

¿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](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: [Alberto Carrasco Montenegro](#)

Puedes encontrarme en [Autentia](#)



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

[Curso Web J2EE](#)

Curso Avanzado en Desarrollo Web con J2EE

[Action XML- Apache Struts](#)

Edit/Validate Action XML for Struts Syntax Help, Easy-to-Use, Free D/L.

[Master Java J2ee Oracle](#)

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

[Master Java Élite Madrid](#)

Master Java-J2ee-ejbs profesional. Prácticas en empresas. Becas. Exes.

Anuncios Goooooogle

Anunciarse en este sitio

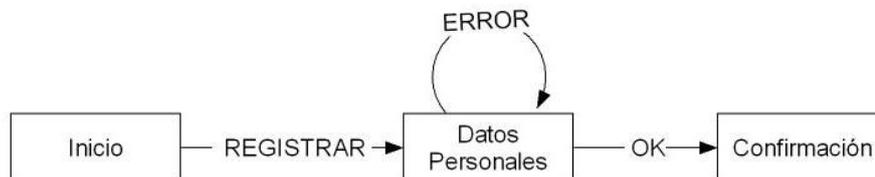
En [Autentia](#) trabajamos constantemente con los frameworks mas avanzados para el diseño de servicios web en nuestros desarrollos. En esta línea, dedicamos especial interés a la investigación y utilización de las extensiones de estos frameworks que faciliten y aporten mayor potencia al diseño de una aplicación web. Hoy os queremos mostrar una introducción al módulo Struts Flow del framework Struts de cara al desarrollo de aplicaciones web.

1. ¿Qué es Struts Flow?

Struts Flow es un módulo de extensión del framework *Struts* dirigido a definir y gestionar los flujos de páginas dentro de una aplicación web.

El flujo de páginas de una aplicación web consiste en la secuencia de páginas por las que pasa dicha aplicación en función de la conversación que mantenga con el usuario. Dependiendo de las opciones que escoja el usuario, resultados de las operaciones de proceso...etc, la aplicación seguirá una ruta de páginas específica u otra.

Por ejemplo, el flujo de páginas de una aplicación web sencilla podría ser el siguiente:



La aplicación se dedica a registrar los datos de los usuarios que la acceden. Desde la pagina inicial el usuario selecciona la opción REGISTRAR para proceder a su registro. Se le muestra una página donde debe introducir sus datos personales para completar el registro. Una vez introducidos, la aplicación muestra una página confirmando el registro si no se produjo ningún error, o bien ofrece al usuario la posibilidad de volver a introducir sus datos correctamente si se produjo algún error.

Como puede observarse, la forma más intuitiva de representar el flujo de páginas de una aplicación es con un diagrama de estados. Si bien otros módulos de gestión de flujos similares, como *Spring Web Flow*, utilizan estructuras declarativas basadas en máquinas de estados, *Struts Flow* realiza una gestión de los flujos de la aplicación basada en código *javascript*, es decir, en estructuras de control de flujo de programación clásica (if, else, etc).

Partiendo de este concepto, los aspectos fundamentales que potencia *Struts Flow* son:

- Proporcionar un motor capaz de capturar los flujos de páginas de una aplicación, integrándolo con algunos frameworks de uso habitual para desarrollo de aplicaciones web.
- Encapsulación de la lógica de los flujos de páginas como un módulo autónomo que puede reutilizarse en diferentes situaciones. Hay aplicaciones web específicas que suelen utilizar la misma lógica de flujo de páginas, como por ejemplo, aplicaciones tipo "carrito de la compra", correo web, etc.

2. Instalación y requisitos

El ejemplo sencillo que se estudiará más adelante en este documento, se desarrolló y ejecutó en un entorno *Windows XP* que disponía de la distribución *Java j2sdk-1.4.2*. Esta distribución puede obtenerse gratuitamente desde la web de *SUN* en el enlace:

<http://java.sun.com/products/archive/j2se/1.4.2/index.html>

Es necesario obtener distribuciones del framework *Struts*, así como de su módulo *Struts Flow*. Para realizar el ejemplo que se comentará más tarde,

se utilizó la distribución *struts-flow-0.2.zip*, que pueden obtenerse gratuitamente desde el enlace:

http://sourceforge.net/project/showfiles.php?group_id=49385&package_id=120079

Esta distribución lleva incluido los archivos *.jar* para la distribución 1.1 de *Struts*. En cuanto a la versión de *Struts* a utilizar, una observación importante es que debe utilizarse la 1.1 o superiores para el correcto funcionamiento con el módulo *Struts Flow*.

Una vez descomprimido el archivo, será necesario que los siguientes archivos sean visibles en el classpath:

- struts-flow-0.2\lib\struts\commons-beanutils.jar
- struts-flow-0.2\lib\struts\commons-collections.jar
- struts-flow-0.2\lib\struts\commons-digester.jar
- struts-flow-0.2\lib\struts\commons-lang.jar
- struts-flow-0.2\lib\struts\commons-logging.jar
- struts-flow-0.2\lib\struts\commons-validator.jar
- struts-flow-0.2\lib\struts\jakarta-oro-2.0.6.jar
- struts-flow-0.2\lib\struts\struts-1.1.jar
- struts-flow-0.2\dist\struts-flow-0.2.jar

Se utilizó la herramienta *Ant* para compilar y empaquetar el ejemplo realizado. La distribución utilizada es *apache-ant-1.6.5-bin.zip*, que puede obtenerse gratuitamente desde el siguiente enlace:

<http://ant.apache.org/bindownload.cgi>

Por último, para desplegar el ejemplo se utilizó el contenedor de aplicaciones web de Tomcat. Se utilizó la distribución *jakarta-tomcat-5.0.30.zip*, que puede obtenerse gratuitamente desde el enlace:

<http://tomcat.apache.org/download-55.cgi>

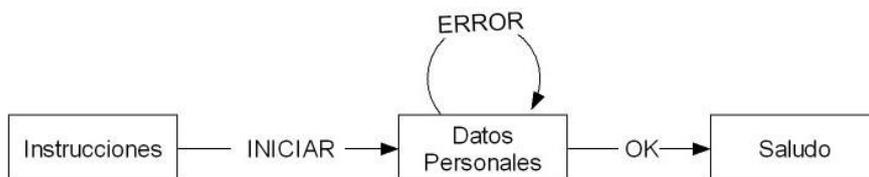
3. Un ejemplo sencillo

3.1. Idea

Para ilustrar el funcionamiento y la forma de trabajar con *Struts Flow* se realizará una aplicación web cuya funcionalidad será la siguiente:

- Obtener nombre y apellidos del usuario a través de un formulario.
- Comprobar que se introdujeron dichos datos en el formulario. Si no se introdujo alguno de ellos, informará al usuario de la situación de error y le ofrecerá la posibilidad de volver a introducirlos.
- Si la captura de los datos anteriores se realiza correctamente, la aplicación muestra una página que envía un saludo al usuario.

El flujo de páginas de la aplicación puede resumirse en el siguiente diagrama de estados:



3.2. Implementación del flujo de páginas

Como se comentó anteriormente, la gestión del flujo de la aplicación a través de *Struts Flow* se realiza a través de un programa *javascript*.

En primer lugar, se declaran las variables que utilizará la aplicación para obtener los datos del usuario y para mostrar mensajes de error por pantalla.

```
var nombre;
var apellidos;
var mensaje="";
```

A continuación el programa entra en un bucle infinito en el que gestionará la captura de los datos del usuario.

Dentro de este bucle, en primer lugar envía un formulario para que el usuario introduzca su nombre y apellidos. Esto lo realiza a través de la función *forwardAndWait*. Esta función envía dicho formulario e indica las variables que almacenarán los datos del usuario y el posible mensaje de error que podría mostrar la aplicación. El primer parámetro ("error") identifica el estado al que corresponde la página que mostrará la aplicación. Se comentará más adelante como se enlaza un estado con su recurso de presentación correspondiente.

```
forwardAndWait("error",
{ "nombre" : nombre,
  "apellidos" : apellidos,
```

```
"mensaje" : mensaje });
```

Tras el envío de este formulario, esta función espera a que el usuario pulse *Entrar* para que se procese el formulario, y devolver el control de nuevo al programa principal. El programa entonces verifica los datos del usuario para decidir la siguiente acción en el flujo de la aplicación: si el usuario no introdujo alguno de los datos (nombre o apellidos) le advertirá de la situación de error y le dará la oportunidad de volver a introducirlos. Esto se traduce en la inserción de un mensaje de error en la variable correspondiente y una nueva iteración en el bucle principal. Si la inserción de los datos fue correcta, se sale del bucle principal, es decir, no se vuelve a enviar al usuario el formulario indicado anteriormente y se pasa al siguiente estado del flujo de la aplicación.

```
nombre = getRequestParams().nombre;
apellidos = getRequestParams().apellidos;

if ((nombre=="") || (apellidos==""))
    mensaje = "Por favor, introduzca su nombre y apellidos";
else
    break;
```

Fuera del bucle principal se encuentra el código necesario para que la aplicación pase al estado en que se envía un saludo al usuario.

```
forwardAndWait("ok",
{ "nombre" : nombre,
  "apellidos" : apellidos } );
```

Puede observarse que en este caso sólo hace falta enviar la información correspondiente a los datos del usuario (no hay que mostrar mensaje de error). El estado "ok" tendrá asociado un recurso de presentación que mostrará un saludo al usuario, cuyos nombre y apellidos se especifican en las variables pasadas como parámetros a la función *forwardAndWait*.

A modo de resumen, se muestra a continuación el código completo de este programa *javascript*.

```
function main() {

var nombre;
var apellidos;
var mensaje="";

while (true)
{
forwardAndWait("error",
{ "nombre" : nombre,
  "apellidos" : apellidos,
  "mensaje" : mensaje } );

nombre = getRequestParams().nombre;
apellidos = getRequestParams().apellidos;

if ((nombre=="") || (apellidos==""))
    mensaje = "Por favor, introduzca su nombre y apellidos";
else
    break;
}

forwardAndWait("ok",
{ "nombre" : nombre,
  "apellidos" : apellidos } );

}
```

3.3. Integración del flujo de páginas

Una vez implementado el flujo de páginas, se mostrará a continuación como integrarlo con la aplicación web diseñada como ejemplo.

En el fichero *struts-config.xml* deben definirse los siguientes *action mapping* y *plug-in*.

```
<struts-config>

<action-mappings>

<action path="/hello"
type="autentia.struts.flow.FlowAction"
className="autentia.struts.flow.FlowMapping">

<set-property property="function" value="main" />

<forward name="error" path="/hello.jsp" />
<forward name="ok" path="/ok.jsp" />
</action>
</action-mappings>
```

```

<plug-in className="autentia.struts.flow.FlowPlugIn">
  <set-property property="scripts" value="/WEB-INF/hello.js" />
  <set-property property="debugger" value="false" />
  <set-property property="timeToLive" value="600000" />
</plug-in>

</struts-config>

```

En el *action mapping* se indica el programa *javascript* que se utilizará para la gestión del flujo de la aplicación, declarado en el elemento *plug-in* situado a continuación. En el *action mapping* se indica también el recurso de presentación que se asociará a cada estado del flujo, a través de los elementos *forward*. Como puede observarse, en este caso serán sendos archivos *.jsp*.

3.4. Ejecución

Una vez compilado, empaquetado y desplegado el proyecto en un contenedor de aplicaciones web (Tomcat, en este caso), la ejecución del ejemplo produce los siguientes resultados. Se mostrará la correspondencia entre el flujo de páginas de la aplicación y el diagrama de estados del mismo que se diseñó al principio.

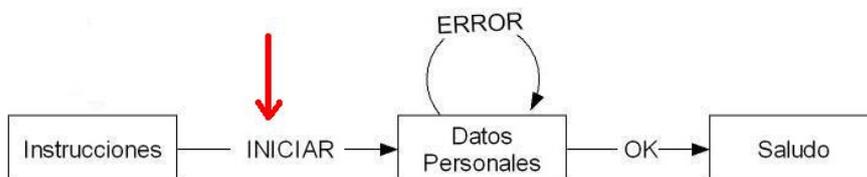
1. Instrucciones de la aplicación

STRUTS FLOW

La aplicación mostrará dos campos de texto para que el usuario introduzca su nombre y apellidos.

Tras pulsar el botón "Entrar" del formulario, la aplicación saludará al usuario.

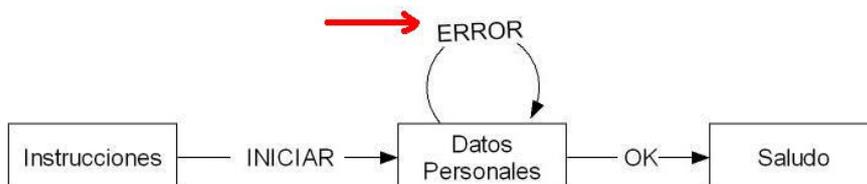
[Iniciar ejemplo](#)



2. Relleno del formulario (error)

Nombre:

Apellidos:



Se vuelve a la misma página y la lógica de validación produce un mensaje de error.

Nombre:

Apellidos:

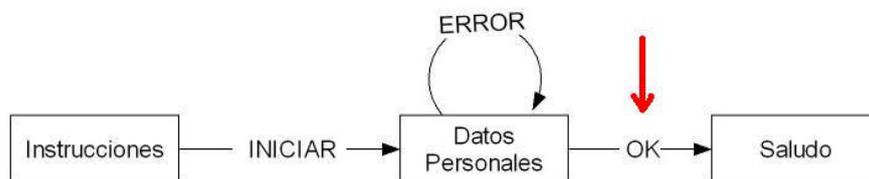
Por favor, introduzca su nombre y apellidos

3. Relleno del formulario (ok)

Nombre:

Apellidos:

←



La aplicación muestra entonces una página saludando al usuario.

Bienvenido, Alberto Fontanilla !

4. Fuentes

Página de Souceforge sobre Struts Flow:
<http://struts.sourceforge.net/struts-flow/>

Apache Ant:
<http://ant.apache.org/>

Apache Tomcat:
<http://tomcat.apache.org/>

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

soluciones reales para su negocio

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	Descripción
Consola de administración de Struts	En este tutorial aprenderemos a simplificar la gestión de Struts a través de una consola gráfica gratuita
Struts y EL en Netbeans 4	Os mostramos como configurar la última contribución de Struts que se integra con el lenguaje de expresiones EL.
Uso de Tiles en Struts	Os mostramos como utilizar el sistema de plantillas proporcionado por Struts (tiles)
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
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.
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
Introducción a ANT	En el mundo Java, la compilación, verificación e instalación de aplicaciones se ha normalizado con este potente paquete llamado ANT.
Evitar doble-click en JSPs y Struts	Os mostramos como construir unas librerías de TAGs para evitar el problema de doble-click en aplicaciones JSP y como se soluciona (qué teneis que hacer) con el framework 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
Aplicación profesional con Struts	En este tutorial, os mostramos como crear una aplicación profesional, usando numerosos patrones y utilizando Struts.

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](#)

