



Componentes de presentación

- Sesión 1: Introducción a JSF

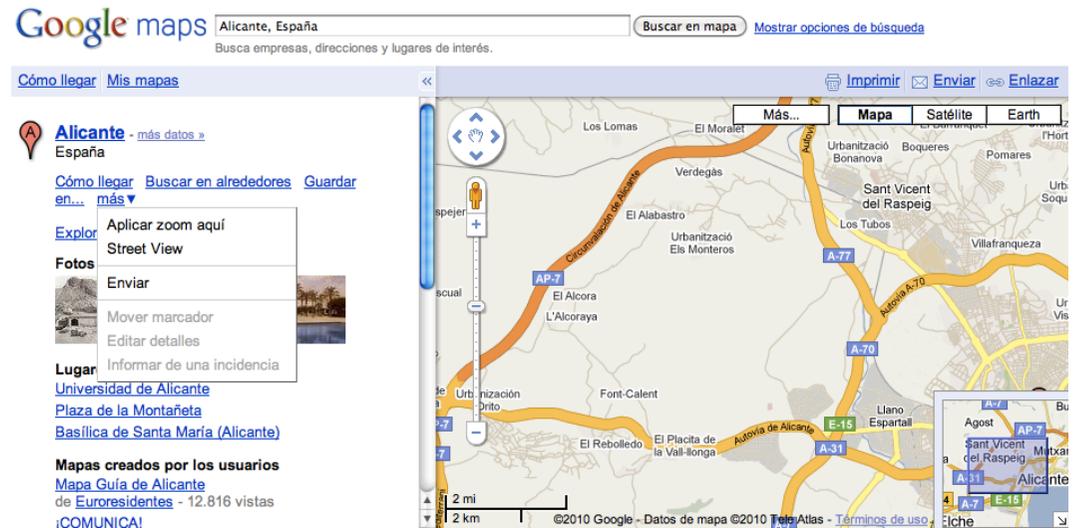


Índice

- Frameworks RIA basados en el servidor
- Características de JSF
- Facelets
- Un ejemplo con RichFaces

Ventajas RIA

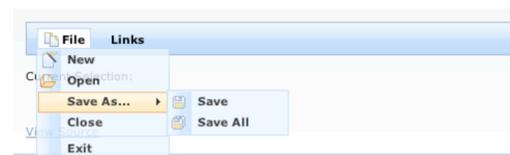
- Aplicaciones RIA vs. aplicaciones web tradicionales
 - Una única página
 - Mayor interacción con el usuario mediante JavaScript
 - Comunicación con el servidor para actualizar partes de la página



Implementación RichFaces de JBoss de JSF

	Expenses			subtotals
	Meals	Hotels	Transport	
San Jose				
25-Aug-97	\$37.74	\$112.00	\$45.00	
26-Aug-97	\$27.28	\$112.00	\$45.00	
	\$65.02	\$224.00	\$90.00	\$379.02
Seattle				
27-Aug-97	\$96.25	\$109.00	\$36.00	
28-Aug-97	\$35.00	\$109.00	\$36.00	
	\$131.25	\$218.00	\$72.00	\$421.25
Totals	\$196.27	\$442.00	\$162.00	\$800.27

Componente Grid



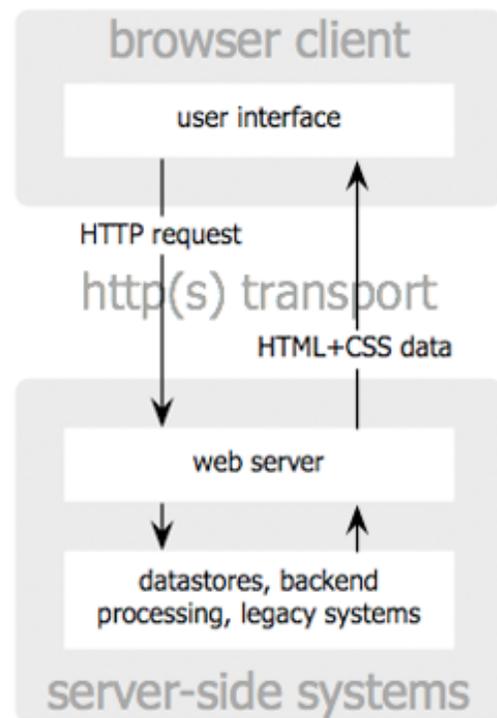
Componente Drop Down Menu



Componente Tree

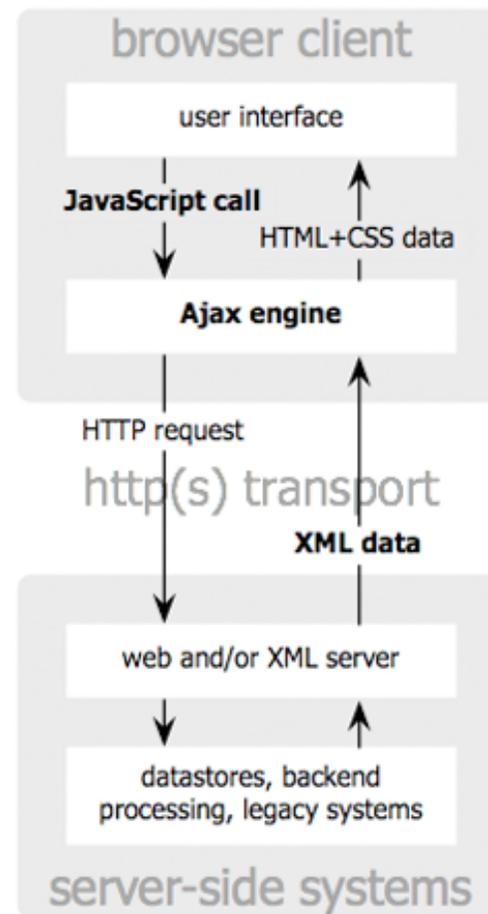


Modelo RIA



classic web application model

Jesse James Garrett / adaptivepath.com



Ajax web application model



Solución 1: JavaScript

- Frameworks:
 - jQuery
 - Prototype
 - ExtJS
 - ...

Ejemplo de jQuery

```
$(document).ready(function() {  
  // generate markup  
  $("#rating").append("Please rate: ");  
  
  for ( var i = 1; i <= 5; i++ )  
    $("#rating").append("<a href='#'>" + i + "</a> ");  
  
  // add markup to container and apply click handlers to anchors  
  $("#rating a").click(function(e){  
    // stop normal link click  
    e.preventDefault();  
  
    // send request  
    $.post("rate.php", {rating: $(this).html()}, function(xml) {  
      // format and output result  
      $("#rating").html(  
        "Thanks for rating, current average: " +  
        $("average", xml).text() +  
        ", number of votes: " +  
        $("count", xml).text()  
      );  
    });  
  });  
});
```



Solución 2: RIA basada en el servidor

- Frameworks:

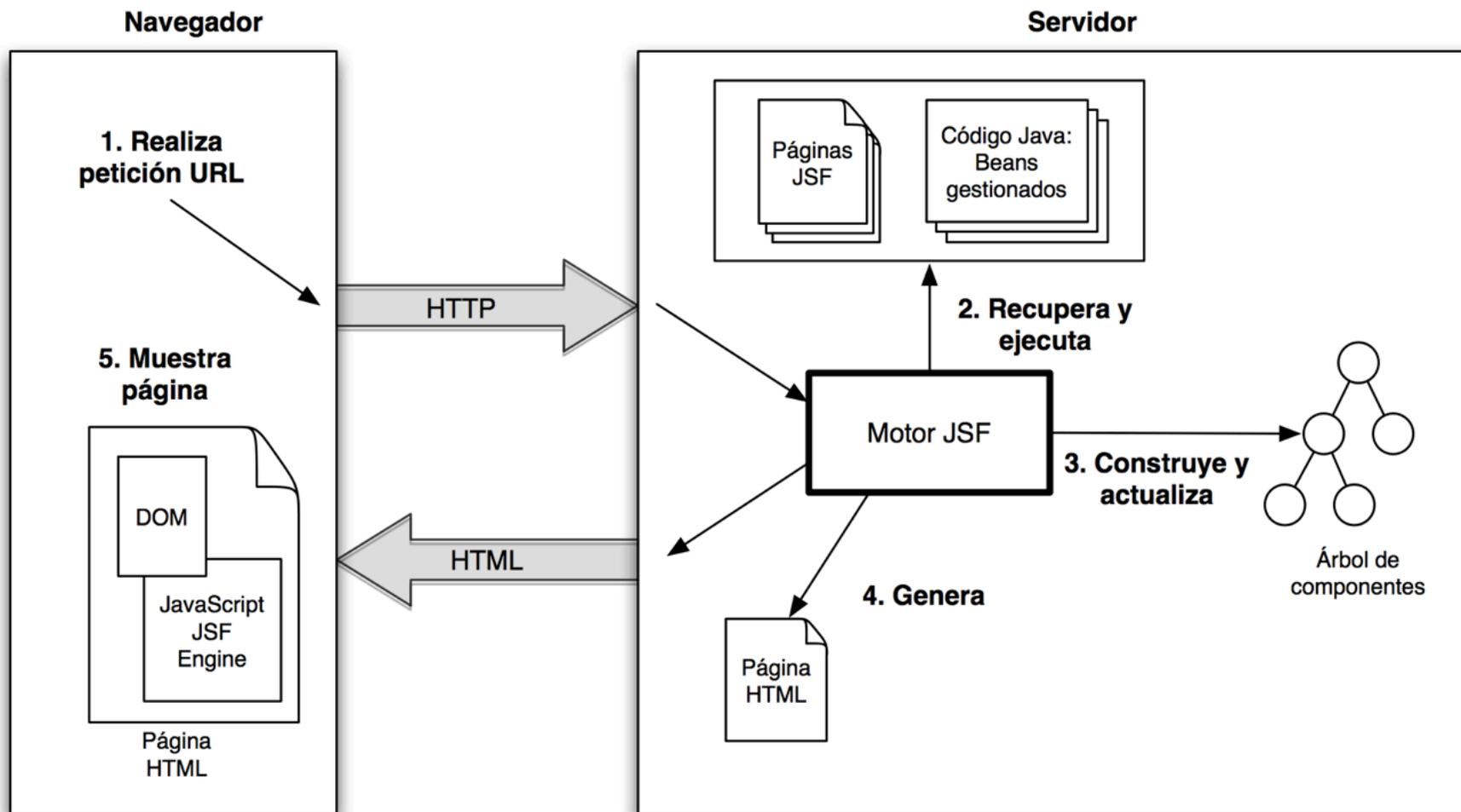
- JSF
- GWT
- ZK
- ...

Ejemplo de JSR RichFaces

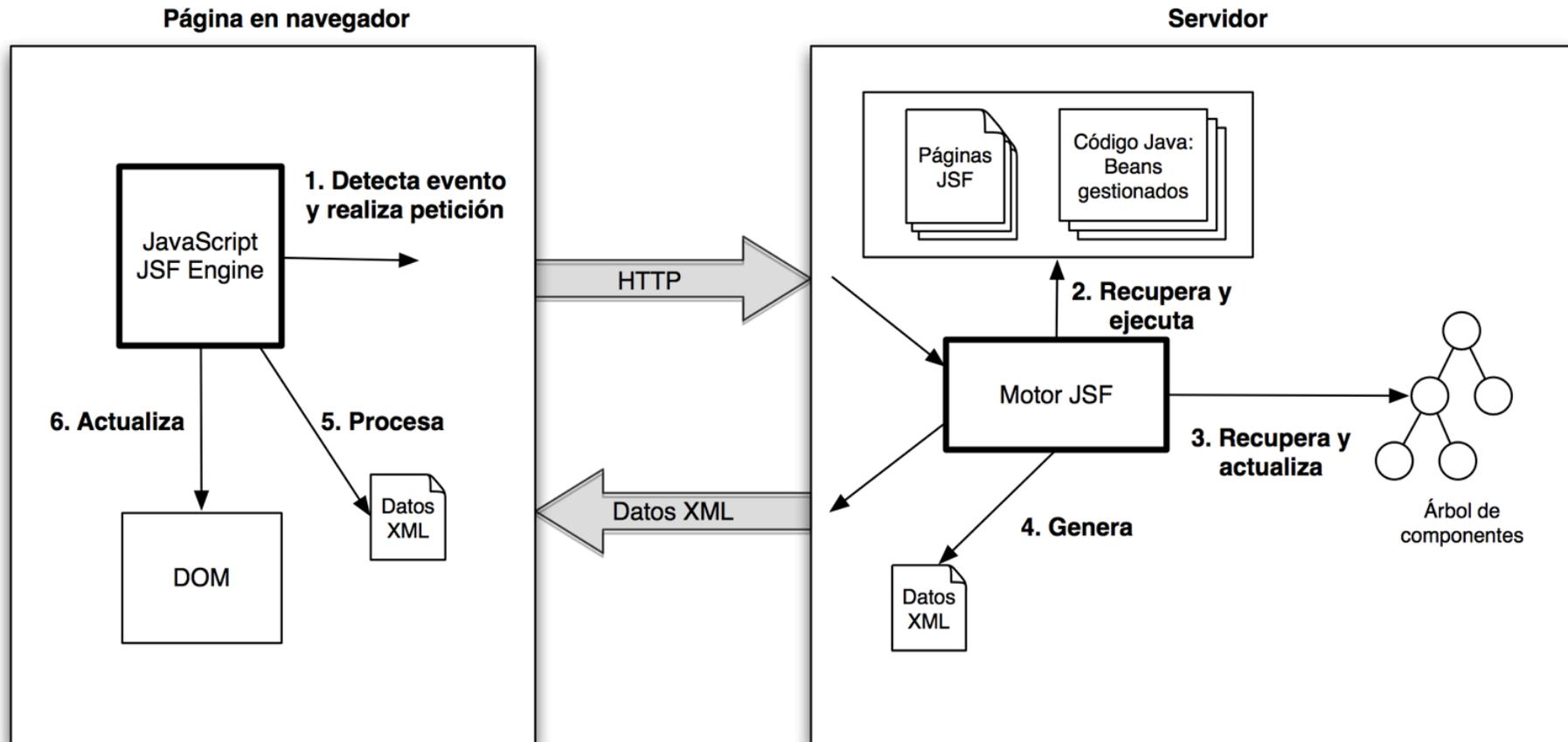
```
<rich:panel>
  <f:facet name="header">
    <h:outputText value="Car Store"></h:outputText>
  </f:facet>
  <h:form>
    <rich:dataGrid value="#{carsBean.allInventoryItems}" var="car"
      columns="3" elements="9" width="600px" border="0">
      <rich:panel bodyClass="tbody">
        <f:facet name="header">
          <h:outputText value="#{car.vendor} #{car.model}"></h:outputText>
        </f:facet>
        <h:panelGrid columns="2">
          <h:outputText value="Price:" styleClass="label"></h:outputText>
          <h:outputText value="#{car.price}" />
          <h:outputText value="Mileage:" styleClass="label"></h:outputText>
          <h:outputText value="#{car.mileage}" />
          <h:outputText value="VIN:" styleClass="label"></h:outputText>
          <h:outputText value="#{car.vin}" />
          <h:outputText value="Stock:" styleClass="label"></h:outputText>
          <h:outputText value="#{car.stock}" />
        </h:panelGrid>
      </rich:panel>
    <f:facet name="footer">
      <rich:dataScroller />
    </f:facet>
  </rich:dataGrid>
</h:form>
</rich:panel>
```



Funcionamiento RIA basado en servidor (1)



Funcionamiento RIA basado en servidor (2)





Introducción a JSF

- JSF permite programar la interfaz de una aplicación web (o aplicación cliente/servidor en general).
 - Definición de la interfaz mediante componentes
 - Conexión de la interfaz con el servidor, donde se sitúa la lógica de negocio y los datos de la aplicación.
 - Posibilidad de componentes ricos que incluyen código Ajax.
- La definición de la interfaz de usuario se realiza de forma declarativa: un fichero XML (HTML con JSP o XHTML).

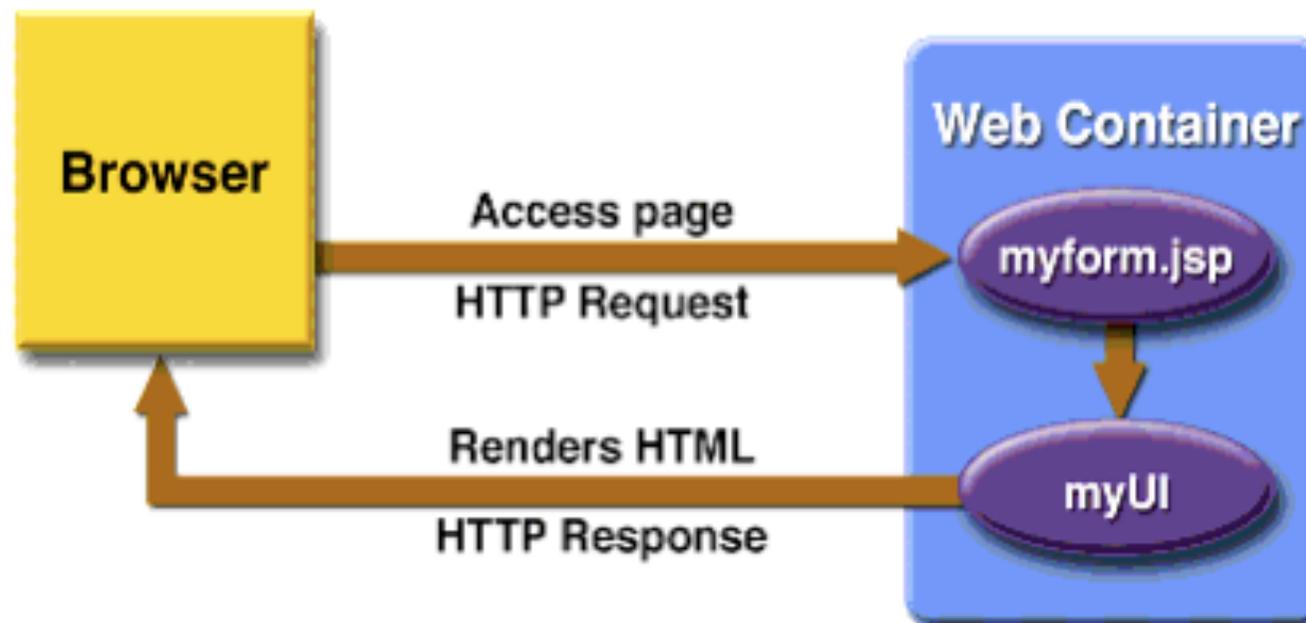


JavaServer Faces ofrece

- Una implementación del patrón MVC orientado a entornos web, cliente/servidor
- Un conjunto estándar de componentes gráficos para la web
- Vinculación entre componentes y *beans* de Java
- Soporte para validación y conversión automática de datos
- Modelo de navegación entre páginas
- Modelo de eventos similar al de Swing
- Soporte para la internacionalización



Funcionamiento básico de JSF





Especificaciones de JSF

- JSF 1.0 (marzo 2004) - Java EE 4
 - Funcionamiento básico de JSF
 - Beans gestionados
 - Lenguaje JSF EL
 - Componentes básicos
 - Navegación entre vistas
- JSF 1.2 (mayo 2006) - Java EE 5
 - Correcciones
 - Lenguaje EL unificado que integra JSP y JSF
- JSF 2.0 (julio 2009) - Java EE 6
 - Facelets
 - Soporte para Ajax
 - Componentes múltiples
 - Gestión de recursos (hojas de estilo, imágenes, etc.)
 - Facilidad de desarrollo y despliegue
- JSF 2.1 (octubre 2010). Se lanza como versión de mantenimiento, con mínimos cambios con respecto a la anterior.
 - Convierte las taglibs de JSP a facelet taglibs, permitiendo una mejor transición de JSF 1.2 a la última especificación
 - Incorpora mejores mecanismos de caché
 - Existen mapeos por defecto en el Faces Servlet
- JSF 2.2 (en desarrollo)
 - Podemos ver qué se cuece para la siguiente versión en <http://jdevelopment.nl/jsf-22/>



Implementaciones de JSF

- Librerías básicas
 - Mojarra (Oracle/Sun)
 - MyFaces (Apache)
- Librerías de componentes
 - RichFaces (JBoss)
 - ADF (Oracle)
 - Tomahawk (Apache)
 - IceFaces
 - PrimeFaces
 - OpenFaces

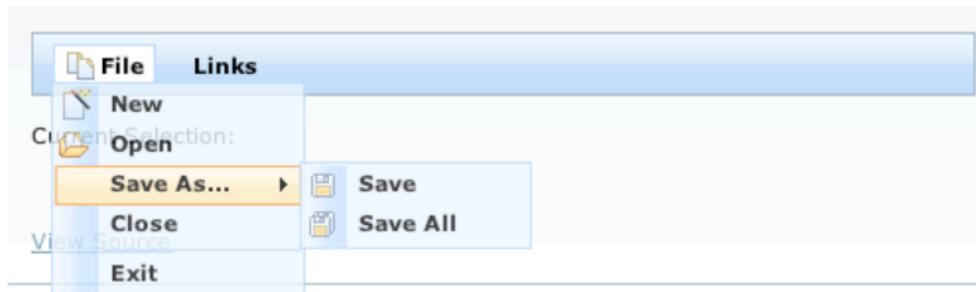


RichFaces

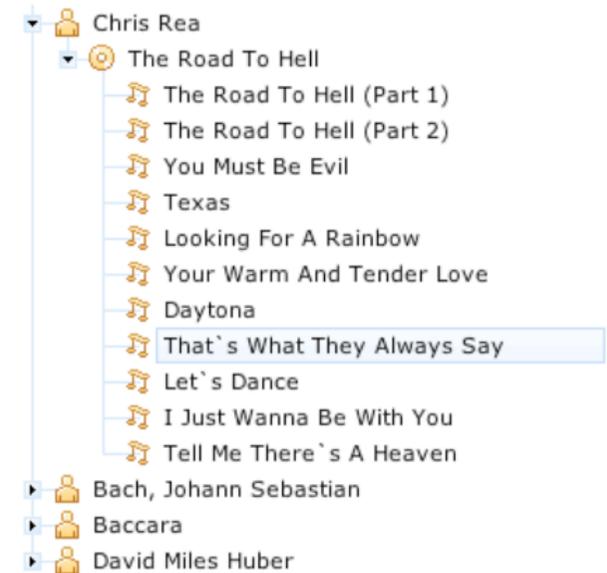
Implementación RichFaces de JBoss de JSF

	Expenses			subtotals
	Meals	Hotels	Transport	
San Jose				
25-Aug-97	\$37.74	\$112.00	\$45.00	
26-Aug-97	\$27.28	\$112.00	\$45.00	
	\$65.02	\$224.00	\$90.00	\$379.02
Seattle				
27-Aug-97	\$96.25	\$109.00	\$36.00	
28-Aug-97	\$35.00	\$109.00	\$36.00	
	\$131.25	\$218.00	\$72.00	\$421.25
Totals	\$196.27	\$442.00	\$162.00	\$800.27

Componente **Grid**



Componente **Drop Down Menu**



Componente **Tree**



Páginas JSF = XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>Welcome</title>
  </h:head>
  <h:body>
    <h:form>
      <h3>Please enter your name and password.</h3>
      <table>
        <tr>
          <td>Name:</td>
          <td><h:inputText value="#{user.name}"/></td>
        </tr>
        <tr>
          <td>Password:</td>
          <td><h:inputSecret value="#{user.password}"/></td>
        </tr>
      </table>
      <p><h:commandButton value="Login" action="welcome"/></p>
    </h:form>
  </h:body>
</html>
```





Librerías de etiquetas de JSF

Librería de etiquetas	Descripción	Ejemplo
JSTL Core	Etiquetas estándar JSP	<code><c:forEach></code> <code><c:catch></code>
JSTL Functions	Funciones estándar JSTL	<code><fn:toUpperCase></code> <code><fn:toLowerCase></code>
Facelets	Lenguaje de plantillas	<code><ui:component></code> <code><ui:insert></code>
JSF HTML	Componentes estándar JSF basados en HTML	<code><h:body></code> <code><h:inputText></code>
JSF Core	Componentes específicos de JSF	<code><f:actionListener></code> <code><f:attribute></code>
RichFaces	Componentes específicos de RichFaces	<code><rich:dataTable></code> <code><rich:panel></code>
Ajax RichFaces	Funciones Ajax de RichFaces	<code><a4j:poll></code> <code><a4j:commandButton></code>



Prefijos y cabecera XHTML

Librería de etiquetas	Prefijo	URI
JSTL Core	c:	http://java.sun.com/jsp/jstl/core
JSTL Functions	fn:	http://java.sun.com/jsp/jstl/functions
JSF Facelets	ui:	http://java.sun.com/jsf/facelets
JSF HTML	h:	http://java.sun.com/jsf/htm
JSF Core	f:	http://java.sun.com/jsf/core
RichFaces	rich:	http://richfaces.org/rich
Ajax RichFaces	a4j:	http://richfaces.org/a4j

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:a4j="http://richfaces.org/a4j"
      xmlns:rich="http://richfaces.org/rich">
```



Facelets

- Lenguaje de plantillas para componer las vistas JSF
- Algunas características
 - Definición de plantillas (como en Tiles)
 - Composición de componentes
 - Etiquetas para definir funciones y lógica
 - Desarrollo de páginas amistoso para el diseñador
 - Posibilidad de crear librerías de componentes

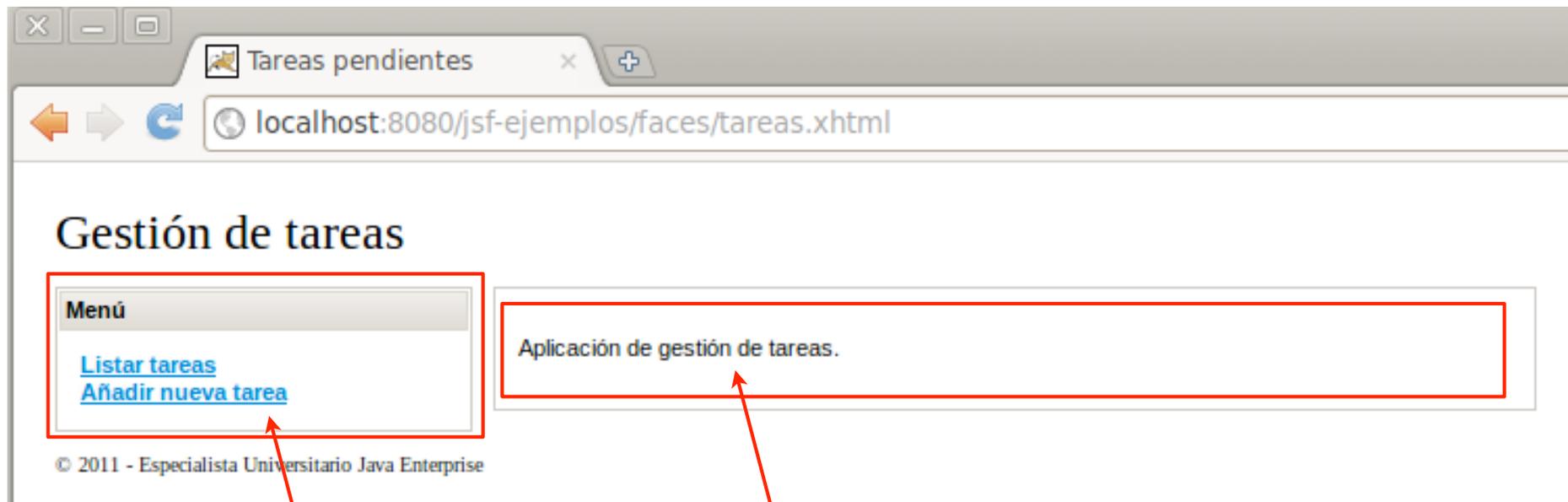


Etiquetas facelets

Etiqueta	Descripción
<code><ui:include></code>	Incluye contenido de otro fichero XHTML
<code><ui:composition></code>	Cuando se usa sin el atributo <code>template</code> una composición es una secuencia de elementos que se pueden insertar en algún otro lugar (mediante la etiqueta <code><ui:include></code> por ejemplo). La composición puede tener partes variables especificadas con el elemento hijo <code><ui:insert></code> . Cuando se usa con el atributo <code>template</code> , se carga la plantilla especificada. Los elementos hijos (etiquetas <code><ui:define></code>) determinan las partes variables de la plantilla. El contenido de la plantilla reemplaza esta etiqueta.
<code><ui:decorate></code>	Cuando se usa sin el atributo <code>template</code> especifica una página en la que se pueden insertar otras partes. Las partes variables se especifican con el elemento hijo <code><ui:insert></code> .
<code><ui:define></code>	Define el contenido que se inserta en una plantilla con un <code><ui:insert></code> que empareja.
<code><ui:insert></code>	Define un lugar en el que se va a insertar contenido en una plantilla. El contenido se define en la etiqueta que carga la plantilla utilizando la etiqueta <code><ui:define></code> .
<code><ui:param></code>	Especifica un parámetro que se pasa a un fichero incluido o a una plantilla.
<code><ui:component></code>	Esta etiqueta es idéntica a <code><ui:composition></code> , excepto en que crea un componente que se añade al árbol de componentes.
<code><ui:fragment></code>	Es idéntica a <code><ui:decorate></code> excepto que crea un componente que se añade al árbol de componentes.
<code><ui:debug></code>	Permite mostrar al usuario una ventana de depuración que muestra la jerarquía de componentes de la página actual y las variables en el ámbito de la aplicación.
<code><ui:remove></code>	JSF elimina todo lo que hay dentro de una etiqueta <code><ui:remove></code> .
<code><ui:repeat></code>	Itera sobre una lista, array, result set o un objeto individual.

Un ejemplo

- Plantilla: `templates/principal.xhtml`
- Menú: `templates/includes/menu.xhtml`
- Página principal: `tareas.xhtml`



Fichero `menu.xhtml`

Contenido variable definido en `principal.xhtml` la etiqueta `<ui:insert name="contenido">`



Fichero templates/principal.xhtml

```
<h:head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Tareas pendientes</title>
  <link rel="stylesheet" type="text/css" href="css/estilos.css"/>
</h:head>
<h:body>
  <table border="0" cellpadding="0" cellspacing="0">
  <tbody>
    <tr> <td colspan="2"><div class="titulo">Gestión de tareas</div></td> </tr>
    <tr>
      <td> <div class="menu"> <ui:include src="includes/menu.xhtml"/> </div> </td>
      <td valign="top">
        <div class="contenido">
          <rich:panel>
            <ui:insert name="contenido"> Contenido por defecto </ui:insert>
          </rich:panel>
        </div>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <div class="pie"> &copy; 2011 - Especialista Universitario Java Enterprise </div>
      </td>
    </tr>
  </tbody>
</table>
</h:body>
```



Fichero templates/includes/menu.xhtml (I)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:rich="http://richfaces.org/rich">
<ui:composition>
  <rich:panel header="Menú" >
    <a href="#">Listar tareas</a><br/>
    <a href="#">Añadir nueva tarea</a>
  </rich:panel>
</ui:composition>
</html>
```



Fichero templates/includes/menu.xhtml (II)

- En algunos casos se nos permite quitar las etiquetas html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:rich="http://richfaces.org/rich">
  <rich:panel header="Menú" >
    <a href="#">Listar tareas</a><br/>
    <a href="#">Añadir nueva tarea</a>
  </rich:panel>
</ui:composition>
```

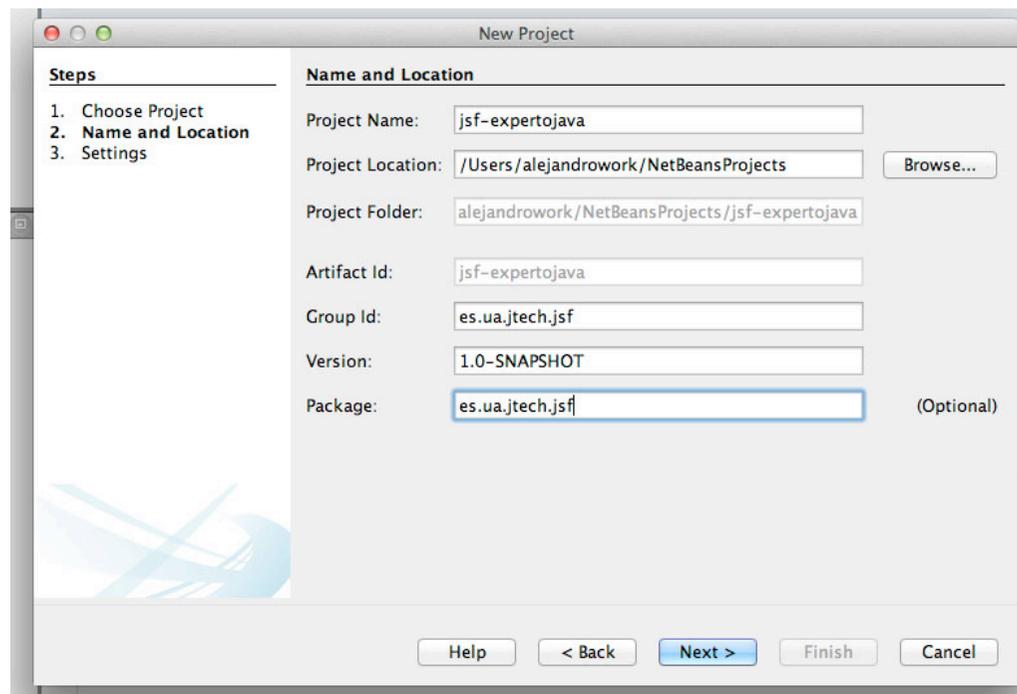


Fichero tareas.xhtml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:a4j="http://richfaces.org/a4j"
      xmlns:rich="http://richfaces.org/rich">
<body>
<ui:composition template="templates/principal.xhtml">
  <ui:define name="contenido">
    <p>Aplicación de gestion de tareas.</p>
  </ui:define>
</ui:composition>
</body>
</html>
```

El primer proyecto RichFaces

- RichFaces 4.0
- Dependencias Maven: incluir el repositorio de JBoss en el fichero pom.xml
- Creación de proyecto con NetBeans





Configuración de una aplicación JSF

- Una aplicación JSF es una aplicación web estándar
- Librerías JSF (JAR)
- Ficheros configuración JSF
 - `web.xml`: redirigir las peticiones al servlet `FacesServlet`
 - `faces-config.xml`: fichero de configuración propio en el que se definen:
 - Beans** de la aplicación, con sus nombres y propiedades iniciales
 - Reglas de validación** de los componentes de entrada
 - Reglas de navegación** entre distintas páginas de la aplicación
 - Ficheros de recursos**, para la internacionalización



Demo



¿Y Ajax?

- La mayoría de librería de componentes tienen componentes Ajax: componentes que generan código HTML con Javascript Ajax
- Ventajas: el código Ajax está “escondido” en el comportamiento del componente. No hay que escribir nada de código.
- Ejemplo en RichFaces:

2123 34| 2123 34

```
<h:form id="myform">
  <h:panelGrid columns="2">
    <h:inputText id="myinput" value="#{userBean.name}">
      <a4j:support event="onkeyup" render="outtext" />
    </h:inputText>
    <h:outputText id="outtext" value="#{userBean.name}" />
  </h:panelGrid>
</h:form>
```



Resumiendo: ¿qué es JSF?

- Un conjunto de controles GUI basados en Web y manejadores asociados
 - JSF proporciona una gran cantidad de controles GUI orientados a HTML junto con código para manejar los eventos
- Una versión mejorada de Struts
 - Al igual que Struts, JSF puede ser visto como un framework MVC para construir formularios HTML, validar sus valores, invocar la lógica de negocio y mostrar los resultados.
- La parte VC del MVC
 - El *modelo* de JSF son los *beans de respaldo*, que no representan el modelo de la aplicación. Por eso, JSF se puede ver sobre todo como un framework para construir la *Vista* y el *Controlador*.



Resumiendo: ventajas de JSF

- Controles GUI HTML
- Manejo de eventos
- Managed beans
- Lenguaje de expresiones
- Conversión y validación de las entradas en los campos
- Estándar
- Cada vez más soporte en la industria (plugins, entornos de desarrollo, etc.)

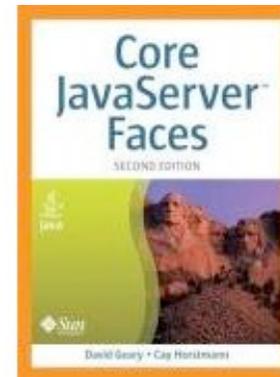
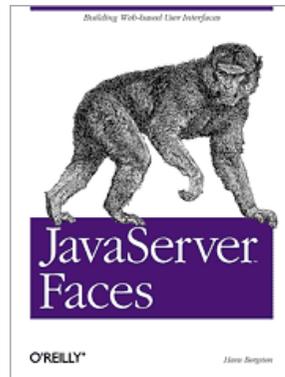


Alternativas

- Si necesitamos una aplicación web muy interactiva (cliente rico con mucha interacción entre los componentes de la interfaz) quizás JSF no es adecuado.
- Frameworks orientados a clientes ricos con Java en el servidor
 - GWT Google Web Toolkit
 - Flex de Adobe
 - Java FX de Sun
 - ZK

Para Saber Más

- Bibliografía
 - **JavaServerFaces** de Hans Bergstern
 - **JavaServerFaces, the Complete Reference** de Shalk y Burns
 - **Core JavaServer Faces 3rd ed.** de Geary y Hortsman



- Enlaces
 - Página oficial de Sun: <http://java.sun.com/javaee/javaxserverfaces/>
 - Implementación de Sun en GlassFish: <https://javaxserverfaces.dev.java.net/>
 - Blog muy interesante: <http://balusc.blogspot.com/>



¿Preguntas?