

# ¿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)

Estas en: Inicio Tutoriales Introducción a Ajax4Jsf Más

### Ultimas Noticias

- » Historia de la informática. Capítulo 60. 1983 (1ª parte)
- » Contratos ágiles: Vendiendo Scrum a tus clientes.
- » Quinta charla Autentia + Projectalis + Agile Spain: Contratos ágiles: Vendiendo Scrum a tus clientes
- » Historia de la informática. Capítulo 59. 1982
- » Resumen de la cuarta charla gratuita de Autentia: SCRUM (con video)

### +Noticias Destacadas

- » Contratos ágiles: Vendiendo Scrum a tus clientes.
- » Quinta charla Autentia + Projectalis + Agile Spain: Contratos ágiles: Vendiendo Scrum a tus clientes
- » Lo mejor de esta semana: Curso de Scrum con Ángel Medinilla
- » Resumen de la cuarta charla gratuita de Autentia: SCRUM (con video)

### +Comentarios Cómic

### +Enlaces

### Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

### Tutorial desarrollado por

**Juan Ramos** **Alonso**



Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión (cursando Ingeniería 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](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [Ajax4Jsf.pdf](#)

Fecha de creación del tutorial: 2007-04-09

## Introducción a Ajax4jsf

### Índice de contenidos

- 1. Introducción
- 2. Etiquetas
- 3. Entorno utilizado
- 4. Instalación
- 5. Pruebas
- 6. Conclusión

### 1. Introducción

Ajax4jsf es una librería open source que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework podemos variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. En definitiva Ajax4jsf permite dotar a nuestra aplicación JSF de contenido mucho más profesional con muy poco esfuerzo.

### 2. Etiquetas



Web  
 [www.adictosaltrabajo](#)

### Últimos tutoriales

2009-06-15  
[Configuración de la desconexión de usuarios con ICEFaces](#)

2009-06-10  
[LWUIT: Una librería gráfica tipo AWT o Swing para J2ME](#)

2009-06-10  
[Mapas mentales con XMind](#)

2009-02-26  
[Redimensionar Imagenes en Windows Vista](#)

2009-06-08  
[UploadFile con Icefaces + Hibernate + Anotaciones](#)

2009-06-05  
[Habilitar exportación en Liferay](#)

2009-06-01  
[Registrar Liferay en Eclipse](#)

El funcionamiento del framework es sencillo. Mediante sus propias etiquetas se generan eventos que envían peticiones al contenedor Ajax. Estos eventos se pueden producir por pulsar un botón, un enlace, una región específica de la pantalla, un cambio de estado de un componente, cada cierto tiempo, etc. No nos tendremos que preocupar de crear el código Javascript y el objeto **XMLHttpRequest** para que envíe la petición al servidor ya que esto lo hará por nosotros el framework. Lo más importante es conocer las distintas etiquetas que contiene, son las siguientes:

**<aj4:support>** : Etiqueta que se puede añadir a cualquier otra etiqueta JSF para dotarla de funcionalidad Ajax. Permite al componente generar peticiones asíncronas mediante eventos (onclick, onblur, onchange,...) y actualizar campos de un formulario de forma independiente, sin recargar toda la página.

**<aj4:poll>** : Realiza cada cierto tiempo una petición al servidor.

**<aj4:commandButton>** : Botón de envío de formulario similar a `<h:commandButton>` de JSF. La principal diferencia es que se puede indicar que únicamente actualice ciertos componentes evitando la recarga de todo el formulario.

**<aj4:commandLink>**: Comportamiento similar a `<aj4:commandButton>` pero en un link.

**<aj4:htmlCommandLink>** : Muy parecida a la anterior etiqueta con pequeñas diferencias en la generación de links y cuando se utilizan etiquetas `<f:param>`.

**<aj4:region>** : Determina un área a decodificar en el servidor después de la petición Ajax.

**<aj4:status>** : Muestra el estado de la petición Ajax. Hay 2 estados posibles: procesando petición y petición terminada. Por ejemplo mientras dure el proceso de la llamada al servidor y la evaluación de la petición se puede mostrar el texto " procesando..." y cuando termine la petición y se devuelva la respuesta a la página se cambia el texto por "petición finalizada".

**<aj4:form>** : Similar al `<h:form>` con la diferencia de que se puede enviar previamente el contenido al contenedor Ajax.

**<aj4:actionparam>** : Etiqueta que combina la funcionalidad de la etiqueta `<f:param>` y `<f:actionListener>`.

**<aj4:outputPanel>** : Se utiliza para agrupar componentes para aplicarles similares propiedades, por ejemplo a la hora de actualizar sus valores tras la petición Ajax.

**<aj4:ajaxListener>** : Similar a la propiedad `actionListener` o `valueChangeListener` pero con la diferencia de que la petición se hace al contenedor Ajax.

**<aj4:jsFunction>** : Se utiliza para pasarle un valor automáticamente a una función Javascript tras recibirlo del servidor.

**<aj4:loadScript>** : Inserta en la página las funciones Javascript contenidas en un archivo .js

**<aj4:loadStyle>** : Igual que la anterior etiqueta pero para una hoja de estilos .css

**<aj4:loadBundle>** : Similar al `<f:loadBundle>` de JSF.

**<aj4:log>** : Carga en la página una consola que muestra las trazas de los logs que devuelve el contenedor Ajax.

**<aj4:include>** : Se utiliza para incluir en la página el contenido de otra de acuerdo a la definición que se haga en las reglas de navegación del faces-config. Es decir la siguiente página a cargar de acuerdo a la navegación especificada se cargaría en la vista actual.

**<aj4:repeat>** : Etiqueta para iterar sobre una colección y mostrar todos sus campos.

**<aj4:keepAlive>** : Permite mantener un bean en un estado determinado durante peticiones.

**<aj4:mediaOutput>** : Componente que permite mostrar contenido multimedia como imágenes, vídeos, archivos sonoros, etc.

**Para conocer mucho más sobre el framework puedes visitar la documentación oficial [aquí](#).**

### 3. Entorno utilizado

- Para las pruebas he utilizado las siguientes herramientas:
- JDK 5.0
- Apache MyFaces 1.1.5
- Ajax4jsf 1.1.0
- Eclipse 3.2.1
- Navegador Mozilla Firefox 2.0

Puedes también utilizar el plugin de eclipse Exadel Studio que incluye soporte para Ajax4jsf (<http://www.exadel.com/web/portal/download/es>).

### 4. Instalación

Partiendo de que se tiene instalada la JDK de Sun y Apache MyFaces nos descargamos el paquete Ajax4jsf de la web oficial (<http://labs.jboss.com/portal/jbossAjax4jsf/downloads>). Abrimos el archivo comprimido y copiamos los fuentes Ajax4jsf.jar y oscache-2.3.jar en el directorio WEB-INF/lib de nuestro proyecto. La librería oscache-2.3.jar se utiliza para añadir al proyecto funcionalidad de cache.

Añadimos al **web.xml** el siguiente filtro para que todas las peticiones al Faces Servlet pasen previamente por el filtro de Ajax4jsf.

2009-05-29  
[Liferay Social Office](#)

2009-05-28  
[Broadcast con Ustream](#)

2009-05-25  
[Tabla datos accesible con ordenación y paginación](#)

2009-05-21  
[Primeros pasos con Audacity: Un editor de sonido libre y multiplataforma.](#)

2009-05-11  
[Introducción a TortoiseSVN](#)

2009-05-07  
[Hacer 'scp' de varios ficheros sin solicitud de clave](#)

2009-05-02  
[Plugin Hibernate3 para Maven](#)

2009-04-26  
[AgileDraw: una técnica rápida de modelado](#)

2009-04-24  
[Spring AOP: Cacheando aplicaciones usando anotaciones y aspectos con Aspectj](#)

2009-04-20  
[Modelos de conocimiento con CmapTools](#)

2009-04-16  
[Informes Crosstab con iReport](#)

2009-04-16  
[Registro de un fichero de datos personales con el formulario NOTA](#)

2009-04-15  
[Estadísticas de www.adictosaltrabajo.com Abril 2009](#)

2009-04-15  
[Iniciación a OSWorkflow con Spring](#)

```

view plain print ?
01. <filter>
02.   <display-name>Ajax4jsf Filter</display-name>
03.   <filter-name>Ajax4jsf</filter-name>
04.   <filter-class>org.Ajax4jsf.Filter</filter-class>
05. </filter>
06.
07. <filter-mapping>
08.   <filter-name>Ajax4jsf</filter-name>
09.   <servlet-name>Faces Servlet</servlet-name>
10.   <dispatcher>REQUEST</dispatcher>
11.   <dispatcher>FORWARD</dispatcher>
12.   <dispatcher>INCLUDE</dispatcher>
13. </filter-mapping>

```

En cada página JSP tenemos que incluir la directiva taglib siguiente:

```

view plain print ?
01. <%@ taglib uri="https://Ajax4jsf.dev.java.net/Ajax" prefix="a4j"%>

```

Si utilizamos páginas XHTML debemos añadir esta otra directiva:

```

view plain print ?
01. <xmlns:a4jxmlns:a4j="https://Ajax4jsf.dev.java.net/Ajax" >

```

## 5. Pruebas

Vamos a hacer una prueba con algunas etiquetas para ver cómo funcionan. Creamos un formulario donde pedimos el nombre, apellidos y aficiones. En los campos de texto se captura el evento onkeyup con la etiqueta **<a4j:support event="onkeyup" ... />** que llama al bean homeBean recuperando los valores del formulario. En el caso del nombre y apellidos se convierten los caracteres a mayúsculas.

También mediante la etiqueta **<a4j:status for="inputData">** se muestra una imagen mientras se está procesando la petición y otra imagen cuando la petición está finalizada. Para comprobar las llamadas al contenedor Ajax se pueden ver las trazas de log mediante la etiqueta

```

view plain print ?
01. <a4j:log level="ALL" popup="false" width="1000" height="100" />

```

Código de la página JSP

2009-04-14  
[Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.](#)

2009-04-13  
[JTAPI. El API de Telefonía para Java](#)

2009-04-13  
[Registro de Web Services con Apache jUDDI. Configuración y ejemplo](#)

2009-04-13  
[Cómo hacer UML con Eclipse y el plugin UML2](#)

2009-04-09  
[Spring WS: Servicios Web a través del correo electrónico](#)

2009-04-02  
[Creación de cursos con Moodle](#)

2009-03-31  
[Integrar Liferay Portal 5.2.1 con Pentaho BI 2.0.0 sobre MySQL 5.1](#)

2009-03-31  
[Spring WS: Construcción de Clientes de Servicios Web con Spring](#)

2009-03-30  
[Administración de sitios Moodle](#)

2009-03-29  
[Empaquetamiento de aplicaciones de escritorio \(standalone\) con Maven](#)

2009-03-27  
[Primeros pasos con Moodle](#)

2009-03-26  
[Introducción a JSF Java](#)

2009-03-25  
[A1 Website Analyzer](#)

2009-03-24  
[Cómo ver el correo de Gmail sin conexión a Internet](#)

[view plain](#) [print](#) ?

```
01. <%@page contentType="text/html" pageEncoding="UTF-8"%>
02. <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
03. <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
04. <%@ taglib uri="https://ajax4jsf.dev.java.net/ajax" prefix="a4j"%>
05.
06. <f:view>
07.     <h:form>
08.         <h:panelGrid columns="2">
09.             <a4j:region id="inputData">
10.                 <h:panelGrid columns="2">
11.                     <h:outputText value="Nombre:" />
12.                     <h:inputText id="name" value="#{homeBean.name}">
13.                         <a4j:support event="onkeyup" reRender="outputName" />
14.                     </h:inputText>
15.
16.                     <h:outputText value="Apellidos:" />
17.                     <h:inputText id="lastName" value="#{homeBean.lastName}">
18.                         <a4j:support event="onkeyup" reRender="outputLastname" />
19.                     </h:inputText>
20.
21.                     <h:outputText value="Aficiones:" />
22.                     <h:selectManyListbox value="#{homeBean.hobbies}">
23.                         <f:selectItems value="#{homeBean.hobbiesItems}" />
24.                         <a4j:support event="onchange" reRender="outputHobbies" />
25.                     </h:selectManyListbox>
26.                 </h:panelGrid>
27.
28.                 <h:panelGrid columns="2">
29.                     <h:outputText value="Estado de la petición: " />
30.                     <a4j:status for="inputData">
31.                         <f:facet name="start">
32.                             <h:graphicImage value="/img/procesando.jpg" />
33.                         </f:facet>
34.
35.                         <f:facet name="stop">
36.                             <h:graphicImage value="/img/ok.jpg" />
37.                         </f:facet>
38.                     </a4j:status>
39.                 </h:panelGrid>
40.             </a4j:region>
41.
42.             <h:panelGrid columns="2">
43.                 <h:outputText value="Has introducido" /><h:outputText value=":" />
44.                 <h:outputText value="Nombre:" /> <h:outputText id="outputName" value="
45.                 <h:outputText value="Apellidos:" /> <h:outputText id="outputLastname"
46.                 <h:outputText value="Aficiones:" /> <h:outputText id="outputHobbies" \
47.             </h:panelGrid>
48.         </h:panelGrid>
49.
50.         <a4j:log level="ALL" popup="false" width="1000" height="100" />
51.     </h:form>
52. </f:view>
```

Código del Bean

2009-03-20  
[JasperReports Maven Plugin](#)

2009-03-16  
[Creación de contenidos SCORM: eXe](#)

2009-03-15  
[Spring WS: Creación de Servicios Web con Spring](#)

2009-03-13  
[Instalación Alfresco \(Labs\)](#)

2009-02-26  
[Maven JXR Plugin: publica el código fuente en el site](#)

2009-03-15  
[Generación de XML Schema \(XSD\) y DTD a partir de documentos XML](#)

2009-03-04  
[Persistencia con Spring](#)

2009-02-26  
[Vistas materializadas](#)

2009-02-03  
[Instalación de MySQL 5.1 en Windows](#)

2009-03-03  
[Instalación de Java Virtual Machine](#)

2009-03-03  
[Primeros Pasos con Liferay 5.2.1](#)

2009-02-27  
[Edición de video MPEG2](#)

2009-02-26  
[Introducción teórica a XPath](#)

2009-02-26  
[Integración Selenium / Maven 2 / Surefire / Cargo / Tomcat 6](#)

2009-02-24  
[Selenium Remote Control](#)

[view plain](#) [print](#) [?](#)

```
01. package com.tutoriales.ajax4jsf.bean;
02. import java.util.ArrayList;
03. import java.util.List;
04. import javax.faces.model.SelectItem;
05. import org.apache.myfaces.component.html.ext.HtmlSelectManyListbox;
06.
07. public class HomeBean {
08.
09.     private String name;
10.     private String lastName;
11.     private List<SelectItem> hobbiesItems = new ArrayList<SelectItem>();
12.     private List<HtmlSelectManyListbox> hobbies = new ArrayList<HtmlSelectManyListbox>
13.
14.     public HomeBean() {
15.         hobbiesItems.add(new SelectItem("Cine"));
16.         hobbiesItems.add(new SelectItem("Teatro"));
17.         hobbiesItems.add(new SelectItem("Television"));
18.         hobbiesItems.add(new SelectItem("Lectura"));
19.         hobbiesItems.add(new SelectItem("Deporte"));
20.         hobbiesItems.add(new SelectItem("Fotografia"));
21.         hobbiesItems.add(new SelectItem("Viajes"));
22.     }
23.
24.     public List<SelectItem> getHobbiesItems() {
25.         return hobbiesItems;
26.     }
27.
28.     public void setHobbiesItems(List<SelectItem> hobbiesItems) {
29.         this.hobbiesItems = hobbiesItems;
30.     }
31.
32.     public String getLastName() {
33.         return lastName;
34.     }
35.
36.     public void setLastName(String lastName) {
37.         this.lastName = lastName.toUpperCase();
38.     }
39.
40.     public String getName() {
41.         return name;
42.     }
43.
44.     public void setName(String name) {
45.         this.name = name.toUpperCase();
46.     }
47.
48.     public List<HtmlSelectManyListbox> getHobbies() {
49.         return hobbies;
50.     }
51.
52.     public void setHobbies(List<HtmlSelectManyListbox> hobbies) {
53.         this.hobbies = hobbies;
54.     }
}
```

## Últimas ofertas de empleo

2009-04-30  
[T. Información - Analista / Programador - MADRID.](#)

2009-04-24  
[Comercial - Ventas - VALENCIA.](#)

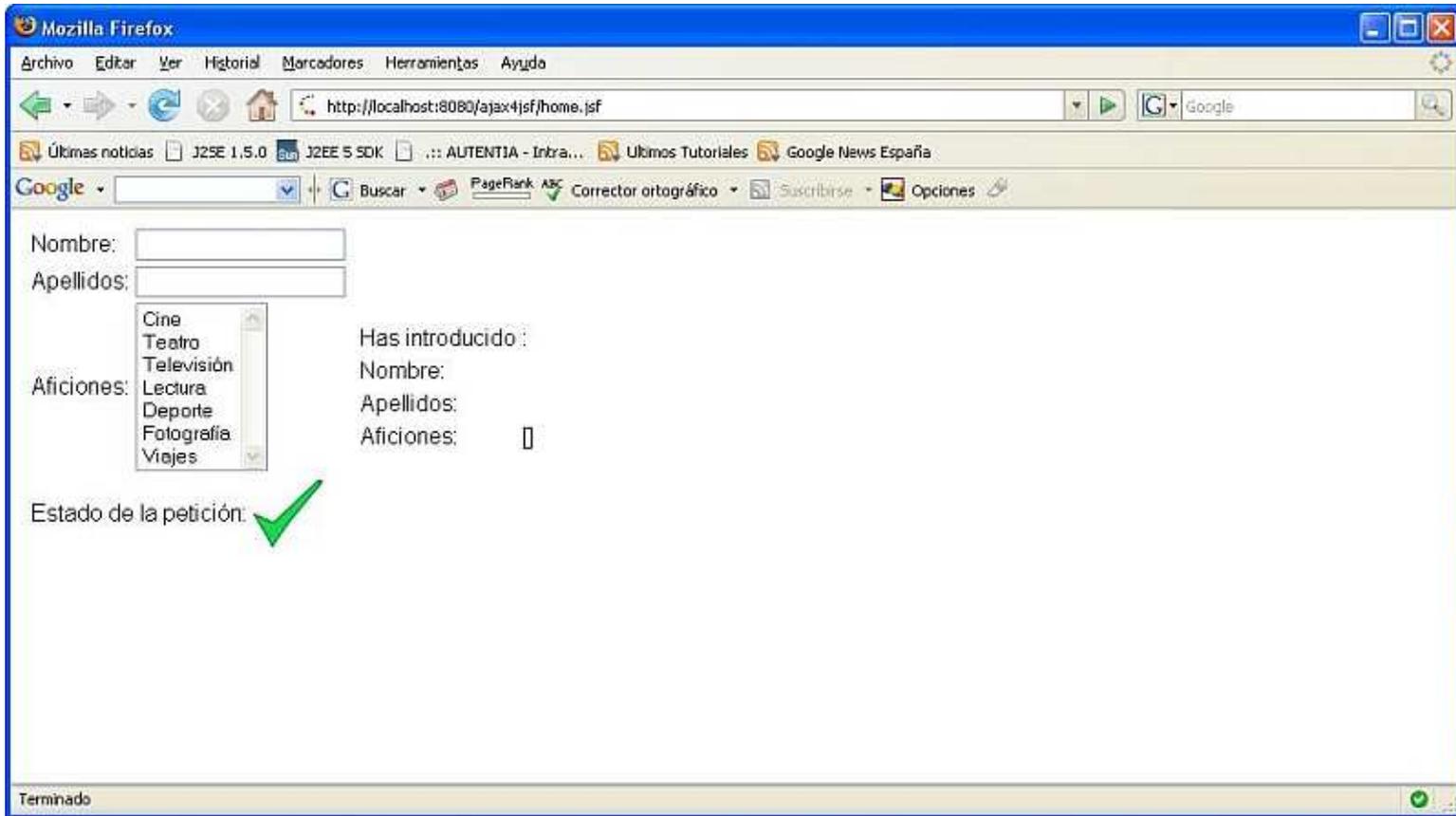
2009-03-26  
[Comercial - Ventas - ALMERIA.](#)

2009-03-12  
[Comercial - Ventas - VALENCIA.](#)

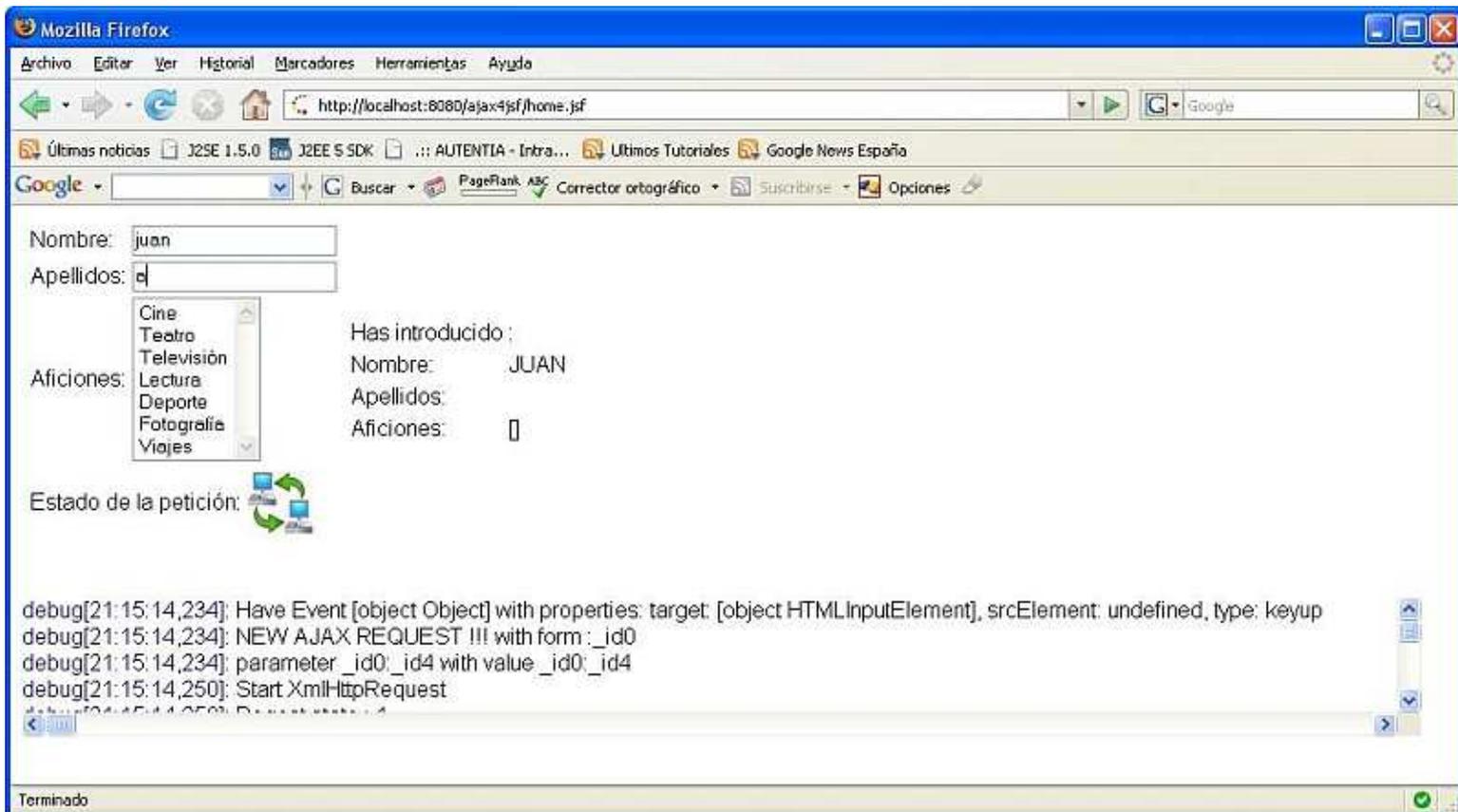
2009-03-12  
[Comercial - Ventas - SEVILLA.](#)

**Anuncios Google**

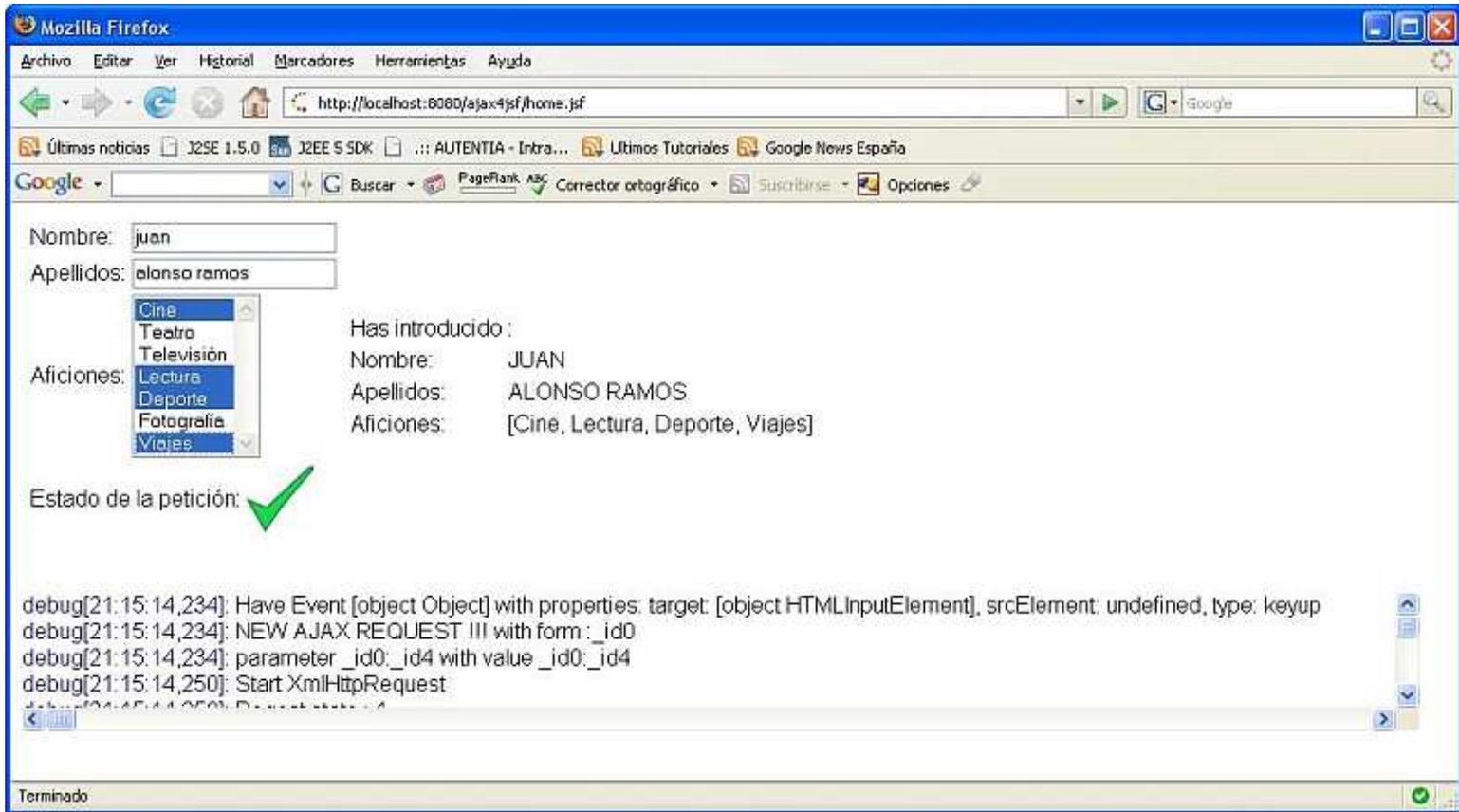
Al arrancar la aplicación se muestra la página de inicio con la imagen de petición procesada ya que actualmente el contenedor está esperando a que el usuario introduzca algo en el formulario.



Introducimos el nombre y vemos cómo se muestra en mayúsculas en la salida de la derecha. Empezamos a introducir los apellidos y mientras se está procesando la información en el bean, lo que vemos es la imagen procesando.jpg que nos indica que aún no hemos recibido la respuesta.



Finalmente seleccionamos los valores del listado de aficiones. Podéis ver cómo van saliendo las trazas que devuelve el contenedor Ajax en la parte inferior.



## 6. Conclusión

A menudo es importante utilizar todos los recursos que tenemos a nuestro alcance para conseguir los efectos deseados en nuestras aplicaciones pero hay que tener cuidado a la hora de elegirlos. Si quieres añadir tecnología Ajax a tu aplicación JSF sin demasiado esfuerzo, este framework está bastante bien ya que resulta muy sencillo de utilizar. Quizá se queda corto en algunas cosas pero parece que en el futuro irán añadiéndole más funcionalidad. Sin duda es una aportación más al mundo open source que siempre es bienvenida.

**¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!**

Muy malo    Malo    Regular    Bueno    Muy bueno



## Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Nombre:

E-Mail:

Comentario:

Enviar comentario

[Texto Legal y condiciones de uso](#)

### Autor Mensaje

Fecha de envío: 2008-04-03 - 05:39:02 PM

Pato

Hola, Realice el ejemplo pero no me funciona, no realiza las funcionalidades que dice que hace.- Bueno.. saludos...

Página 1 de 1

[Anterior](#) [Saltar a la página 1](#) [Siguiente](#)

- [Puedes inscribirte en nuestro servicio de notificaciones 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](mailto:info@autentia.com)

Formación en nuevas tecnologías

## Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
<a href="#">Configuración de la desconexión de usuarios con ICEFaces</a>	Este tutorial muestra la manera de configurar y traducir la ventana de desconexión o pérdida de sesión del usuario en ICEFaces.	2009-06-15	46	Muy bueno	3	
<a href="#">UploadFile con Icefaces + Hibernate + Anotaciones</a>	Este tutorial pretende servir de guía de inicio para todos aquellos usuarios que comienzan a dar sus primeros pasos en ICEfaces, JSF e Hibernate	2009-06-08	378	Bueno	10	
<a href="#">Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.</a>	En este tutorial vamos a hablar de cómo escribir tests funcionales con Selenium IDE sobre interfaces de usuario construidas con librerías de componentes visuales JSF y, en concreto, con Apache Tomahawk y uno de sus componentes.	2009-04-14	1169	Muy bueno	1	
<a href="#">Introducción a JSF Java</a>	Hablaremos del funcionamiento de JSF, las páginas JSF, las etiquetas JSF, los managed beans, la navegación entre páginas, el ciclo de vida de la petición al servidor y otros temas.	2009-03-26	4942	Bueno	31	
<a href="#">El componente outputChart de ICEfaces</a>	En este tutorial se explica cómo utilizar el componente outputChart de ICEfaces para crear gráficas circulares o de ejes y mostrarlas dentro de las páginas JSP.	2008-12-29	1044	Muy bueno	9	
<a href="#">Cómo crear carruseles con detalle con jcarrusel y jtip</a>	Nuestro compañero Raul, nos explica, como ya hiciera con Slimbox y modalbox el uso de unas nuevas librerías que nos ayudarán a hacer nuestras webs más vistosas y amigables.	2008-11-17	2192	Muy bueno	9	
<a href="#">Migración de JSP a Facelets</a>	Si tienes alguna aplicación que mantener basada en JSF y construida en JSPs, este tutorial te servirá para evaluar el coste de su migración a Facelets	2008-10-22	3326	Bueno	6	
<a href="#">Arquetipos de maven: cómo crear, distribuir y generar proyectos con JSF e ICEfaces, JBoss y EJB3</a>	Maven permite la creación de arquetipos de manera muy flexible. En este tutorial veremos cómo crear y distribuir uno que integre las tecnologías JSF e IceFaces, JBoss y EJB3	2008-06-09	4514	Muy bueno	8	
<a href="#">Prototype.js: la sombra que se esconde detrás de todo</a>	¿Usas prototypejs o script.aculo.us? ¿Quizás IceFaces o RichFaces? Si usas Ajax este tutorial te resultará interesante	2008-05-05	5215	Bueno	8	
<a href="#">Icefaces, JBoss, Maven2 y EJB3: Parte 5</a>	Quinta y última parte de este completo tutorial. Nuestro compañero utiliza ICEfaces para el ejemplo de la biblioteca y ofrece el código fuente para su descarga	2008-01-23	4986	Muy bueno	4	

### 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](mailto:rcanales@adictosaltrabajo.com) para su resolución.