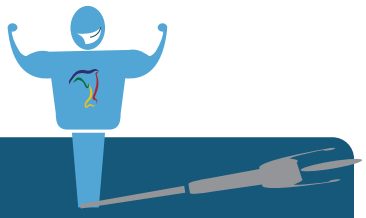
autentia

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...

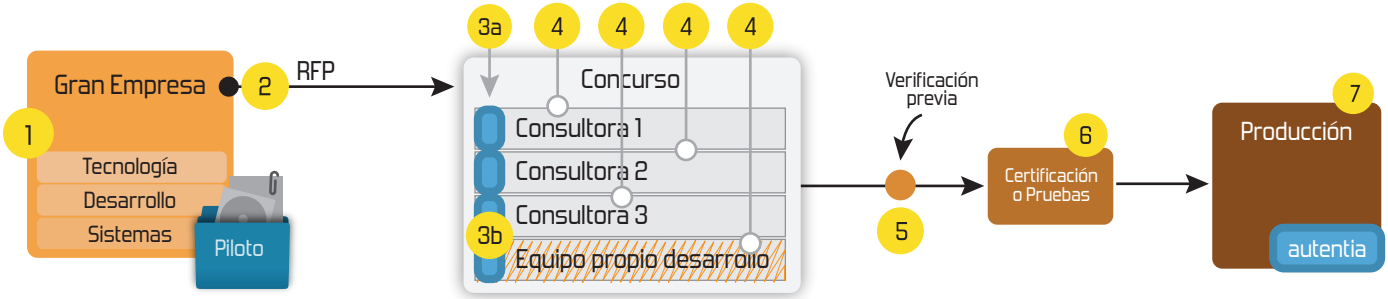
1. Desarrollo de componentes y proyectos a medida



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

Tutorial desarrollado por: Francisco Javier Martínez Páez

Puedes encontrarme en Autentia
Somos expertos en Java/J2EE
Contacta en info@autentia.com

autentia
real business solutions

Descargar este documento en formato PDF WsSSL.pdf

CREACIÓN E INVOCACIÓN DE WEBSERVICES POR SSL

Los ejemplos de este tutorial están hechos con el siguiente entorno de desarrollo:

- Jboss Eclipse IDE Milestone 5.
- JDK 1.4
- JBoss 4.0.4 GA
- Axis 1.3

En este tutorial se pretende enseñar al lector, (si aún no lo sabe), a desplegar un webservice usando SSL y a invocarlo correctamente. Se presupone que el lector ya sabe instalar axis, crear un webservice y desplegarlo en axis, generar un certificado autofirmado, importar y exportar el certificado e instalarlo en JBoss . Si no es así, hay varios tutoriales en adictosaltrabajo.com que explican como realizarlo. No obstante, durante el desarrollo del tutorial se mostrará como realizarlo sin entrar en detalle.

GENERACIÓN DEL CERTIFICADO.

Lo primero que hemos de hacer para trabajar con SSL es generarnos un certificado con su par de claves.

Lo haremos con la herramienta keytool de la JDK 1.4.

> *keytool -v -genkey -alias* ***PACO_PAIR*** *-keystore* ***AUTENTIA_KEYS.ssl***

Rellenad la información que os pide (no os olvideis de la contraseña)

Generamos un par que llamamos PACO_PAIR y la almacenamos en el fichero AUTENTIA_KEYS.ssl.

INSTALACIÓN DEL CERTIFICADO.

Una vez generado el certificado, lo "instalaremos" en JBoss. Para ello, copiaremos el fichero que acabamos de generar (AUTENTIA_KEYS.ssl) a la ruta de JBoss:

<INSTALACION_JBOSS>\server\default\conf

Vamos a configurar el JBoss para que abra el puerto SSL e indicarle el almacén de certificados que queremos usar:

Editamos el fichero:

<INSTALACION_JBOSS>\server\default\deploy\jbossweb-tomcat55.sar\server.xml

Buscad la siguiente entrada y modificar los siguiente (no olvidéis descomentarla):

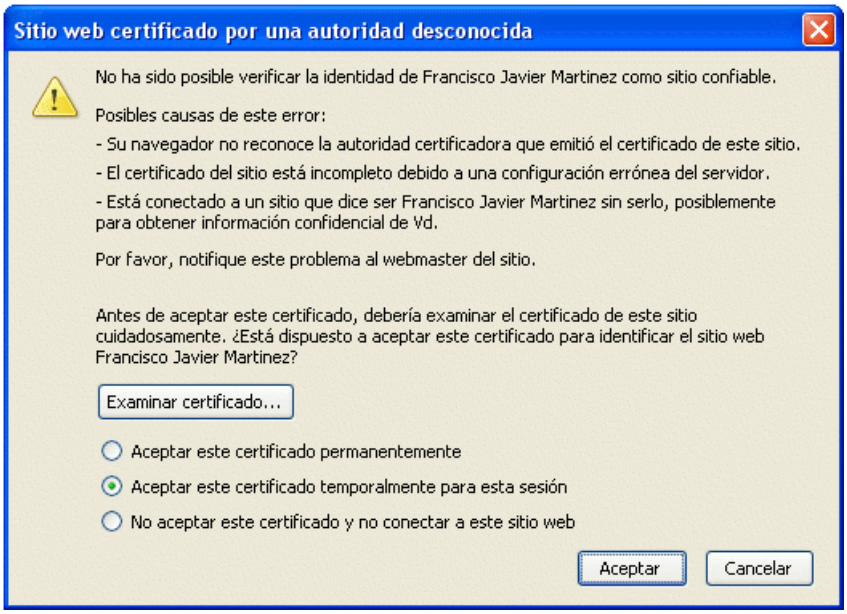
```
<!-- SSL/TLS Connector configuration using the admin devl guide keystore -->
   <Connector port="8443" address="${jboss.bind.address}"
        maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
        emptySessionPath="true"
        scheme="https" secure="true" clientAuth="false"
        keystoreFile="${jboss.server.home.dir}/conf/AUTENTIA_KEYS.ssl"
        keyAlias="PACO_PAIR"  keystorePass="autentia" sslProtocol = "TLS"        />
```

Le hemos "dicho" a JBoss (a tomcat en realidad) que queremos usar el puerto 8443 para comunicación SSL, usando el par de claves PACO_PAIR almacenado en AUTENTIA_KEYS.ssl, usando autentia como password para abrir el almacén.

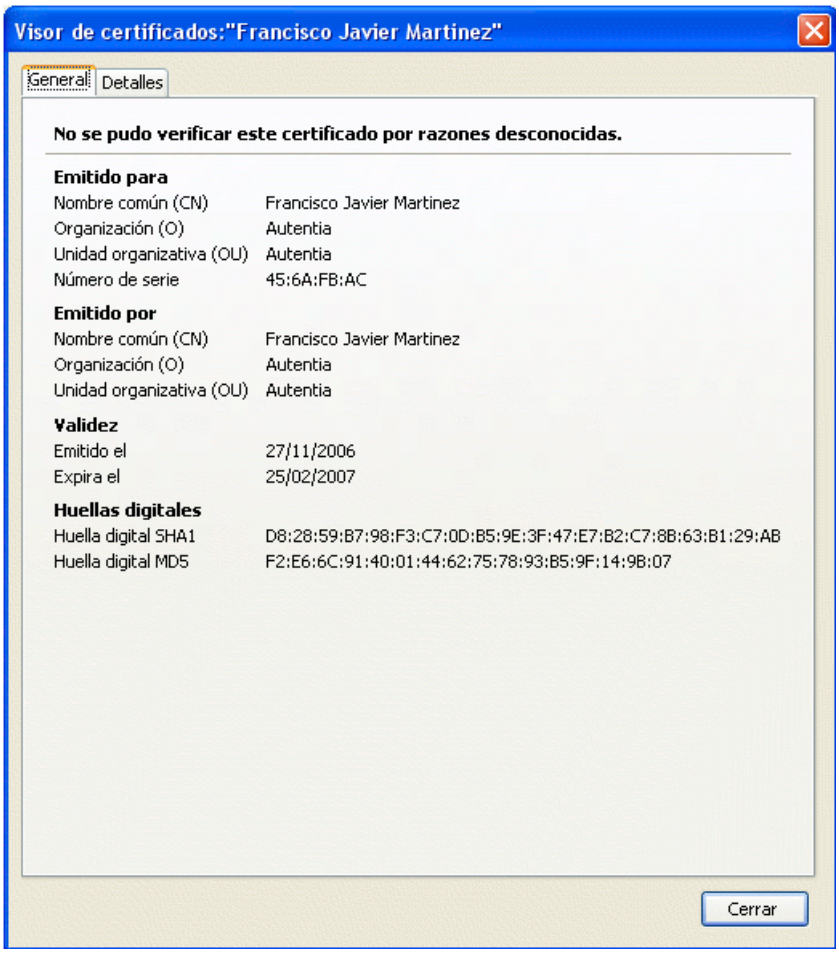
Si arrancamos ahora JBoss, podemos comprobar el resultado arrancando algun navegador y conectándonos al servidor:

http://localhost:8443

Al ser un certificado autofirmado, nos muestra el siguiente mensaje:



Si le decimos examinar certificado, veremos la información que rellenamos previamente al generarlo:



CREAMOS Y DESPLEGAMOS EL WEBSERVICE EN AXIS.

Nos creamos un interfaz donde definir los métodos del webservice. Nosotros, para hacerlo fácil y rápido, crearemos una clase Calculadora y le crearemos el método suma (un tanto típico):

```
package com.autentia.tutoriales.ws;

import java.rmi.RemoteException;

public class Calculadora {
   public int suma(int sum1, int sum2) throws RemoteException {
      return sum1 + sum2;
   }
}
```

Nos generamos su descriptor de despliegue (deployCalcu.wsdd):

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <service name="Calculadora" provider="java:RPC">
  <parameter name="className" value="com.autentia.tutoriales.ws.Calculadora"/>
      <parameter name="allowedMethods" value="*"/>
   </service>
</deployment>
```

Desplegamos el servicio:

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

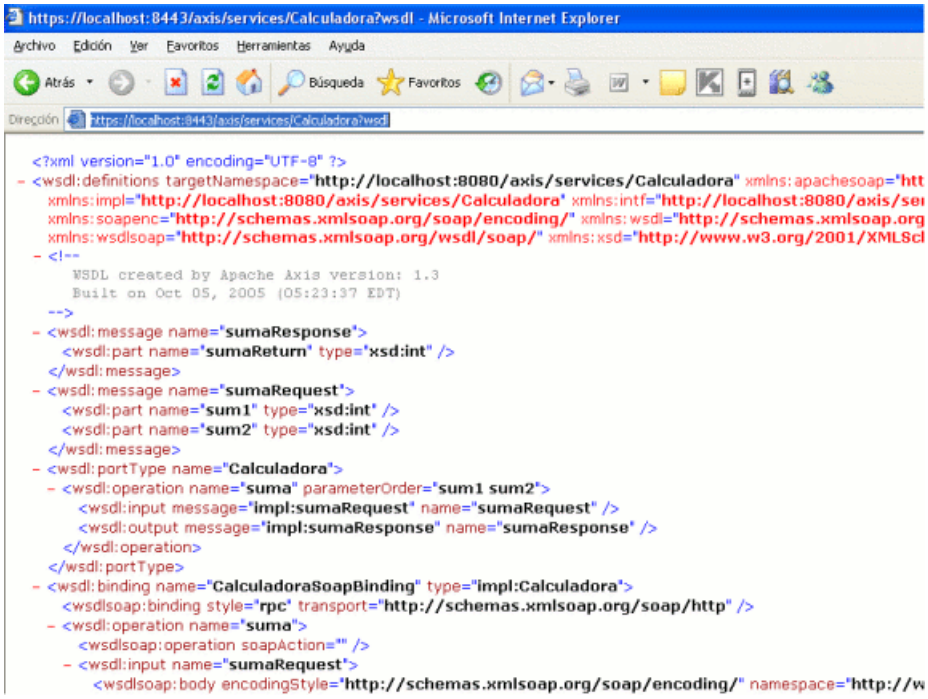
No olvidéis copiar la clase compilada en (Calculadora.class) en:

**<INSTALACION_JBOSS>\server\default\deploy\webapps\axis.war\
WEB-INF\classes**

Recordad que debéis copiar la ruta completa: com/autentia/....

Vamos a comprobar que está desplegado:

https://localhost:8443/axis/services/Calculadora?wsdl

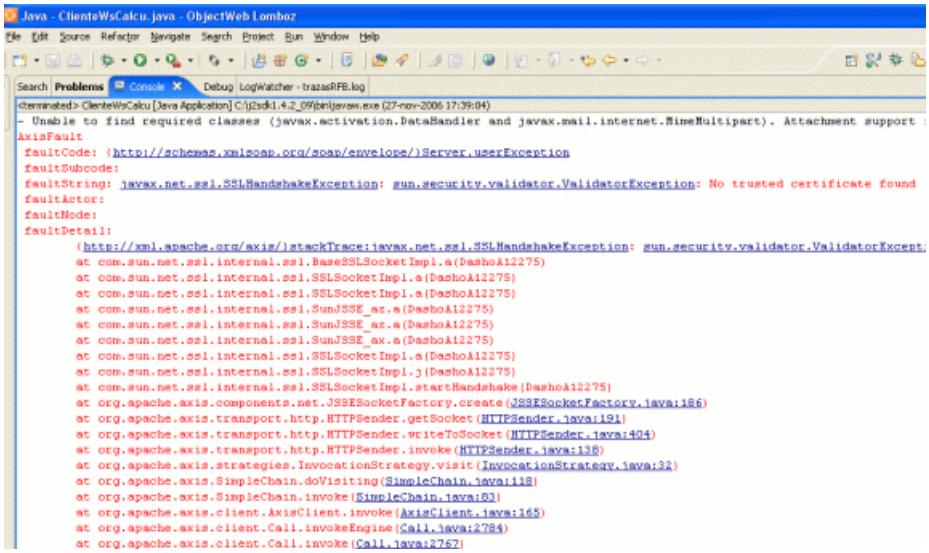


INVOQUEMOS EL WEBSERVICE DESDE UN CLIENTE:

Nos creamos la clase cliente, y la invocaremos desde su método main:

```
package com.autentia.tutoriales.ws;

import org.apache.axis.client.Call;
import org.apache.axis.client.Service;


public class ClienteWsCalcu {

   public static void main(String[] args) {
      Service service = new Service();
      Call call = null;
      try {
         call = (Call) service.createCall();
         String endpoint = "https://localhost:8443/axis/services/Calculadora";
         call.setTargetEndpointAddress( new java.net.URL(endpoint) );
         Object[] datos = new Object[2];
         datos[0] = new Integer(1);
         datos[1] = new Integer(2);
         Integer res = (Integer)call.invoke("suma",datos);
         System.out.println("RESULTADO (DEBERIA DAR 3)---->:"+res.toString());
      } catch (Exception e) {
         // TODO Auto-generated catch block
         e.printStackTrace();
      }
   }

}
```

Si ejecutamos el método main, veréis lo que ocurre:

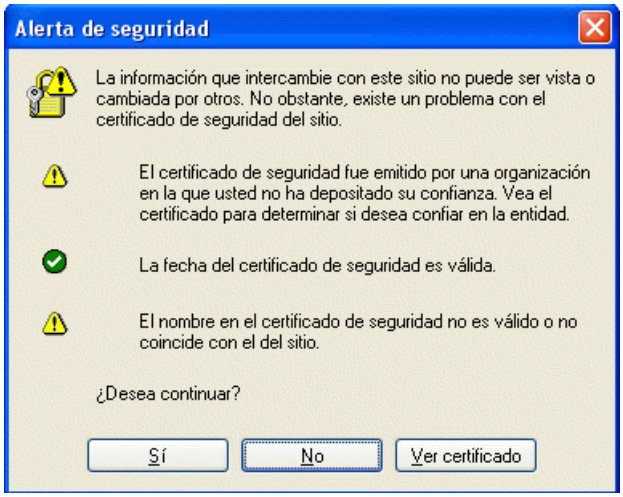


Obtenemos una excepción de seguridad: **"No trusted certicate found"**

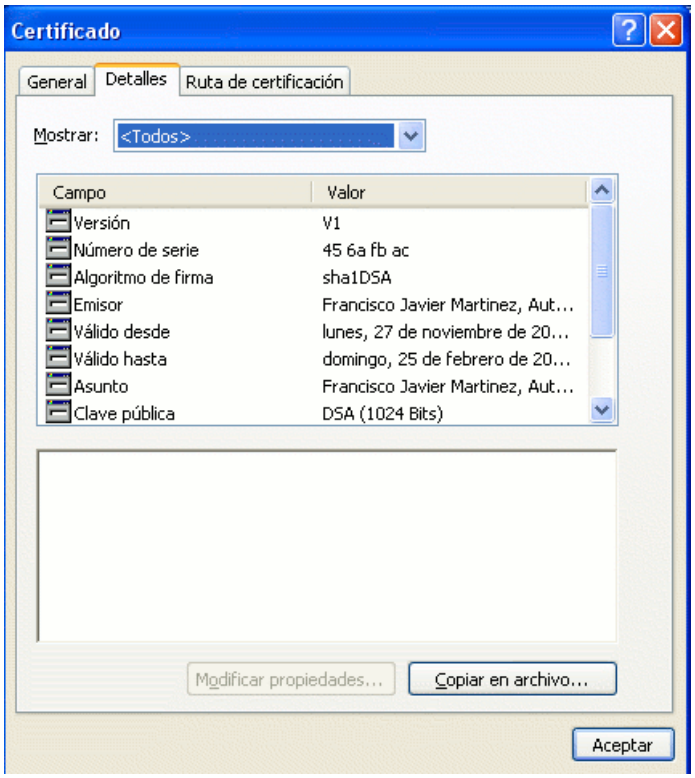
Es decir, se nos está diciendo que el certificado con el que estamos tratando no es de confianza. ¿Cómo lo resolvemos ?. Pues haciéndolo de confianza. Para ello, debemos importar el certificado en nuestro almacén de certificados de confianza:

Lo primero que haremos es importar el certificado. Vamos primero a obtenerlo (podríamos exportarlo del fichero de claves inicial pero lo vamos a hacer de otra manera)

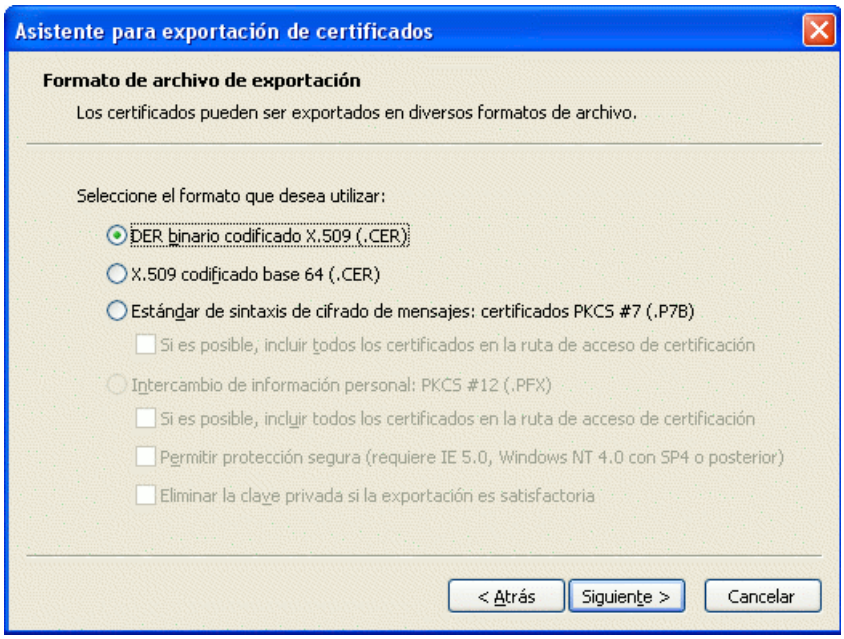
Abrimos internet explorer: **https://localhost:8443**



Ahora pulsamos en Ver certificado, pestaña detalles:



Pulsamos sobre Copiar en Archivo:

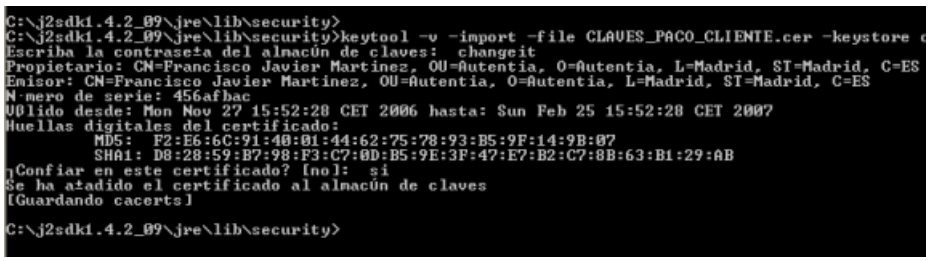


Seleccionamos DER binario codificado X.509 y guardamos el fichero en algún lugar del disco. Yo le he llamado: CLAVES_PACO_CLIENTE.cer

Vamos ahora a importarlo ahora al almacén de certificados de confianza:

Copiamos el fichero CLAVES_PACO_CLIENTE.cer a la ruta:
<RUTA_JDK>\jre\lib\security

Vamos a importar el certificado al fichero cacerts:

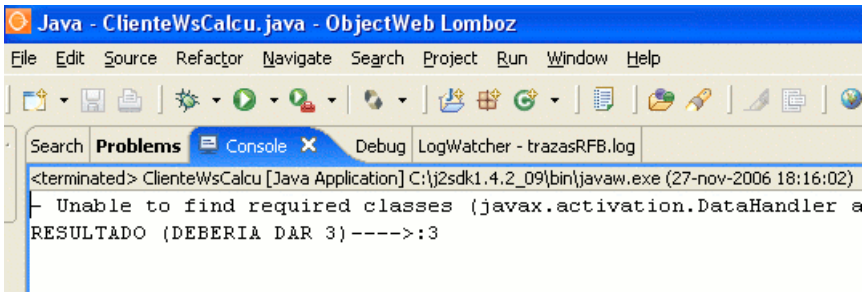


> keytool -v -import -file CLAVES_PACO_CLIENTE.cer -keystore cacerts

La contraseña por defecto del almacén de certificados de confianza de la JRE es "**changeit**"

Cuando pregunte: ¿Confiar en este certificado? : contesta: **si**

Ahora, si ejecutáis el código anterior debería funcionar.



Si aún no os funciona, probar a indicarle donde está el almacén de certificados de confianza en el arranque de la máquina virtual:

java .... -Djavax.net.ssl.trustStore=<RUTA_COMPLETA_FICHERO_ALMACEN>

(fichero cacerts)

Bueno, ya está todo.

Si necesitáis ayuda, ya sabéis donde encontrarnos: http://www.autentia.com

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

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	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
Activar SSL en IIS	Os mostramos comos activar el soporte de https en IIS, creando vuestros propios certificados autofirmados, usando OpenSSL
Certificados en IIS para activación SSL	En este tutorial vamos a habilitar el soporte SSL (Secure Socket Layer, comunicación segura por https) en un servidor IIS (Internet Information Server de Microsoft).
Generador automático de Webservices	Os mostramos como crear un servicio Web a partir de una clases, gracias a generadores automáticos de código y NetBeans
Activar soporte SSL en Tomcat	Os mostramos como activar el acceso SSL en Tomcat, utilizando certificados generados por Keygen (java)
Activar el soporte SSL en Struts	Os mostramos las particularidades de uso y configuración de Struts para trabajar con SSL
Guía rápida de instalación de JBOSS Application Server 4.	En este manual veremos paso a paso la forma de instalar en tu equipo JBoss Application Server 4.
WebServices con Axis y JBoss	En este tutorial os mostramos como realizar servicios web utilizando Axis y el contenedor de aplicaciones web JBoss
Trabajando con Axis	Utilizando Apache Axis, os mostramos otro interesante tutorial que ilustra su utilización para implementar web services