

# ¿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)



Hosting Patrocinado por  
**enREDados.com**  


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

	<p><b>Tutorial desarrollado por:</b>  <b>Roberto Canales Mora 2003-2005</b>  <a href="#">Creador de AdictosAlTrabajo.com</a> y</p> <p><b>Director General de Autentia S.L.</b></p> <p><b>Recuerda que me puedes contratar para echarte una mano:</b></p> <p>Desarrollo y arquitectura Java/J2EE          Asesoramiento tecnológico Web          Formación / consultoría integrados en tu proyecto</p> <p>No te cortes y contacta: 655 99 11 72 <a href="mailto:rcanales@autentia.com">rcanales@autentia.com</a>.</p>	
---	--	---

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

**Curso Web J2EE**

Curso Avanzado en Desarrollo Web con J2EE

**JSP Tutoriales**

JSP Made Easy With XMLSpy, Syntax & Editing Help, Free D/L.

**IntelliJ IDEA**

Professional Java IDE for professional developers. Get Trial!

**Download Java Report Tool**

Drag & drop Report creation for Java applications. Powerful charts!

Anuncios Goooooogle

Anunciarse en este sitio

## JSP 2.0 y EL (Expresion Lenguaje)

La especificación 2.0 de JSP ha introducido una nueva librería estandar de etiquetas, denominada JSTL.

Estas etiquetas tratan de abstraer la complejidad de introducir código Java (scriptlet) dentro de JSP, del mismo modo que trata de evitar que cada equipo de desarrollo cree un juego de etiquetas no estandar para las mismas labores.

Dentro de estas etiquetas, se utiliza un lenguaje llamado EL, lenguaje de expresiones, que pretende ser un lenguaje más sencillo que Java, para realizar operaciones.

Uno de los primeros contenedores de JSP que soporta estas capacidades es Tomcat5 ([ver tutorial que describe su instalación](#)).

### Fichero descriptor

Lo primero que vamos a hacer es cambiar el fichero descriptor, copiando el que viene en los ejemplos. Si intentais usar uno que ya tengais, no se entera el Tomcat que tiene que interpretar los nuevos elementos de EL.

Fijaos en algunas lineas:

Indicamos que queremos interpretar EL	<code>&lt;el-ignored&gt;false&lt;/el-ignored&gt;</code>
Indicamos que no soportamos scriptlet	<code>&lt;scripting-invalid&gt;true&lt;/scripting-invalid&gt;</code>
Podemos poner una cabecera por defecto	<code>&lt;include-prelude&gt;prelude.jspf&lt;/include-prelude&gt;</code>
Y tambien un pie	<code>&lt;include-coda&gt;coda.jspf&lt;/include-coda&gt;</code>

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee web-app_2_4.xsd"
version="2.4">

<description>Ejemplos de EL de rcanales@adictosaltrabajo.com</description>
<display-name>Ejemplos de JSP 2.0 </display-name>

<jsp-config>
<jsp-property-group>
<description>Definicion de atributos</description>
<display-name>ConfiguracionJSP</display-name>
<url-pattern>/*</url-pattern>
<el-ignored>false</el-ignored>
<page-encoding>ISO-8859-1</page-encoding>
<scripting-invalid>true</scripting-invalid>
<include-prelude>prelude.jspf</include-prelude>
<include-coda>coda.jspf</include-coda>
  
```

```
</jsp-property-group>
</jsp-config>
</web-app>
```

Si ahora nos vamos al directorio raíz y creamos un fichero que se llame **el.jsp** e intentamos crear un scriptlet .... veremos que no nos deja ejecutar.

Es, decir, está prohibido introducir código Java `<% out.println("Mostramos un Scriptlet"); %>`

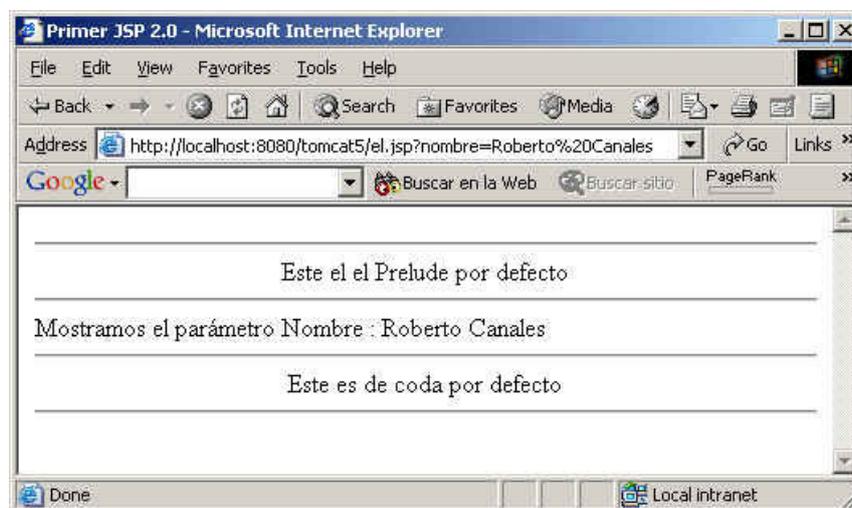
Y también podemos ver que ya introducimos la primera sentencia EL

```
<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>

<% out.println("Mostramos un Scriptlet"); %>
Mostramos el parámetro Nombre : {param.nombre}

</body>
</html>
```

Si eliminamos el scriptlet ..... veremos la respuesta



## Objetos Implícitos

Bueno, parece que las expresiones están dentro de `{expression}`

Para que sea útil este sistema, existen unos objetos implícitos que nos permiten acceder de un modo sencillo a los parámetros, contexto, petición, etc...

<b>pageScope</b>	requestScope	<b>sessionScope</b>	applicationScope
param	<b>paramValues</b>	header	<b>headerValues</b>
<b>pageContext</b>	initParam	<b>cookie</b>	

Y para acceder a ellos y sus miembros, ya hemos visto que podemos hacerlo con el (.)

## Operaciones

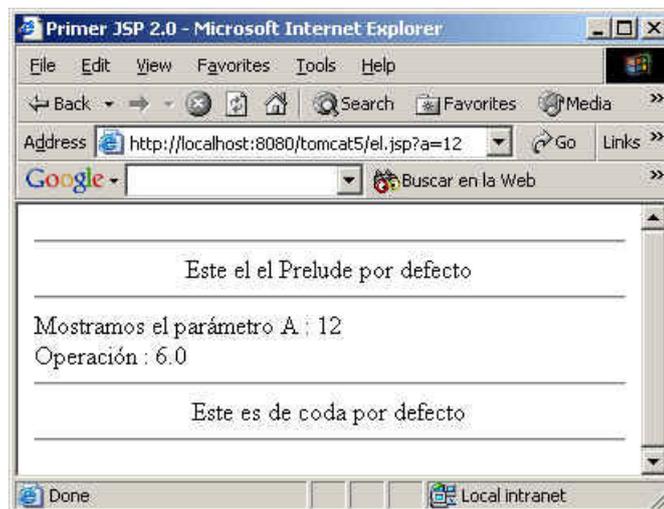
Podemos, de un modo sencillo e implícito realizar conversiones desde los parámetros recibidos a otros tipos de datos

```
<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>

<%-- <% out.println("Mostramos un Scriptlet"); %> --%>
Mostramos el parámetro A : {param.a}

<br>Operación : {param.a / 2}
</body>
</html>
```

Y la salida es:

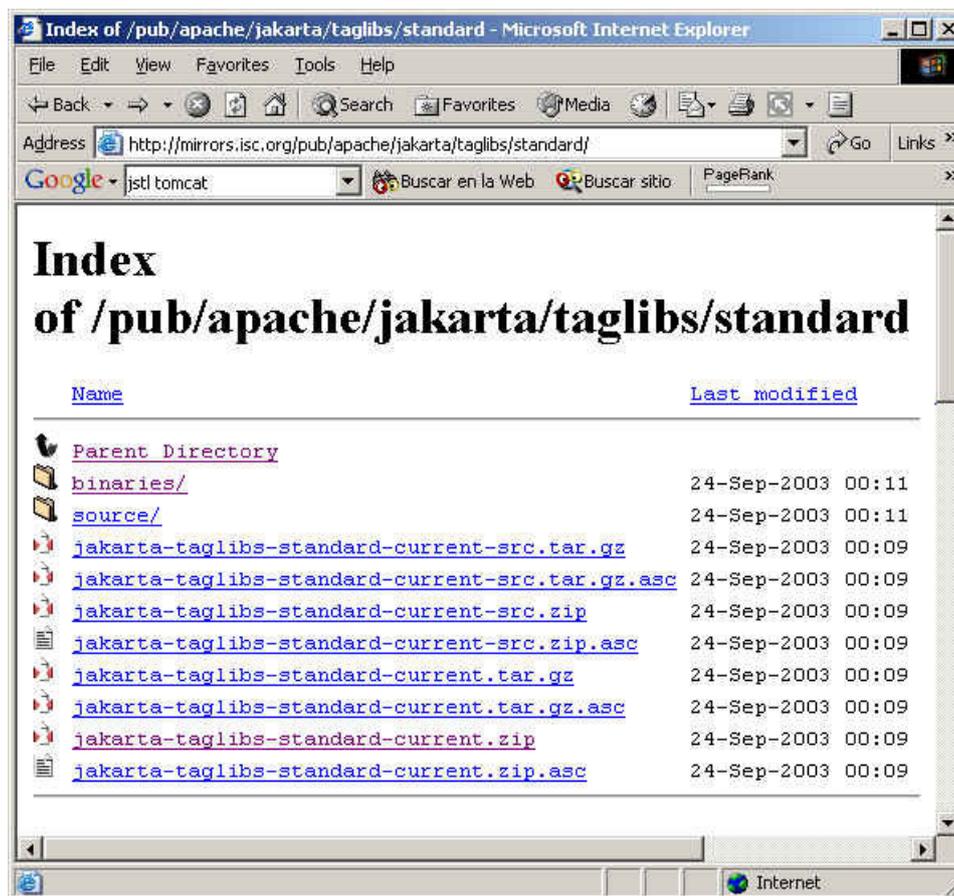


## Descarga y Uso de TagLibraries

Para poder usar las librerías de tags de JSTL, debemos descargarnos los ficheros binarios y registrarlos en el fichero web.xml

<http://jakarta.apache.org/downloads/binindex.html>.

<http://mirrors.isc.org/pub/apache/jakarta/taglibs/standard/>



Descargamos y descomprimos el fichero marcado.

Los ficheros con extensión **.tld**, los copiamos a nuestro directorio **WEB\_INF** y los **jar**, al directorio **WEB\_INF/lib**

Ahora modificamos el fichero **web.xml** para añadir la referencia a las librerías de etiquetas.

```
<taglib>
  <taglib-uri>http://java.sun.com/jstl/fmt</taglib-uri>
  <taglib-location>/WEB-INF/fmt.tld</taglib-location>
</taglib>

<taglib>
  <taglib-uri>http://java.sun.com/jstl/fmt-rt</taglib-uri>
```

```

<taglib-location>/WEB-INF/fmt-rt.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/core</taglib-uri>
<taglib-location>/WEB-INF/c.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/core-rt</taglib-uri>
<taglib-location>/WEB-INF/c-rt.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/sql</taglib-uri>
<taglib-location>/WEB-INF/sql.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/sql-rt</taglib-uri>
<taglib-location>/WEB-INF/sql-rt.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/x</taglib-uri>
<taglib-location>/WEB-INF/x.tld</taglib-location>
</taglib>

<taglib>
<taglib-uri>http://java.sun.com/jstl/x-rt</taglib-uri>
<taglib-location>/WEB-INF/x-rt.tld</taglib-location>
</taglib>

```

Debemos rearrancar Tomcat para que recoja los cambio (bueno, con rearrancar la Webapp es suficiente)

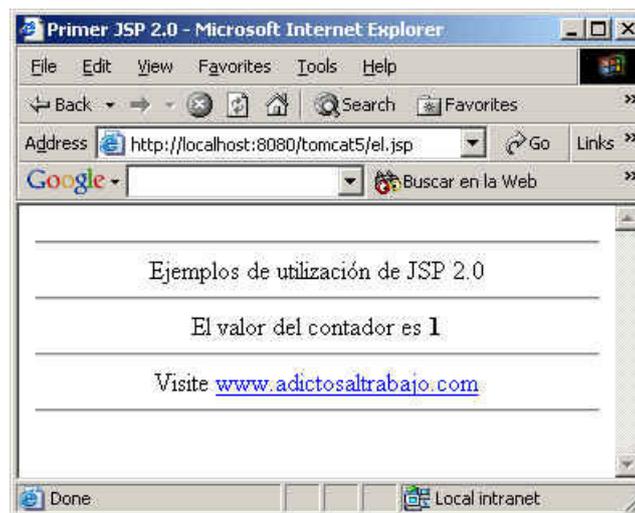
## JSPs con TagLibs

Introducimos la directiva para usar la librería de etiquetas (he elegido el prefijo jsp2 para las etiquetas).

```

<%@ taglib prefix="jsp2" uri="http://java.sun.com/jstl/core" %>
<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>
<center>
<jsp2:set var="contador" scope="session" value="1"/>
El valor del contador es <b> ${contador} </b>
</center>
</body>
</html>

```



Ahora solo tenemos que saber las etiquetas que hay disponibles. Podeis encontrar una buena referencia en <http://www.manning.com/bayern/appendixA.pdf>

## JSP y XML

Una de las preguntas que se hace mucha gente a la hora de utilizar la tecnología J2EE es ¿debería utilizar JSP o XSLT para la capa de presentación?

La misma pregunta nos puede surgir al utilizar Frameworks, como struts.

Un modo sencillo de combinar estas tecnologías puede ser utilizar etiquetas que permitan dentro de JSP introducir el procesamiento de XML.

```
<%@ taglib prefix="jsp2" uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="jsp2x" uri="http://java.sun.com/jsp/jstl/xml" %>

<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>
<center>

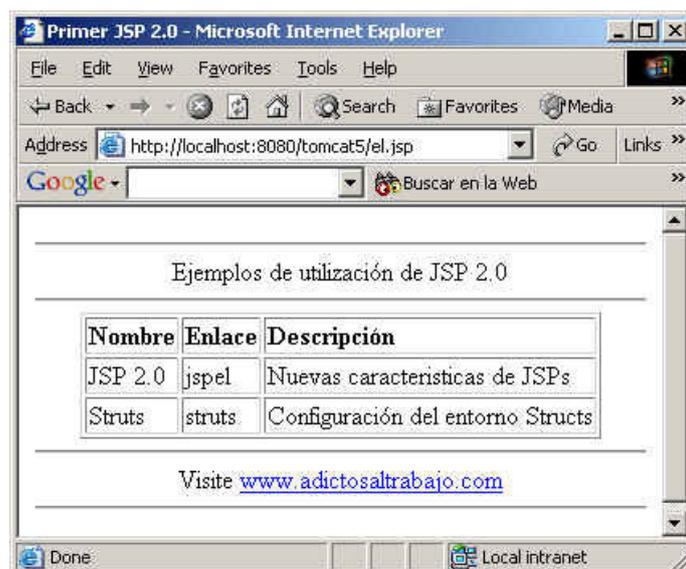
<jsp2:set var="mixsl">
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:template match="/">
    <table border="1">
      <tr>
        <td><b>Nombre</b></td>
        <td><b>Enlace</b></td>
        <td><b>Descripción</b></td>
      </tr>
      <xsl:for-each select="//tutorial">
        <tr>
          <td><xsl:value-of select="nombre"/></td>
          <td><xsl:value-of select="enlace"/></td>
          <td><xsl:value-of select="descripcion"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </xsl:template>
</xsl:stylesheet>
</jsp2:set>

<jsp2x:transform xslt="{mixsl}">
  <tutoriales>
    <tutorial>
      <nombre>JSP 2.0</nombre>
      <enlace>jspel</enlace>
      <descripcion>Nuevas características de JSPs</descripcion>
    </tutorial>
    <tutorial>
      <nombre>Struts</nombre>
      <enlace>struts</enlace>
      <descripcion>Configuración del entorno Struts</descripcion>
    </tutorial>
  </tutoriales>
</jsp2x:transform>

</center>
</body>
</html>
```

Y la salida sería



Los ficheros no tienen que estar dentro del documento ... pueden ser ficheros normales, externos.

## Acceso a Base de datos

Es igual de sencillo realizar accesos a la base de datos... usaremos la configuración vista en el tutorial sobre [configuración de Pool de conexiones en Tomcat 5](#)

```

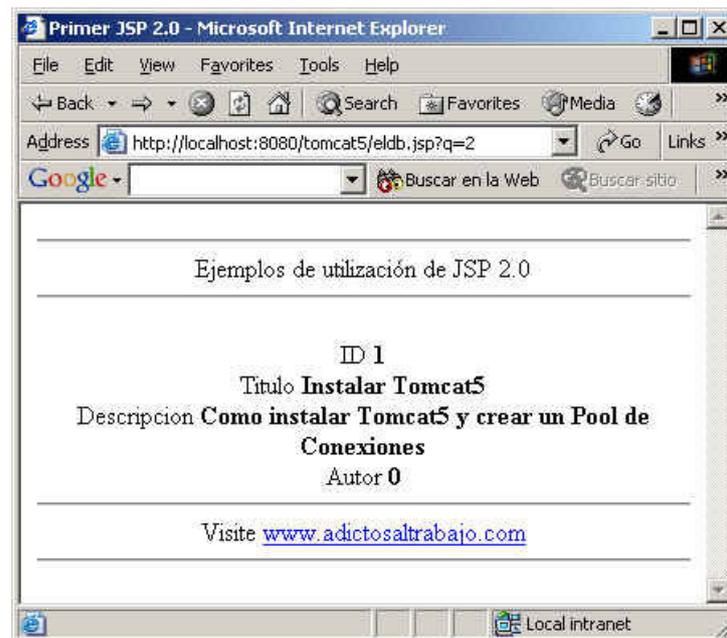
<%@ taglib prefix="sql" uri="http://java.sun.com/jstl/sql" %>
<%@ taglib prefix="jsp2" uri="http://java.sun.com/jstl/core" %>
<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>
<center>
<sql:setDataSource dataSource="jdbc/tutoriales" />

<sql:query var="r">
SELECT * FROM TUTORIALES where id=1
</sql:query>

<br>ID <b>${r.rows[0].id}</b>
<br>Titulo <b>${r.rows[0].titulo}</b>
<br>Descripcion <b>${r.rows[0].descripcion}</b>
<br>Autor <b>${r.rows[0].autor}</b>

</center>
</body>
</html>

```



También podemos cambiar un poquito el ejemplo para iterar. En este caso utilizamos `forEach` (ver que he cambiado la librería `core` por `core-rt`, sino, no me funcionaba)

```

<%@ taglib prefix="sql" uri="http://java.sun.com/jstl/sql" %>
<%@ taglib prefix="jsp2" uri="http://java.sun.com/jstl/core_rt" %>
<html>
<head>
<title>Primer JSP 2.0</title>
</head>
<body>
<center>
<sql:setDataSource dataSource="jdbc/tutoriales" />

<sql:query var="r">
SELECT * FROM TUTORIALES where id<3
</sql:query>

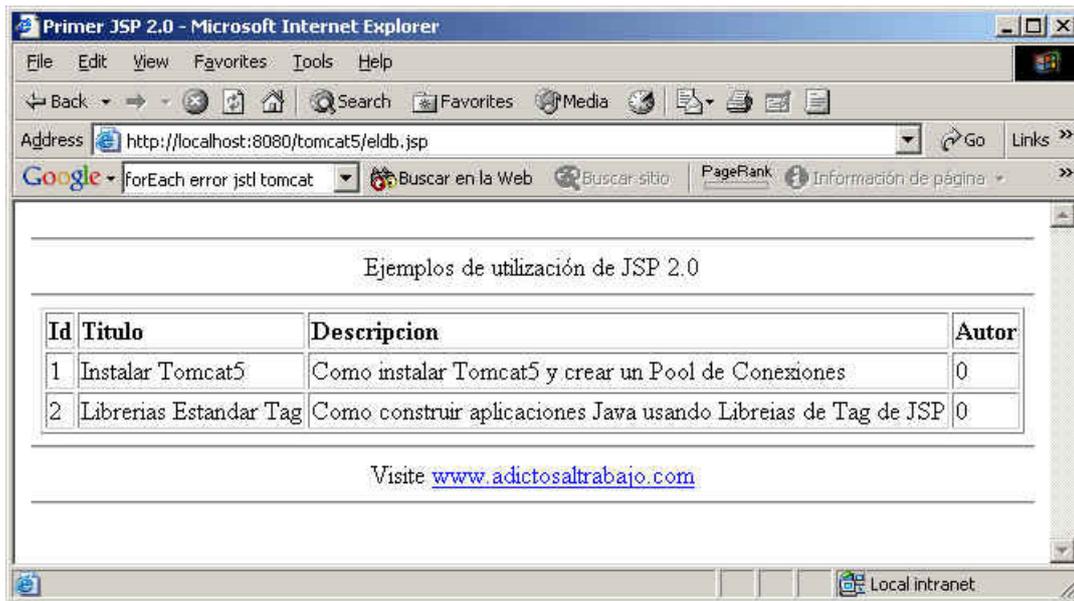
<table border="1">
<tr>
<td><b>Id</b></td>
<td><b>Titulo</b></td>
<td><b>Descripcion</b></td>
<td><b>Autor</b></td>
</tr>

<jsp2:forEach items="${r.rows}" var="fila">
<tr>
<td>${fila.id}</td>
<td>${fila.titulo}</td>
<td>${fila.descripcion}</td>
<td>${fila.autor}</td>
</tr>
</jsp2:forEach>

</table>
</center>
</body>

```

&lt;/html&gt;



### Conclusión

Básicamente, parece que se está intentando limitar las capacidades dentro de JSP (de un modo controlado y voluntario), con el objetivo de hacer mucho más accesible esta tecnología a programadores de otros entornos .... como por ejemplo programadores PHP.

Estoy seguro que esta tecnología va a tener muy buena acogida porque realmente permitirá separar roles en equipos de desarrollo en aplicaciones empresariales.

Existen muchas más librerías de etiquetas (en apache.org encontrareis un montón) que seguro que se van a ir estandarizando .... y ya os la iremos mostrando poco a poco....

[Sobre el Autor ..](#)

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con



[Autentia S.L.](#) Somos expertos en:  
**J2EE, C++, OOP, UML, Vignette, Creatividad ..**  
 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	
<b>e-mail</b>	<input type="text"/>
	<input type="button" value="Enviar"/>

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

### Nombre Corto

[Reingeniería JDO con Druid](#)

[JDO con OJB](#)

[Cachear porciones de JSPs](#)

[JSP's y Modelo-Vista-Controlador](#)

[Otra implementación JDO con TJDO](#)

### Descripción

Os mostramos como crear vuestras clases y descriptores JDO, de tablas existentes, con la herramienta gratuita Druid.

Os mostramos como configurar el entorno OJB de apache para construir la primera aplicación JDO

En este tutorial os enseñamos como incrementar increíblemente el rendimiento de vuestro Web basado en tecnología JSP con el FrameWork de cache OSCACHE

En este tutorial os enseñamos como crear un JSP, su relación con los servlets y como crear un ejemplo MVC en Tomcat

Os mostramos como montar un ejemplo simple de JDO, a través de la implementación

<a href="#">gratuita TJDO</a>	Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.
<a href="#">Novedades en Java 1.5</a>	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)
<a href="#">Forzar diálogo Guardar Como en JSPs</a>	En este tutorial os enseñamos como formatear 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.
<a href="#">XML y XSL en Cliente</a>	Os mostramos como construir una aplicación con JSP que acceda a MySQL
<a href="#">Aplicaciones con JSPs</a>	Os mostramos como realizar ciertas labores intermedias en JSPs: Comentarios, gestión de errores, formateo de fechas y precompilación de ficheros
<a href="#">Gestión errores en JSPs</a>	

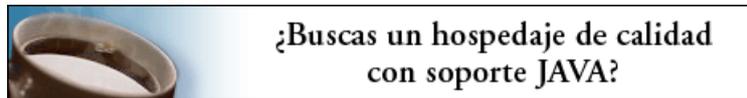
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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