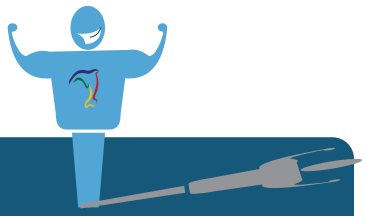
autentia

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...

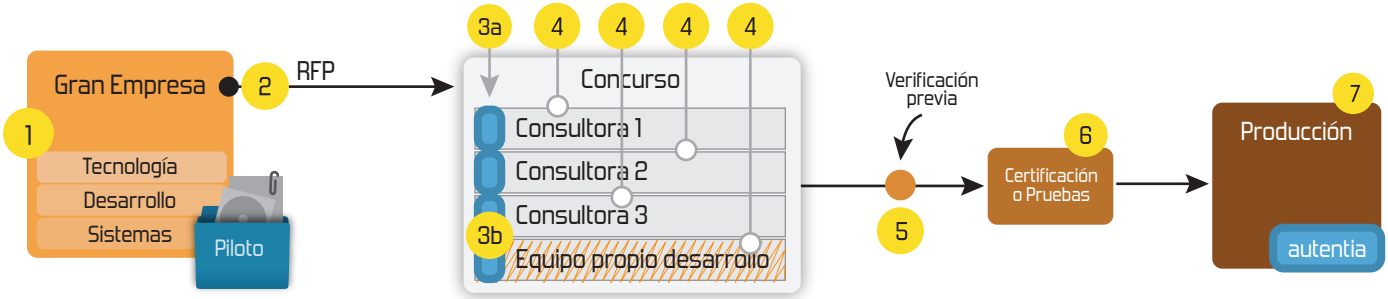
1. Desarrollo de componentes y proyectos a medida



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

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

Ver cursos que ofrece Autentia

Descargar comics en PDF y alta resolución

[¡NUEVO!] 2008-04-01

2008-03-25

2008-03-17

2008-03-11

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion comic.

Tutorial desarrollado por

Carlos García Pérez

Creador del pionero Web MobileTest.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática *

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.

autentia real business solutions

Catálogo de cursos

Descargar este documento en formato PDF: xmlSignature.pdf

Fecha de creación del tutorial: 2008-04-03

XML Signature - Firma Digital sobre XML

En este tutorial vamos a ver ejemplos de cómo firmar digitalmente y validar un documento.

Introducción a XML Signature

Con las firmas digitales conseguimos:

- **Integridad**: Si la información firmada es modificada, la validación de la firma fallará.
- **No repudiación**: La entidad que firmo la información no puede luego decir que no lo hizo.
- **Autentificación**: Sabemos qué entidad es la que realizó la firma, pues lo podemos validar con su clave pública.

Además podemos conseguir **confidencialidad** (cifrado) con el par de claves pública/privada, pues lo que se encripta con la clave privada sólo puede ser descifrado con la clave pública.

Para este tutorial utilizaremos la implementación de Apache (http://xml.apache.org/security/dist/).

Veamos unos ejemplos auto comentados:

1. Firma digitalmente un documento XML
2. Validación de un documento firmado digitalmente

Generamos una clave privada y su correspondiente certificado

La herramienta keytool crea pares de claves públicas y privadas, certificados auto firmados, y gestiona almacenes de claves. Creamos una clave pública/privada:

```
keytool -genkey -alias mi_cert_ejemplo -keyalg DSA -dname "CN=Usuario, O=Autentia, C=ES"
-keypass abc1234 -storepass abc12345 -keystore myKeyStore.jks -validity 365
```

Clase de utilidades usada en los ejemplos de este tutorial

Catálogo de servicios Autentia (PDF 6,2MB)

En formato comic...

Google

Web

www.adictosaltrabajo.com

Buscar

Últimos tutoriales

2008-04-05 JMX y monitorización de JBoss

2008-04-05 Jersey: la implemetación de RESTFull de Sun

2008-04-05 Metro: pila de webservices de Sun. Integración con Maven 2

2008-04-05 Metro: pila de webservices de Sun.

2008-04-04 Espectaculares efectos visuales en el escritorio de Linux, con Compiz Fusion

2008-04-04 Monitorización de Web Services con Glassfish Wsmonitor

2008-04-04 Axis2. Ejemplo de creación de un servicio Web

2008-04-03 Servicios Web RESTful en Axis 2

2008-04-03 XML Signature - Firma Digital sobre XML

2008-04-03 XML Encryption, Criptografía sobre XML

Anuncios Google Document Signatures Unprotect PDF Outlook Signature Sign XML Documents Electronic Signature

view plain print ?

```
package com.autentia.examples.xmlsignature;

import java.io.File;
import java.io.FileOutputStream;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;
import org.w3c.dom.Element;

/**
 * Clase de utilidad para el ejemplo
 * @author Carlos García. Autentia.
 * @see http://www.mobiletest.es
 */
public class DOMUtils {

    /**
      * Serializa un objeto Document en un archivo
      */
    public static void outputDocToFile(Document doc, File file)  throws Exception {
        FileOutputStream    f              =  new FileOutputStream(file);
        TransformerFactory factory         = TransformerFactory.newInstance();
        Transformer        transformer     = factory.newTransformer();

        transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION,  "yes");

        transformer.transform( new DOMSource(doc), new StreamResult(f));

        f.close();
    }

    /**
      * Lee un Document desde un archivo
      */
    public static Document loadDocumentFromFile(java.io.File file)  throws Exception {
        DocumentBuilderFactory  factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder         builder =  null;

        factory.setNamespaceAware( true);

        builder = factory.newDocumentBuilder();

        return builder.parse(file);
    }

    /**
      * @return Crea un elemento <tag>value</tag>
      */
    public static Element createNode(Document document, String tag, String value){
        Element node = document.createElement(tag);
        if (value != null){
            node.appendChild(document.createTextNode(value));
        }
        return node;
    }

    /**
      * @return Devuelve un Document a firmar
      * @throws Exception Cualquier incidencia
      */
    public static Document createSampleDocument()  throws Exception {
        DocumentBuilderFactory  factory  = DocumentBuilderFactory.newInstance();
        DocumentBuilder         builder  = factory.newDocumentBuilder();
        Document                document = builder.newDocument();

        Element person = document.createElement( "persona");
        person.setAttribute( "id", "468300000");

        person.appendChild(DOMUtils.createNode(document,  "nombre",    "Pepito"));
        person.appendChild(DOMUtils.createNode(document,  "apellidos", "Pérez Luna"));
        person.appendChild(DOMUtils.createNode(document,  "email",     "pepito.perez@servidor.com" ));

        document.appendChild(person);

        return document;
    }
}
```

Ejemplo. Firmamos digitalmente el documento XML

Últimas ofertas de empleo

2008-04-04
Banca - Genérico - MADRID.

2008-04-03
Banca - Genérico - MADRID.

2008-04-02
T. Información - Analista / Programador - MADRID.

2008-04-02
T. Información - Analista / Programador - MADRID.

2008-03-29
T. Información - Analista / Programador - MADRID.

Anuncios Google

view plain print ?

```
package com.autentia.examples.xmlsignature;

import java.io.File;
import java.io.FileInputStream;
import java.security.KeyStore;
import java.security.PrivateKey;
import java.security.cert.X509Certificate;


import org.w3c.dom.*;
import org.apache.xml.security.signature.XMLSignature;
import org.apache.xml.security.transforms.Transforms;
import org.apache.xml.security.utils.Constants;


/**
 * Firma un documento XML
 * @author Carlos García. Autentia.
 * @see http://www.mobiletest.es
 */
public class CreateSignature {

    private static final String KEYSTORE_TYPE         = "JKS";
    private static final String KEYSTORE_FILE         = "myKeyStore.jks";
    private static final String KEYSTORE_PASSWORD     = "abc12345";
    private static final String PRIVATE_KEY_PASSWORD  = "abc1234";
    private static final String PRIVATE_KEY_ALIAS     = "mi_cert_ejemplo";



    /**
      * Punto de entrada al ejemplo
      */
    public static void main(String args[]) throws Exception {
        org.apache.xml.security.Init.init();

        Document doc = DOMUtils.createSampleDocument();

        Constants.setSignatureSpecNSprefix( ""); // Sino, pone por defecto como prefijo: "ns"


        // Cargamos el almacen de claves
        KeyStore ks  = KeyStore.getInstance(KEYSTORE_TYPE);
        ks.load( new FileInputStream(KEYSTORE_FILE), KEYSTORE_PASSWORD.toCharArray());

        // Obtenemos la clave privada, pues la necesitaremos para encriptar.
        PrivateKey privateKey = (PrivateKey) ks.getKey(PRIVATE_KEY_ALIAS, PRIVATE_KEY_PASSWORD.toCharArray());
        File    signatureFile =  new File("signature.xml");
        String  baseURI       = signatureFile.toURL().toString();    // BaseURI para las URL Relativas.

        // Instanciamos un objeto XMLSignature desde el Document. El algoritmo de firma será DSA
        XMLSignature xmlSignature =  new XMLSignature(doc, baseURI, XMLSignature.ALGO_ID_SIGNATURE_DSA);


        // Añadimos el nodo de la firma a la raiz antes de firmar.
        // Observe que ambos elementos pueden ser mezclados en una forma con referencias separadas
        doc.getDocumentElement().appendChild(xmlSignature.getElement());

        // Creamos el objeto que mapea: Document/Reference
        Transforms transforms =  new Transforms(doc);
        transforms.addTransform(Transforms.TRANSFORM_ENVELOPED_SIGNATURE);

        // Añadimos lo anterior Documento / Referencia
        // ALGO_ID_DIGEST_SHA1 = "http://www.w3.org/2000/09/xmldsig#sha1";
        xmlSignature.addDocument( "", transforms, Constants.ALGO_ID_DIGEST_SHA1);

        // Añadimos el KeyInfo del certificado cuya clave privada usamos
        X509Certificate cert = (X509Certificate) ks.getCertificate(PRIVATE_KEY_ALIAS);
        xmlSignature.addKeyInfo(cert);
        xmlSignature.addKeyInfo(cert.getPublicKey());


        // Realizamos la firma
        xmlSignature.sign(privateKey);

        // Guardamos archivo de firma en disco
        DOMUtils.outputDocToFile(doc, signatureFile);
    }
}
```

Documento firmado digitalmente (`signature.xml`).

En el siguiente documento se muestra el documento firmado digitalmente, en donde se puede observar:

1. La información relacionada con la tarjeta de crédito ha sido sustituida por el elemento **`xenc:EncryptedData`**

view plain print ?

```
<persona id="468300000">
    <nombre>Pepito</nombre>
    <apellidos>Pérez Luna</apellidos>
    <email>pepito.perez@servidor.com</email>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <CanonicalizationMethod
                Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
                xmlns="http://www.w3.org/2000/09/xmldsig#" />
            <SignatureMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"
                xmlns="http://www.w3.org/2000/09/xmldsig#" />
            <Reference URI=""
                xmlns="http://www.w3.org/2000/09/xmldsig#">
                <Transforms
                    xmlns="http://www.w3.org/2000/09/xmldsig#">
                    <Transform
                        Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
                        xmlns="http://www.w3.org/2000/09/xmldsig#" />
                </Transforms>
                <DigestMethod
                    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
                    xmlns="http://www.w3.org/2000/09/xmldsig#" />
                <DigestValue
                    xmlns="http://www.w3.org/2000/09/xmldsig#">
                    Oyyx+K28+cp7kuUgcnANtTBdUwg=
                </DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">
            ZVzRud7G4mEZsDnBavbnZoFUmm5J2OBDkQ+IooDLn95ndGYdrq6uPQ==
        </SignatureValue>
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <X509Data xmlns="http://www.w3.org/2000/09/xmldsig#">
                <X509Certificate
                    xmlns="http://www.w3.org/2000/09/xmldsig#">
                    MIICmDCCAlYCBEfrim8wCwYHKoZIzjgEAwUAMDIxCzAJBgNVBAYTAkVTMREwDwYDVQQKEwhBdXRl
                    bnRpYTEQMA4GA1UEAxMHVXN1YXJpbzAeFw0wODAzMjcxMTUyMTVaFw0wOTAzMjcxMTUyMTVaMDIx
                    CzAJBgNVBAYTAkVTMREwDwYDVQQKEwhBdXRlbnRpYTEQMA4GA1UEAxMHVXN1YXJpbzCCAbcwggEs
                    BgcqhkjOOAQBMIIBHwKBgQD9f1OBHXUSKVLfSpwu7OTn9hG3UjzvRADDHj+AtlEmaUVdQCJR+1k9
                    jVj6v8X1ujD2y5tVbNeBO4AdNG/yZmC3a5lQpaSfn+gEexAiwk+7qdf+t8Yb+DtX58aophUPBPuD
                    9tPFHsMCNVQTWhaRMvZ1864rYdcq7/IiAxmd0UgBxwIVAJdgUI8VIwvMspK5gqLrhAvwWBz1AoGB
                    APfhoIXWmz3ey7yrXDa4V7l5lK+7+jrqgvlXTAs9B4JnUVlXjrrUWU/mcQcQgYC0SRZxI+hMKBYT
                    t88JMozIpuE8FnqLVHyNKOCjrh4rs6Z1kW6jfwv6ITVi8ftiegEkO8yk8b6oUZCJqIPf4VrlnwaS
                    i2ZegHtVJWQBTDv+z0kqA4GEAAKBgDUPDwxDZFXMrZha74VNmgyFslLM01wKw17nbt9UFTJALk71
                    iPpozeZMP2u0SoYst2nbxkCs1hziiuaNJnykzcjVf3+PmL3sQES8SxwJBRUME2UTA2006WD3xNy9
                    iZ9yibcWQimB8eKIJyBBxSk5TueAzvTA8HN2+Rvgh8RMaOzhMAsGByqGSM44BAMFAAMvADAsAhQH
                    4+nQZdFvlvsfyOfq1t02h9MJEgIUEvYDfxeygKCmrIlA0sQLtaCs0Qo=
                </X509Certificate>
            </X509Data>
            <KeyValue xmlns="http://www.w3.org/2000/09/xmldsig#">
                <DSAKeyValue
                    xmlns="http://www.w3.org/2000/09/xmldsig#">
                    <P xmlns="http://www.w3.org/2000/09/xmldsig#">
                        /X9TgR11EilS30qcLuzk5/YRt1I870QAwx4/gLZRJmlFXUAiUftZPY1Y+r/F9bow9subVWzXgTuA
                        HTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOu
                        K2HXKu/yIgMZndFIAcc=
                    </P>
                    <Q xmlns="http://www.w3.org/2000/09/xmldsig#">
                        l2BQjxUjC8yykrmCouuEC/BYHPU=
                    </Q>
                    <G xmlns="http://www.w3.org/2000/09/xmldsig#">
                        9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT+ZxBxCBgLRJFnEj6EwoFhO3
                        zwkyjMim4TwWeotUfI0o4KOuHiuzpnWRbqN/C/ohNWLx+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKL
                        Zl6Ae1UlZAFMO/7PSSo=
                    </G>
                    <Y xmlns="http://www.w3.org/2000/09/xmldsig#">
                        NQ8PDENkVcytmFrvhU2aDIWyUszTXArDXudu31QVMkAuTvWI+mjN5kw/a7RKhiy3advGQKzWHOKK
                        5o0mfKTNyNV/f4+YvexARLxLHAkFFQwTZRMDbTTpYPfE3L2Jn3KJtxZCKYHx4ognIEHFKTlO54DO
                        9MDwc3b5G+CHxExo7OE=
                    </Y>
                </DSAKeyValue>
            </KeyValue>
        </KeyInfo>
    </Signature>
</persona>
```

Ejemplo. Validamos el documento XML digitalmente firmado (`signature.xml`)

En el siguiente ejemplo vamos a validar el documento que ha sido firmado en el ejemplo anterior.

view plain print ?

```
package com.autentia.examples.xmlsignature;


import java.io.File;
import java.security.PublicKey;
import java.security.cert.X509Certificate;
import org.apache.xml.security.keys.KeyInfo;
import org.apache.xml.security.signature.XMLSignature;
import org.w3c.dom.*;

import javax.xml.parsers.*;

/**
 * Verifica un archivo firmado
 * @author Carlos García. Autentia.
 * @see http://www.mobiletest.es
 */
public class VerifySignature {

    /**
      * Punto de inicio
      */
    public static void main(String args[]) throws Exception {
        org.apache.xml.security.Init.init();

        String signatureFileName =  "signature.xml";

        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        dbf.setNamespaceAware( true);
        dbf.setAttribute( "http://xml.org/sax/features/namespaces" , Boolean.TRUE);

        File            f   = new File(signatureFileName);
        DocumentBuilder db  = dbf.newDocumentBuilder();
        Document     doc    = db.parse( new java.io.FileInputStream(f));
        Element      sigElement = (Element) doc.getElementsByTagName( "Signature").item(0);
        XMLSignature signature  = new XMLSignature(sigElement, f.toURL().toString());

        KeyInfo keyInfo = signature.getKeyInfo();
        if (keyInfo != null) {
            X509Certificate cert = keyInfo.getX509Certificate();
            if (cert != null) {
                // Validamos la firma usando un certificado X509
                if (signature.checkSignatureValue(cert)){
                    System.out.println( "Válido según el certificado" );
                } else {
                    System.out.println( "Inválido según el certificado" );
                }
            } else {
                // No encontramos un Certificado intentamos validar por la cláve pública
                PublicKey pk = keyInfo.getPublicKey();
                if (pk != null) {
                    // Validamos usando la clave pública
                    if (signature.checkSignatureValue(pk)){
                        System.out.println( "Válido según la clave pública" );
                    } else {
                        System.out.println( "Inválido según la clave pública" );
                    }
                } else {
                    System.out.println( "No podemos validar, tampoco hay clave pública" );
                }
            }
        } else {
            System.out.println( "No ha sido posible encontrar el KeyInfo" );
        }
    }

}
```

Como prueba, modifique el documento y observará que el proceso de validación falla debido a que no se cumple la propiedad de integración de la información.

Saludos, Carlos García.

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati.

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

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿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?

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

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Formación en nuevas tecnologías

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail [ ] Aceptar

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Firmar applets usando keytool y jarsigner	En este tutorial os mostramos como firmar un applet con las herramientas incluidas en el JDK estándar	2006-01-02	7892	pdf
Firma digital de un Applet	Para que un applet Java pueda ejecutarse en un cliente Web con la configuración de seguridad por defecto y/o adquirir privilegios de seguridad, es necesario firmarla digitalmente. Alejandro Perez nos enseña como hacerlo de un modo rápido y sencillo.	2004-06-01	15399	pdf
Soporte XML en Eclipse con X-MEN	Alejandro Perez nos enseña como potenciar el entorno eclipse para facilitarnos el trabajo con ficheros xml, gracias al pluggin X-MEN	2003-12-27	17077	pdf
Validando XML contra Schema	En este tutorial nuestro compañero describe las posibilidades de validación de un documento XML con Schema	2008-01-31	917	pdf
Firmar Applets Java para MS Internet Explorer	Cristhian Herrera nos enseña a firmar un Applets Java, para adquirir privilegios, en Internet Explorer, usando herramientas Microsoft	2005-05-18	12406	pdf
XML básico	Si quieres ver de un modo visual como crear un documento XML, este es tu tutorial. Este es el primero de un conjunto de tutoriales que iremos publicando sobre esta fascinante y amplia tecnología	2003-06-10	21700	pdf
Firmas Digitales muy Fácil con Firefox	En este tutorial vamos a ver como poder firmar documentos (certificados digitales) usando FireFox	2007-06-27	2552	pdf
Schemas XML. Introducción esquemas XML	Los esquemas XML (schemas XML) son una evolución natural de las DTDs. Os mostramos como emprezar con esta tecnología.	2003-12-17	17625	pdf
XML y XSL en Cliente	En este tutorial os enseñamos como formaterar documentos XML directamente en vuestro navegador a través de Plantillas XSL. En cursos sucesivos veremos como hacerlo en el servidor, para no crear dependencias con el navegador del cliente.	2003-06-11	16069	pdf
Transformación de XML y XSL en JSPs	Os mostramos como poder utilizar XML y XSL en JSPS, combinado con el Patrón MVC	2003-12-06	25683	pdf

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.