

¿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)



Powered by
autentia

Hosting Patrocinado por
enREDados.com



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

<p>Tutorial desarrollado por: Carlos García Pérez</p> <p>Puedes encontrarme en Autentia Somos expertos en Java/J2EE Contacta en info@autentia.com</p>	 <p>autentia real business solutions</p>
--	---

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

[Firma en nuestro libro de Visitas](#)

Infoco

Soluciones Contact Center.
Solicite información.

Iasaf Quality

Consultoría e implantación
Implante calidad por 200 € al mes

Sus propiedades online

Diseño de sitios para inmobiliarias
Sistema de Gestion integral online

Informático, envía tu CV

Somos cazatalentos con 3200
ofertas de empleo en C, PHP,
HTML, VB, etc.

Anuncios Goooooogle

Anunciarse en este sitio

Guia de Apache Axis

- 1.Introducción a Axis
- 2.Instalación de Axis sobre Tomcat
- 3.Creación del Servicio Web
- 4.Activación el servicio Web
- 5.Creación de un cliente utilizando un Proxy generado con las herramientas de Axis
- 6.Creación de un cliente nativo.
- 7.Otros comandos y herramientas de Axis
- 8.Conclusiones

En [Autentia](#) nos involucramos diariamente en el desarrollo de proyectos software de diversos tipos, uno de estos sectores es el de la interoperabilidad e integración de sistemas heterogéneos distribuidos.

1. Introducción

Apache Axis es una implementación OpenSource de SOAP que proporciona un entorno de ejecución para Servicios Web implementados en Java.

A grandes rasgos, un Servicio Web es un conjunto de métodos que realizan una funcionalidad que se exponen al resto de las aplicaciones.

Cualquier aplicación sea cual sea su plataforma o lenguaje en la que está implementada podrá invocar los métodos que expone el Servicio Web.

Por ejemplo, una aplicación .Net (Implica una plataforma Windows) podría invocar métodos expuestos por un Servicio Web Java ejecutándose en una plataforma Linux.

Esto se consigue utilizando protocolos estándar como XML y HTTP y se evitan los problemas con Firewalls, etc. que otras tecnologías similares como CORBA o RMI tenían.

Entre otras cosas Axis proporciona:

- Un entorno de ejecución para Servicios Web Java (*.jws)
- Herramientas para crear WSDL desde clases java.
- Herramientas para crear clientes Java desde un WSDL.
- Herramientas para desplegar, probar y monitorizar Servicios Web.

- Integración con servidores de aplicaciones y contenedores de Servlets.

2. Instalación de Axis sobre Tomcat

- Descárgate la versión binaria de Axis desde <http://ws.apache.org/axis/>
- Descomprime el fichero.
- Copia el directorio completo axis al directorio: TOMCAT_HOME/webapps/axis

Para ver si Axis está instalado mostramos la URL <http://localhost:8080/axis/happyaxis.jsp> y nos mostrará una página de bienvenida con la configuración de Axis.

3. Creación del Servicio Web

Para crear un Servicio Web creamos las clases con la lógica y métodos que deseamos para nuestro Servicio Web. Para nuestro ejemplo vamos a crear un Servicio Web que proporcione métodos para sumar, restar, multiplicar y dividir dos números enteros.

Calculadora.java

```
/**
 * Servicio Web que realiza las operaciones Suma, Resta, Multiplicación y División de dos números
 * @author Autentia Real Business Solutions.
 */
public class Calculadora {
    /**
     * Realiza la suma dos números enteros
     * @param x Primer operando
     * @param y Segundo operando
     * @return Devuelve el resultado de la operación (x+y)
     */
    public int suma(int x, int y) {
        return x + y;
    }

    /**
     * Realiza la resta dos números enteros
     * @param x Primer operando
     * @param y Segundo operando
     * @return Devuelve el resultado de la operación (x-y)
     */
    public int resta(int x, int y) {
        return x - y;
    }

    /**
     * Realiza la multiplicación de dos números enteros
     * @param x Primer operando
     * @param y Segundo operando
     * @return Devuelve el resultado de la operación (x*y)
     */
    public int multiplica(int x, int y) {
        return x * y;
    }

    /**
     * Realiza la división de dos números enteros
     * @param x Primer operando
     * @param y Segundo operando
     * @return Devuelve el resultado de la operación (x/y)
     */
    public int divide(int x, int y) {
        return x / y;
    }
}
```

4. Activación del Servicio Web

Ahora vamos a desplegarlo (= activarlo) en el Tomcat para que pueda ser invocado desde otra aplicación.

(Axis también proporciona un servidor Stand Alone para probar Servicios Web, pero para nuestro ejemplo nos apoyaremos en el servidor de aplicaciones Tomcat)

Para hacerlo tenemos dos formas:

1. Renombrar el fichero Calculadora.java a Calculadora.jws y copiarlo al directorio TOMCAT_HOME/webapps/axis. (Si hemos

instalado correctamente AXIS, todos los ficheros acabados en .jws serán procesados por Axis.). De esta manera la primera vez que se invoque el Web Service, será compilado automáticamente.

2. Compilar el fichero Calculadora.java y colocamos el .class en el directorio TOMCAT_HOME/webapps/axis/WEB-INF/classes. Ahora definimos su descriptor de activación o despliegue (wsdd) y ejecutamos el comando:

```
java org.apache.axis.client.AdminClient Calculadora.wsdd
```

Calculadora.wsdd

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

  <!--
    Definimos el servicio Web a activar:
    Nombre y tipo de Servicio Web.
    RPC Llamadas a procedimientos remotos con ejecución síncrono
  -->

  <service name="CalculadoraWS" provider="java:RPC">

    <!-- Nombre de la clase que implementa los métodos expuestos -->
    <parameter name="className" value="Calculadora"/>

    <!-- Expone todos los métodos como visibles desde el exterior -->
    <parameter name="allowedMethods" value="*" />
  </service>
</deployment>
```

Ahora si deseamos ver una lista de los Servicios Web activados en la máquina local ejecutamos navegamos a la dirección:

<http://localhost:8080/axis/servlet/AxisServlet>

Deberá aparecer nuestro Servicio Web CalculadoraWS y una lista con los métodos que exporta.

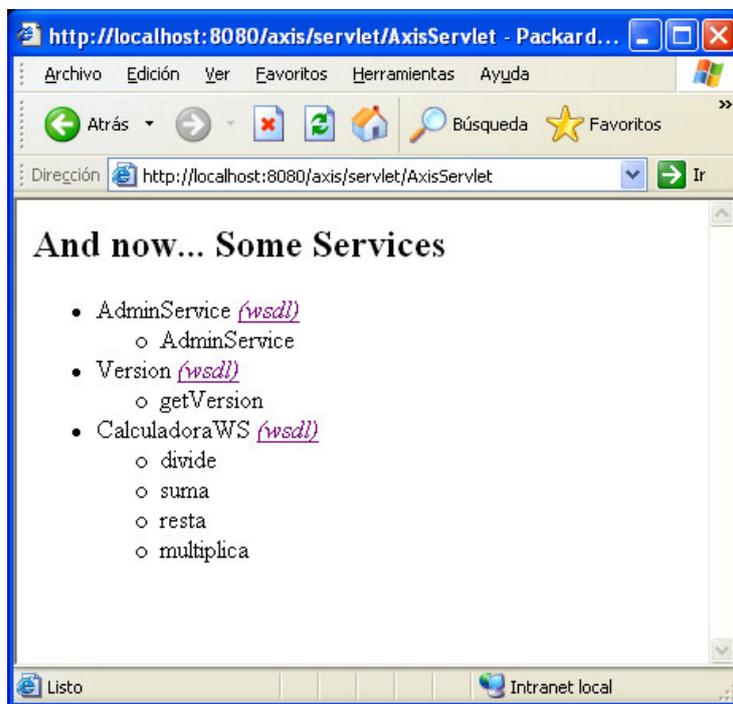


Illustration 1: Servicios Web activos

5. Creación de un cliente utilizando un Proxy generado con las herramientas de Axis

La forma más fácil y rápida para comenzar a utilizar los métodos que expone un Servicio Web es apoyarnos en un Proxy. Un Proxy o Stub no es más q un conjunto de clases que nos abstraen de los detalles de la comunicación con el Webservice.

Para generar el Proxy debemos tener la especificación del Servicio Web en formato WSDL (Web Service Description Language). Este formato es un estándar de definición de servicios Web. Es independiente de la plataforma, de lenguajes de programación. Y básicamente define: donde, como y que hace un Servicio Web.

Para obtener la definición del servicio Web (WSDL) podemos realizar lo de dos formas:

- Navegando a la dirección:

<http://localhost:8080/axis/services/CalculadoraWS?wsdl>

- Ejecutando el comando:

```
java org.apache.axis.wSDL.Java2WSDL
```

```
-o CalculadoraWS.wsdl
```

```
-l http://localhost:8080/axis/services/CalculadoraWS
```

```
-p Calculadora.class
```

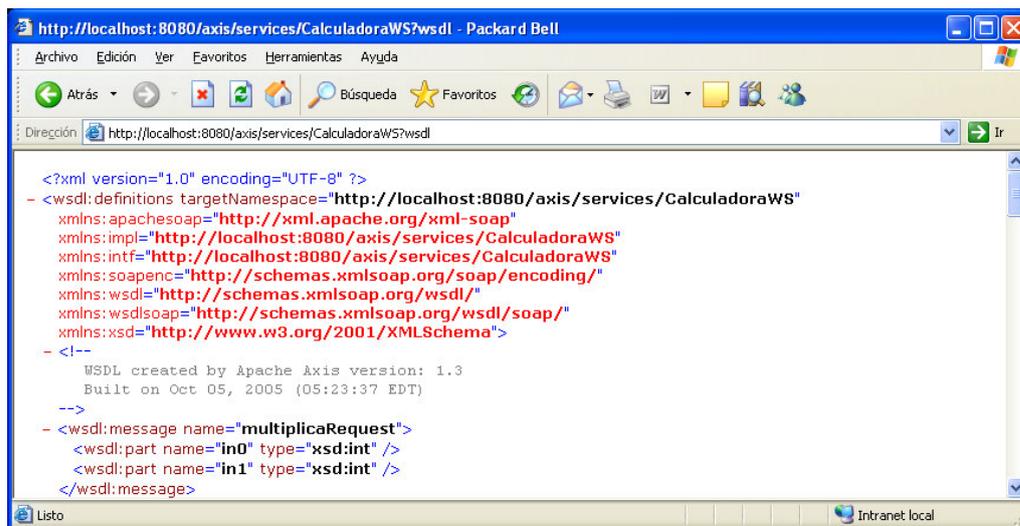


Illustration 2: WSDL del Servicio Web CalculadoraWS

Guardamos esa información en el fichero CalculadoraWS.wsdl .

Ahora ejecutamos el comando:

```
java org.apache.axis.wSDL.WSDL2Java CalculadoraWS.wsdl
```

Y Axis nos generará automáticamente las clases Java que hacen de Proxy para que nuestras aplicaciones puedan hacer uso de los métodos expuestos por el Servicio Web.

```
import localhost.axis.services.CalculadoraWS.*;

/**
 * Prueba el Web Service
 * @author Autentia Real Business Solutions
 */
public class Class1 {

    public static void main(String[] args) {
        localhost.axis.services.CalculadoraWS.CalculadoraServiceLocator locator;
        localhost.axis.services.CalculadoraWS.Calculadora calc;

        try {
            locator = new CalculadoraServiceLocator();
            calc = locator.getCalculadoraWS();

            System.out.println(calc.suma(6, 2));
            System.out.println(calc.resta(3, 2));
            System.out.println(calc.multiplica(9, 2));
            System.out.println(calc.divide(100, 4));

        } catch (Exception ex){
            System.out.println(ex);
        }
    }
}
```

Compile y ejecute la aplicación para ver el resultado.

6. Creación de un cliente Nativo

Ahora vamos a hacer un cliente a manita utilizando las clases de Axis.

El cliente simplemente multiplicará dos números, pero si observa el código fuente este es bastante didáctico. Observe el texto resaltado pues estos deben ser iguales a los que contiene el WSDL. Es decir, deben coincidir la dirección del servicio, el nombre del método a invocar y el nombre de los parámetros necesarios para invocar el método.

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import javax.xml.rpc.ParameterMode;

/**
 * Prueba el Web Service 2
 * @author Autentia Real Business Solutions
 */
public class Class2 {
    public static void main(String [] args) throws Exception {
        String endpoint = "http://localhost:8080/axis/services/CalculadoraWS";
        Integer op1      = new Integer(100);
        Integer op2      = new Integer(200);
        Service service  = new Service();
        Call call        = (Call) service.createCall();

        // Establecemos la dirección en la que está activado el Webservice
        call.setTargetEndpointAddress( new java.net.URL(endpoint) );

        // Establecemos el nombre del método a invocar
        call.setOperationName( "multiplica" );

        // Establecemos los parámetros que necesita el método
        // Observe que se deben especificar correctamente tanto el nombre como el tipo de datos..
        // esta información se puede obtener viendo el WSDL del servicio Web

        call.addParameter( "in0", XMLType.XSD_INT, ParameterMode.IN );
        call.addParameter( "in1", XMLType.XSD_INT, ParameterMode.IN );

        // Especificamos el tipo de datos que devuelve el método.
        call.setReturnType( XMLType.XSD_INT );

        // Invocamos el método
        Integer result = (Integer) call.invoke( new Object [] { op1, op2 } );

        // Imprimimos los resultados
        System.out.println("El resultado de la multiplicación es: " + result);
    }
}
```

7. Otros comandos y herramientas de AXIS

Además de los comandos mostrados anteriormente, Axis nos proporciona entre otros los siguientes comandos o utilidades:

Para obtener información acerca de la configuración de axis puede ejecutar el comando:

```
java org.apache.axis.client.AdminClient list
```

Ejecutar Axis sin Tomcat ni ningún otro servidor de aplicaciones:

```
java org.apache.axis.transport.http.SimpleAxisServer <port>
```

Utilidad para ver y depurar los mensajes SOAP implicados en la comunicación:

```
Java org.apache.axis.utils.tcpmon <listeningPort> <targetServeName> <targetServerPort>
```

8. Conclusiones

Apache Axis es una implementación sólida, madura y extendida para ejecutar, testear y administrar Servicios Web implementados en Java. Además detrás de Axis hay importantes organizaciones como Apache, por lo que eso nos da confianza a la hora de elegir Axis como entorno de ejecución de servicios Web.

Axis nos proporciona herramientas suficientes para crear y depurar nuestros Servicios Web de una manera fácil.



[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	Descripción
Instrumentación de componentes Java usando JMX	Os mostramos como utilizar las características avanzadas de los servidores de aplicaciones Java/J2EE para realizar la instrumentación de componentes Java usando JMX y poniendo como ejemplo la gestión dinámica de propiedades entre nodos
Extender la validación en Struts	Os mostramos con un ejemplo como extender los mecanismos de validación en Struts, utilizando el framework Commons Validator
Web Services en tu IPAQ	Cesar Crespo nos enseña como programar accesos Web Services desde tu IPAQ en Visual C++ con PocketSOAP, Apache SOAP y Axis
Upload de ficheros en JSF	Os mostramos de una forma sencilla y guiada como crear una utilidad de upload de ficheros utilizando JSF
Realizar Instaladores con NSIS 2.0	Cesar Crespo no enseña como construir instaladores/desinstaladores, de un modo sencillo, a través de la herramienta gratuita NSIS 2.0.
Arquitectura SOA e Integración de aplicaciones	Os presentamos un breve pero intenso pdf sobre la tecnología Arquitectura SOA y la Integración de aplicaciones
WebServices con Axis y JBoss	En este tutorial os mostramos como realizar servicios web utilizando Axis y el contenedor de aplicaciones web JBoss
Validar en JSF con Commons Validator	En este nuevo tutorial sobre el framework JSF os mostramos como utilizar y extender la validación del Commons Validator
Conversión y validación en JSF	En este nuevo tutorial sobre JSF os mostramos como utilizar y extender los mecanismos básicos de conversión y validación
Pruebas de Rendimiento y Funcionales Web	Jose María Toribio, nos enseña en este tutorial como podemos utilizar la aplicación gratuita JMeter para realizar pruebas de rendimiento y funcionales (vitales para la regresión y reingeniería) sobre nuestras aplicaciones Web

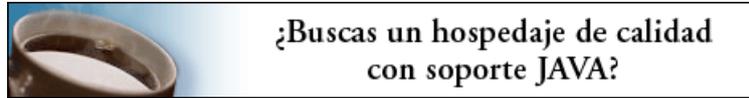
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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](#)



www.AdictosAlTrabajo.com Optimizado 800X600