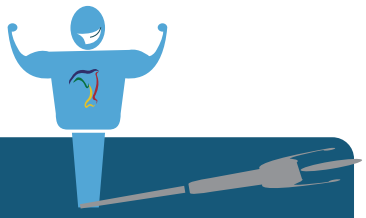
autentia

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...

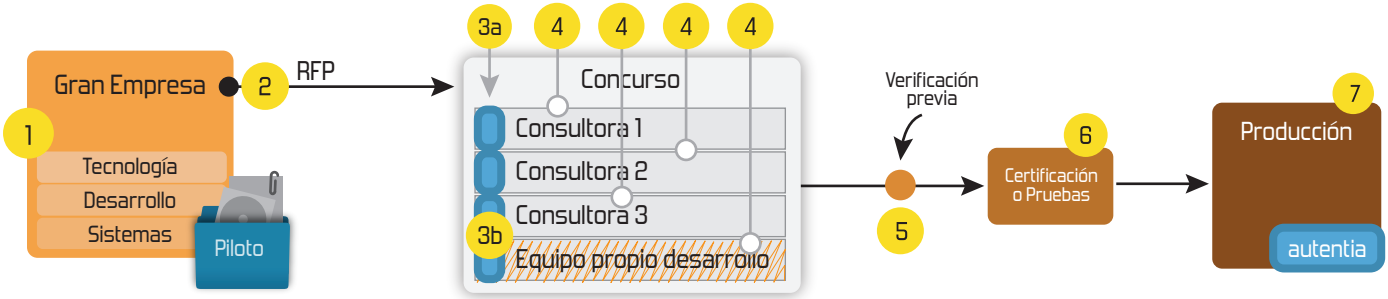
1. Desarrollo de componentes y proyectos a medida



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:
Roberto Canales Mora 2003-2004.

Si te gusta lo que ves, **puedes contratarme** para impartir **cursos presenciales** en tu empresa o para ayudarte en proyectos (Madrid).

Estamos creando las bases de la empresa en la que seguro te gustaría trabajar ... ayudanos a hacerla crecer ... presentándonos a tus jefes

Contacta: rcanales@autentia.com.

Una vez escuche esta frase:

Cuando todo esta bajo control, es que no vamos suficientemente deprisa

La mayoría de las veces construimos estos tutoriales al mismo tiempo que recordamos como hacer cosas o probamos nuevas tecnologías, poniendo más celo en el contenido que en la forma (cosa que veo que perturba a muchos de los lectores por lo que os pido disculpas). Es cuestión de criterios .... aunque trataremos de mejorar.

Espero que seáis comprensivos y generosos ayudando a encontrar las erratas, faltas de ortografía, enlaces rotos u otros posibles errores en estas páginas. Escribidme siempre que encontréis algun error y pronto ya no habrá .... y os estaremos francamente agradecidos. Este es tu Web .... colabora libremente ...

Roberto Canales

Descargar este documento en formato PDF jsps.pdf

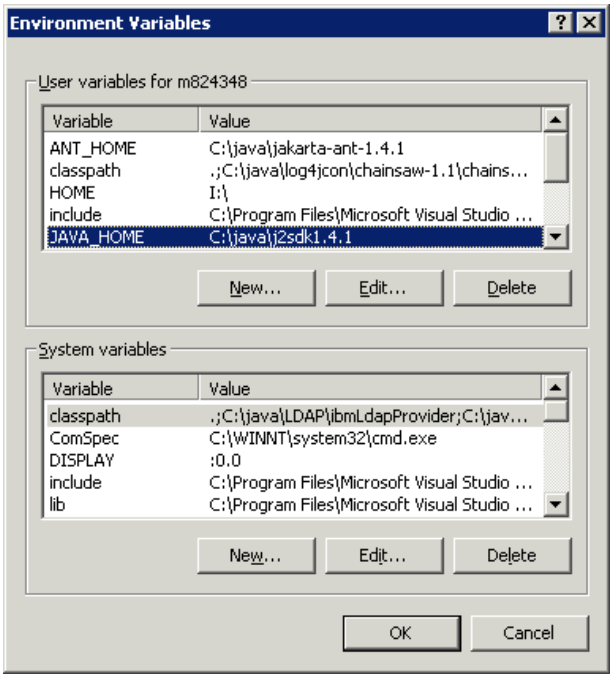
JSPs en Tomcat y MVC

La creación de JSP es algo bastante trivial. Vamos a describir es este tutorial como se construyen JSPs y se ejecutan en Tomcat.

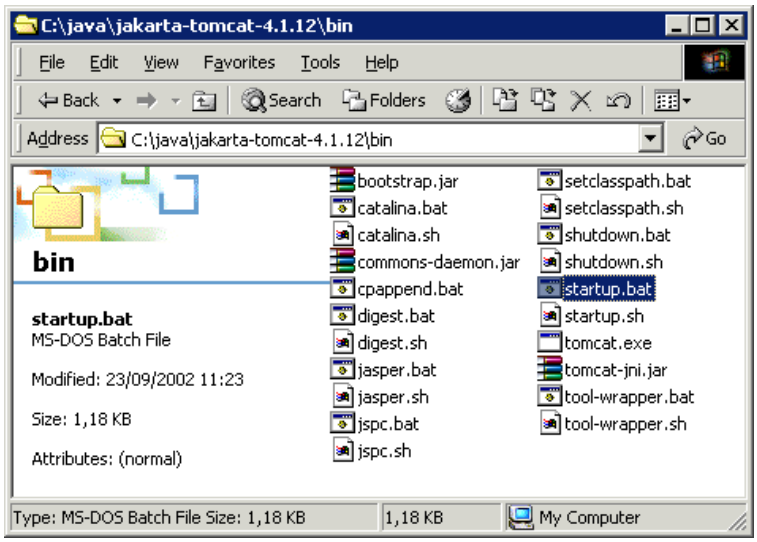
Posteriormente hablaremos del famoso modelo vista controlado .... y de temas más avanzados como las librerías de Tags

Lo primero es descargarse Tomcat y arrancarlo.

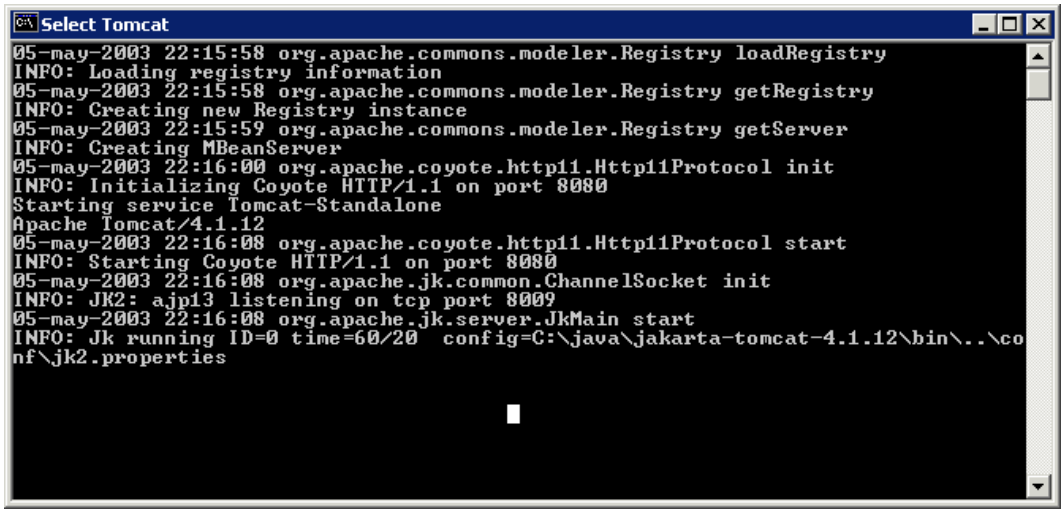
No olvidar que tenemos que tener JAVA_HOME apuntando al directorio donde tenemos instalado el JDK



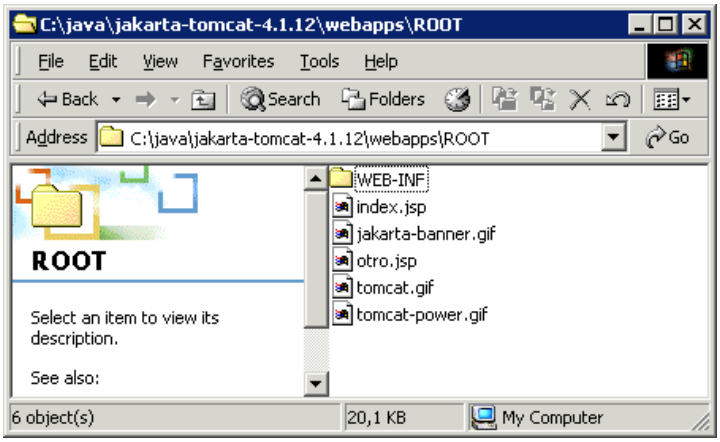
Ejecutamos el comando startup.bat



Y comprobamos que todo funciona correctamente

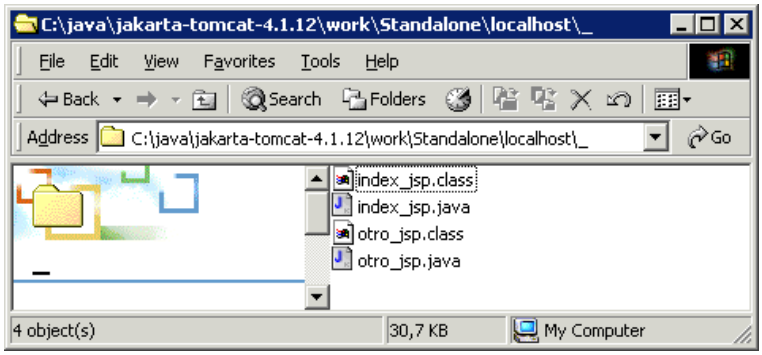


Empezamos construyendo nuestro primer JSP. Lo vamos a llamar **otro.jsp** y lo vamos a colgar del WebApp por defecto (ROOT)



Vemos que se ejecuta

Internamente, un JSP gerena un Servlet. Podemos incluso ver el código que realmente se genera. (esto no lo necesitan saber los usuarios noveles aunque es muy importante a posteriori para depuraciones avanzadas)




```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;

public class otro_jsp extends HttpJspBase {


private static java.util.Vector _jspx_includes;

public java.util.List getIncludes() {
return _jspx_includes;
}

public void _jspService(HttpServletRequest request, HttpServletResponse response)
throws java.io.IOException, ServletException {

JspFactory _jspxFactory = null;
javax.servlet.jsp.PageContext pageContext = null;
HttpSession session = null;
ServletContext application = null;
ServletConfig config = null;
JspWriter out = null;
Object page = this;
JspWriter _jspx_out = null;
```

```
try {
_jspxFactory = JspFactory.getDefaultFactory();
response.setContentType("text/html;charset=ISO-8859-1");
pageContext = _jspxFactory.getPageContext(this, request, response,
null, true, 8192, true);
application = pageContext.getServletContext();
config = pageContext.getServletConfig();
session = pageContext.getSession();
out = pageContext.getOut();
_jspx_out = out;

out.write("<html>\r\n");
out.write("<h1>Esto es otro ejemplo");
out.write("</h1>\r\n");
out.write("</html>");
} catch (Throwable t) {
out = _jspx_out;
if (out != null && out.getBufferSize() != 0)
out.clearBuffer();
if (pageContext != null) pageContext.handlePageException(t);
} finally {
if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
}
}
}
```

Como podemos comprobar, un JSP es un pequeño artificio que sigue toda la lógica de los servlets.

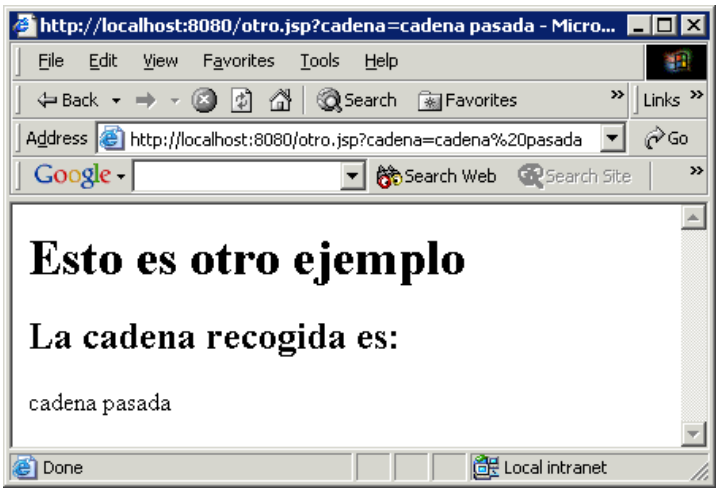
En una aplicación normal, lo primero que tenemos que hacer es recoger parámetros de la petición que nos han realizado.

```
<html>
<h1>Esto es otro ejemplo</h1>

<% String cadena = request.getParameter("cadena"); %>

<h2>La cadena recogida es: </h2> <%= cadena %>

</html>
```



Cuando se construyen aplicaciones con JSP´s hay que procurar desacoplar la lógica de presentación de la lógica de negocio. Para esto, se suele utilizar el denominado esquema **modelo-vista-controlador (veremos esto en otro tutorial donde hablaremos de Patrones de Diseño).**

En otras palabras, lo que realizamos es una consulta a un componente (normalmente un servlet) que realiza las labores complejas y delega sobre un JSP la labor de presentación (dejándole en variables de sesión o petición los valores a ser presentados).

Vamos a crear un Servlet que tenga este comportamiento

```
import javax.servlet.*;
import javax.servlet.http.*;

public class ejemploMvc extends HttpServlet {

public void doGet (HttpServletRequest request, HttpServletResponse response)
{

try
{
request.setAttribute ("edad", "24");
```

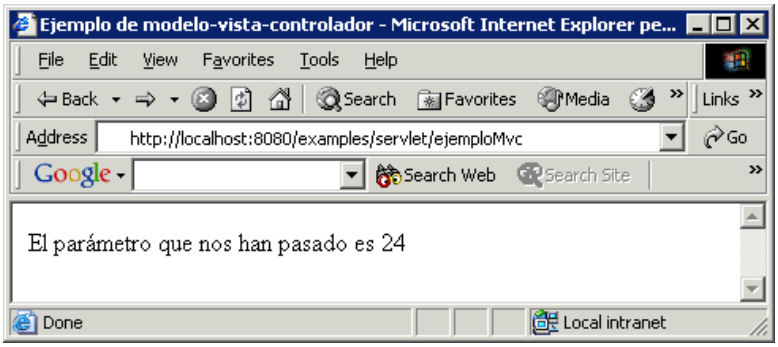
```
getServletConfig().getServletContext().getRequestDispatcher("/jsp/muestra.jsp").forward(request, response);
}
catch (Exception ex)
{
ex.printStackTrace ();
}
}
}
```

Nuestro JSP, para recoger estos valores deben tener la siguiente forma:

```
<html>
<head>
<title>Ejemplo de modelo-vista-controlador</title>
</head>

<body>
El parámetro que nos han pasado es
<% out.print (request.getAttribute("edad").toString()); %>
</body>
</html>
```

Invocamos



También podríamos haber escrito así el JSP

```
<html>
<head>
<title>Ejemplo de modelo-vista-controlador simple</title>
</head>

<body>
El parámetro que nos han pasado es:
<b><%= request.getAttribute("edad") %> </b>
</body>
</html>
```

A su vez, tambien podemos redirigir desde un jsp a otro JSP o servlet, utilizando

```
<jsp:forward page="/jsp/otro.jsp"/>
```

Sobre el Autor ..

de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
Mensajes multi-idioma en Java	Os mostramos como aprovechar las caracteristicas mutilenguaje de Java, usando las clases: Locate, ResourceBundle, MessageFormat, etc. Fundamental para un correcto diseño ...
Creación avanzada de Tags con cuerpo	En este tutorial os mostramos como crear TAGs para JSP que gestionen el cuerpo.
Uso de JNDI, includes y cookies en Servlets	En este tutorial veremos como usar variables de entorno desde JNDI, incluir un servlet en otro (include) y como usar cookies en Servlets
Seguridad en Tomcat	Os mostramos como proteger de un modo básico el acceso a recursos dentro de vuestro servidor de componentes Tomcat
Ejecutar JSPs almacenados en Base de Datos	Atendiendo una pregunta de nuestro foro, os mostramos como, con unos sencillos pasos, podemos ejecutar JSPs almacenandos en la base de datos. Esto puede ser una idea base para un gestor de contenidos construido en Java.
Activar soporte SSL en Tomcat	Os mostramos como activar el acceso SSL en Tomcat, utilizando certificados generados por Keygen (java)
Struts Jakarta	Cuando se ha trabajado creando aplicaciones Java poco a poco se va viendo la necesidad de normalizar los desarrollo. Uno de los Framework (entornos) más extendidos es Struts
Filtros de Servlets en Tomcat	En este tutorial os enseñamos la técnica (poco conocida) del encadenamiento de filtros en la activación de servlets, dentro del entorno Tomcat
Comunicación entre TAGs, Beans y JSPs	Os mostramos las posibilidades de comunicación entre JSPs, Bean y etiquetas de usuario.
Aplicaciones con JSPs	Os mostramos como construir una aplicación con JSP que acceda a MySQL

www.AdictosAlTrabajo.com Opimizado 800X600