

¿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)



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

Tutorial desarrollado por: [Miguel Angel Mena Sevilla](#)

Puedes encontrarme en Autentia



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

[Firma en nuestro libro de Visitas](#)

Master Java J2ee Oracle

Prácticas laborales 100% aseguradas Nuevo temario de Struts. Trabaja ya

XML to PDF in Java & J2EE

Output PDF, PCL5, HTML in Java J2EE Websphere, Weblogic, Tomcat, Jetty.

eBusiness Designer

Herramienta fácil y potente para el desarrollo de soluciones web

Java Reporting Tool

Powerful reporting solution for Java applications. Add charts too!

Anuncios Google

Anunciarse en este sitio

Subir ficheros al servidor con Struts

En [Autentia](#) trabajamos a diario en el desarrollo de aplicaciones Web a partir de las tecnologías de desarrollo más avanzadas; la más utilizada de éstas es la API de desarrollo de Aplicaciones Web **Struts**, de Jakarta.

En este tutorial vamos a crear un sencillo programa para subir archivos a un servidor. Esta funcionalidad la implementa `FormFile` (de la librería `org.apache.struts.upload`). Esta interfaz es la principal herramienta de subida de ficheros de Struts; representa un fichero que se desea subir (a un servidor).

Para facilitar la presentación del diseño, optamos por presentarlo siguiendo el modelo de arquitectura utilizado por Struts: MVC (Modelo-Vista-Controlador).

1. Modelo: creando un Form Bean

Creamos un Bean **UploadForm**; este fichero contiene una variable privada **theFile** de tipo **FormFile**, que utilizamos para definir el modelo de datos de la subida de ficheros. Este es el código del Bean:

```
package autentia;

import org.apache.struts.action.*;
import org.apache.struts.action.upload.FormFile;

/**
 * Autor: Miguel Mena
 */
/**
 * UploadForm.java
 */
public class UploadForm extends ActionForm {

    /**
     * The file that the user has uploaded
     */
    private FormFile theFile;

    /**
     * The file path to write to
     */
    protected String filePath;

    /**
     * Retrieve a representation of the file the user has uploaded
     */
    public FormFile getTheFile() {
        return theFile;
    }

    /**
     * Set a representation of the file the user has uploaded
     */
    public void setTheFile(FormFile theFile) {
        this.theFile = theFile;
    }
}
```

El controlador accederá a esta variable utilizando los métodos **getTheFile** y **setTheFile**.

2. Controlador (I): creando Action Class

A continuación definimos un Action Class que implemente la funcionalidad del sistema: **UploadAction**. Este Bean lee el fichero a subir, imprime sus parámetros y almacena el contenido del fichero en un array de salida (configurable); nótese que el máximo tamaño de fichero que permitimos es 4 MB, el máximo que hemos elegido para Struts. Más adelante podremos configurar este parámetro en el fichero **struts-config.xml**.

```
package autentia;

import javax.io.*;

import javax.servlet.http.*;

import org.apache.struts.action.*;
import org.apache.struts.upload.*;

/*
 * Autor: Miguel Mena
 */

/*
 * UploadAction.java
 */
public class UploadAction extends Action
{
    public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {

        if (form instanceof UploadForm) {
            UploadForm theForm = (UploadForm) form;

            // mostramos los parametros del fichero
            FormFile theFile = theForm.getTheFile();
            String contentType = theFile.getContentType();
            String fileName = theFile.getFileName();
            int fileSize = theFile.getFileSize();
            byte[] fileData = theFile.getFileData();
            System.out.println("Tipo: " + contentType);
            System.out.println("Nombre: " + fileName);
            System.out.println("Tamano: " + fileSize);

            try {
                //guarda los datos del fichero
                ByteArrayOutputStream baos = new ByteArrayOutputStream();
                InputStream stream = file.getInputStream();

                // solo si el archivo es de menos de 4MB
                if (fileSize() < (4*1024000)) {
                    byte[] buffer = new byte[8192];
                    int bytesLeidos = 0;
                    while ((bytesLeidos = stream.read(buffer, 0, 8192)) != -1) {
                        baos.write(buffer, 0, bytesLeidos );
                    }
                    data = new String(baos.toByteArray());
                }
                else {
                    data = new String("Fichero de más de 4MB: no pudo almacenarse." +
                                     " Tamano del fichero: " + fileSize() + " bytes.");
                }
            }

            return mapping.findForward("success");
        }
        // solo si no lee un UploadForm
        return null;
    }
}
```

3. Controlador (II): declaraciones en struts-config.xml

Ya tenemos el modelo y el controlador definidos. Ahora tenemos que declararlos en el fichero **struts-config.xml** a utilizar en esta aplicación. Además, también definimos el tamaño máximo a permitir:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
    "http://struts.apache.org/dtds/struts-config_1_2.dtd">

<struts-config>

    <form-beans>
```

4. Vista: creando upload.jsp y display.jsp

upload.jsp

display.jsp:

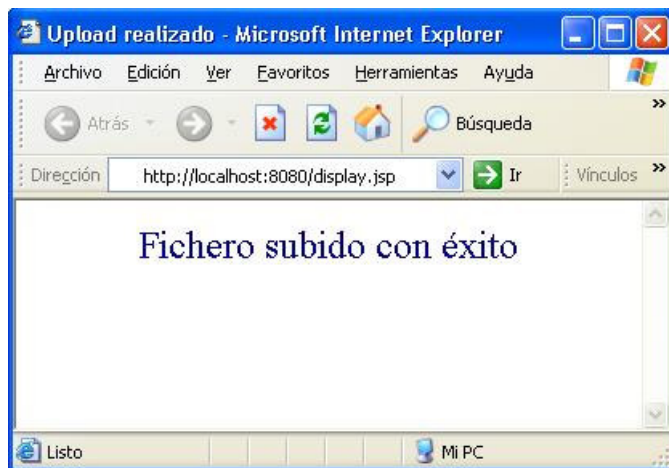
09/03/2006

5. Ejecución

Una vez desplegada, vuestra aplicación Web debería mostrar la siguiente ventana:



Una vez introducido el fichero a subir, nos devuelve la correspondiente página de éxito:



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	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto

[Consola de administración de Struts](#)

[Struts Jakarta](#)

Descripción

En este tutorial aprenderemos a simplificar la gestión de Struts a través de una consola gráfica gratuita

Cuando se ha trabajado creando aplicaciones Java poco a poco se va viendo la necesidad de normalizar los desarrollos. Uno de los Framework (entornos) más extendidos es Struts

[Plantear una aplicación Web y Struts](#)

Os mostramos un posible modo de plantear una aplicación Web (análisis) y darla forma. El Framework utilizado es struts y tratamos de identificar qué depende de este Framework y qué no.

[Aplicación profesional con Struts](#)

En este tutorial, os mostramos como crear una aplicación profesional, usando numerosos patrones y utilizando Struts.

[Introducción a Struts Flow](#)

Struts Flow es un módulo de extensión del conocido framework Struts, que facilita la implementación del flujo de páginas de una aplicación web

[Desarrollo Struts con XDoclet](#)

Alejandro Perez nos enseña como simplificar el desarrollo de aplicaciones J2EE basadas en Struts, automatizando la generación de código con XDoclet

[Pool de conexiones a BBDD con struts](#)

Os mostramos como configurar un pool de conexiones a base de datos en vuestras aplicaciones construidas con struts

[JSF y comparativa con Struts](#)

Os mostramos los pasos necesarios para empezar a utilizar JSF (Java Server Faces) y su comparación / relación con Struts

[Activar el soporte SSL en Struts](#)

Os mostramos las particularidades de uso y configuración de Struts para trabajar con SSL

[Integración de Struts y eclipse](#)

Alejandro Perez nos enseña como construir un entorno de alta eficiencia de desarrollo on Struts a través de plugins de eclipse

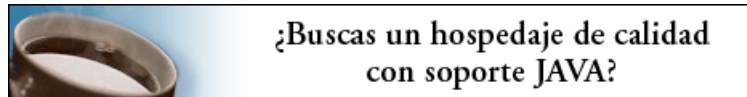
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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