancado el servidor de aplicaciones, la aplicación se estará desplegando en caliente en este momento.

Podemos comprobar si este despliegue se ha hecho correctamente, en la página: (Comprabad el puerto por si no coincide con el vuestro.)

<http://localhost:8080/axis/happyaxis.jsp>



Si hemos llegado aquí, ya estamos preparados para hacer nuestro primer webservice.

Lo primero que haremos será crear un EJB de sesión si estado, que hará la lógica de negocio de nuestro webservice, lo que se suele llamar **endpoint** (la capa subyacente que realiza el servicio). Para el cliente, esto será completamente transparente. Supongo que ya estarás deseando saber que va a hacer nuestro webservice, pues vamos a implementar el primer euroconvertor euros2pesetas, pesetas2euros de la historia vía SOAP.

Lo primero, definiremos el interfaz que debe cumplir el interfaz remoto de nuestro EJB. Sencillo, el interfaz:

```
/**
 * Interface para obligar al interface remoto de nuestro EJB
 *
 * @author Francisco Javier Martínez Páez
 */

public interface IEuroConvertor {

    public static final float PTAS_EURO = 166.386f;

    public String euros2pesetas(String euros) throws RemoteException;

    public String pesetas2euros(String pesetas) throws RemoteException;

}
```

El interfaz Remoto:

```
/**
 * Interface Remota de nuestro EJB
 * @author Francisco Javier Martínez Páez
 */

public interface IEuroConversorEJB extends EJBObject, IEuroConversor {
}
```

El interfaz Home:

```
/**
 * Interface Home de nuestro EJB
 * @author Francisco Javier Martínez Páez
 */

public interface IEuroConversorEJBHome extends EJBHome {

    public IEuroConversorEJB create() throws RemoteException, CreateException;

}
```

El bean:(sin entrar a discutir sobre la implementación de la euroconversión, que sin duda se puede hacer mucho mejor)

```
/**
 * Clase que implementa los métodos de los interface Home, Remoto y INegocioAcademia
 * @author Francisco Javier Martínez Páez
 */

public class EuroConversorEJBBean implements SessionBean, IEuroConversor {

    private SessionContext ctx = null;

    public String euros2pesetas(String euros) throws RemoteException {

        float fEuros = 0f;

        try {

            fEuros = Float.parseFloat(euros);

            fEuros *= IEuroConversor.PTAS_EURO;

            fEuros=Math.round(fEuros);

        } catch (Exception e) {

            System.out.println("error convirtiendo");

            fEuros = 0f;

        }

        return ""+fEuros;
    }
}
```

```

    }

    public String pesetas2euros(String pesetas) throws RemoteException {

        float fPesetas = 0f;

        try {

            fPesetas = Float.parseFloat(pesetas);

            fPesetas /= IEuroConversor.PTAS_EURO;

            fPesetas=Math.round(fPesetas);

        } catch (Exception e) {

            System.out.println("error convirtiendo");

            fPesetas = 0f;

        }

        return ""+fPesetas;

    }

    public void ejbCreate() throws CreateException{System.out.println("Me están creando...");}

    public void ejbActivate() throws EJBException, RemoteException { System.out.println("Me Activan...");}

    public void ejbPassivate() throws EJBException, RemoteException {System.out.println("Me Pasivan..."); }

    public void ejbRemove() throws EJBException, RemoteException { System.out.println("Me están borrando..."); }

    public void setSessionContext(SessionContext ctx) throws EJBException,RemoteException {this.ctx = ctx;}

}

```

El descriptor del EJB:

```

<ejb-jar>

<enterprise-beans>

<!-- Session Beans -->

<session>

    <ejb-name>EuroEJB</ejb-name>

    <description>EJB de eurocalculadora</description>

    <display-name>EuroEJB</display-name>

    <home>com.cursos.inet.ejbEuro.IEuroConversorEJBHome</home>

    <remote>com.cursos.inet.ejbEuro.IEuroConversorEJB</remote>

    <ejb-class>com.cursos.inet.ejbEuro.EuroConversorEJBBean</ejb-class>

    <session-type>Stateless</session-type>

    <transaction-type>Container</transaction-type>

</session>

</enterprise-beans>

</ejb-jar>

```

El descriptor del ejb para jboss:(debemos fijarnos en el nombre JNDI para recordarlo)

```

<jboss>

<enterprise-beans>

<session>

<ejb-name>EuroEJB</ejb-name>

<jndi-name>ejb/EuroEJB</jndi-name>

</session>

</enterprise-beans>

</jboss>

```

Creamos el jar para despliegue del ejb (ejbEuro.jar) con esta estructura:

- META-INF/
 - ejb-jar.xml
 - jboss.xml
- com/.../IEuroConversorEJBHome.class
- com/.../IEuroConversorEJB.class
- com/.../IEuroConversor.class
- com/.../EuroConversorEJBBean.class

Y lo desplegamos (lo copiamos en <RUTA_JBOSS>\server\default\deploy\)

Ya tenemos instalado Axis y creado y desplegado nuestro EJB de endpoint.

¿ Y ahora ?. Pues ahora vamos a desplegar el webservice haciendo uso de la magia de Axis.

Para ello, y para evitar conocer la sintaxis de los ficheros WSDL, usaremos un fichero intermedio (wsdd webservice deployment descriptor) que nos va a hacer transparente este tema:

```

<deployment xmlns="http://xml.apache.org/axis/wsdd/"

xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">

<service name="EuroCalculadora" provider="java:EJB">

<parameter name="beanJndiName" value="ejb/EuroEJB"/>

<parameter name="homeInterfaceName" value="com.cursos.inet.ejbEuro.IEuroConversorEJBHome"/>

<parameter name="remoteInterfaceName" value="com.cursos.inet.ejbEuro.IEuroConversorEJB"/>

<parameter name="allowedMethods" value="*/>

<parameter name="jndiURL" value="jnp://localhost:1099"/>

<parameter name="jndiContextClass" value="org.jnp.interfaces.NamingContextFactory"/>

</service>

</deployment>

```

Lo vamos a llamar **deploy-euro.wsdd**. Los parámetros se comentan por si solos (fijaros en el nombre JNDI del EJB, ha de ser el mismo que pusimos en jboss.xml)

Antes de usar la magia de axis, debemos configurar el CLASSPATH del sistema para que java pueda encontrar las librerías de axis. Una vez hecho esto, nos vamos al directorio donde está el fichero deploy-euro.wsdd y lanzamos la siguiente aplicación:

```
java org.apache.axis.client.AdminClient deploy-euro.wsdd:
```

```

Processing file deploy-euro.wsdd
<Admin>Done processing</Admin>

```

En este momento, ya deberíamos haber desplegado el webservice.

Lo comprobamos en: **<http://localhost:8080/axis/servlet/AxisServlet>**

Que nos muestra la lista de servicios desplegados:

And now... Some Services

- ◆ MyAcademiaService ([wsdl](#))
 - ◊ ejecutar
- ◆ AdminService ([wsdl](#))
 - ◊ AdminService
- ◆ EuroCalculadora ([wsdl](#))
 - ◊ euros2pesetas
 - ◊ pesetas2euros
- ◆ Version ([wsdl](#))
 - ◊ getVersion

Illustration 8: Página de los servicios registrados o desplegados

Nos fijamos en el que acabamos de desplegar: EuroCalculadora, que tienes dos métodos a los que se puede invocar.

Fijémonos en el wsdl generado: **<http://localhost:8080/axis/services/EuroCalculadora?wsdl>**

Podemos ver ahí, la descripción del webservice, métodos y parámetros...

Probemos nuestro webservice directamente:

Convertiremos 12 euros a pesetas:

<http://localhost:8080/axis/services/EuroCalculadora?method=euros2pesetas&in0=12>

El resultado: (1997.0 pesetas ...sin entrar a valorar el resultado que podría ser más ajustado)

```

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  >
  <soapenv:Body>
    <euros2pesetasResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <euros2pesetasReturn xsi:type="soapenc:string"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1997.0</euros2pesetasReturn>
    </euros2pesetasResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Illustration 9: Resultado de la invocación del webservice

Vamos a crear por último un cliente web, que solicite la conversión al webservice. Debemos asegurarnos, que la aplicación cliente que quiera invocar al webservice tenga las librerías de AXIS.

Vamos a hacer un formulario con este aspecto:

Euro Conversor usando un Servicio WEB

Introduzca la cantidad a convertir:

Elija la conversión:

Euros a pesetas: ☐

Pesetas a euros: ☒

Illustration 10: Imagen del JSP de captura de los datos

Veremos el resultado de la conversión de 2000 pesetas a euros:

Segun mi WebService el valor buscado es: 12.0 euros

Illustration 11: Resultado de la invocación

El código del jsp que recoge los parámetros e invoca al webservice podría ser:

```
<%@page import="java.net.URL,org.apache.axis.client.Call" %>

<%

String valor=request.getParameter("valor");

String tipo=request.getParameter("tipo");

URL url = new URL( "http://localhost:8080/axis/services/EuroCalculadora?wsdl");

Call llamada= new Call(url);

Object [] parametros=new Object[1];

parametros[0]=valor;

String resultado="";

String moneda="";

if("1".equals(tipo)) {

    resultado=(String) llamada.invoke("euros2pesetas",parametros);

    resultado+=" pesetas";

} else {

    resultado=(String) llamada.invoke("pesetas2euros",parametros);

    resultado+=" euros";

}

%>

<h1>Segun mi WebService el valor buscado es: <%=resultado %></h1>
```

Ya está terminado. ¿ Sencillo verdad ?

No obstante, si necesitas ayuda, ya sabes como encontrarnos...



[Puedes opinar sobre este tutorial aquí](#)

Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación

Creatividad Internet

[Autentia S.L.](#) Somos expertos en:

J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

| | |
|---------------------------------|---------------------------------------|
| Subscribirse a Novedades | |
| e-mail | <input type="text"/> |
| | <input type="button" value="Enviar"/> |

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Activar soporte SSL en Tomcat](#)

[Apache, MySQL y PHP](#)

[Rendimiento de aplicaciones Web](#)

[Forzar diálogo Guardar Como en JSPs](#)

[Aplicaciones con JSPs](#)

[mod_jk en Linux / Apache2-JBoss](#)

[Serialización Servlet-Applet](#)

[mod_jk en Ubuntu / Apache2-JBoss](#)

[Generador automático de Webservices](#)

[mod_jk en WindowsXP / Apache2-JBoss](#)

Descripción

Os mostramos como activar el acceso SSL en Tomcat, utilizando certificados generados por Keygen (java)

Os mostramos como configurar Apache, MySQL y PHP en vuestra máquina

En este tutorial veremos una introducción al funcionamiento de la Suite e-Test de Empirix.

Os mostramos como afrontar tareas comunes en JSP: Forzar el diálogo Guardar Como al generar dinámicamente un fichero desde un JSP y asegurarnos que no se cachean nuestros ficheros (probado en IExplorer 6)

Os mostramos como construir una aplicación con JSP que acceda a MySQL

El conector mod_jk se encarga de enviar las peticiones dinámicas de Apache2 a un servidor de aplicaciones JBoss

Os mostramos un ejemplo para serializar una respuesta en la comunicación servlet-applet

Os mostramos como instalar el conector mod_jk sobre la distribución linux Ubuntu utilizando Apache2 y JBoss

Os mostramos como crear un servicio Web a partir de una clases, gracias a generadores automáticos de código y NetBeans

Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando Apache2 y JBoss

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)

