



# JavaServer Faces

- Sesión 4: Internacionalización.  
Mensajes Flash. RichFaces



# Índice

- Internacionalización
  - Declaración
  - Múltiples idiomas
- Mensajes Flash
- RichFaces
  - Librerías de componentes profesionales
  - Características de RichFaces
  - Peticiones Ajax y ciclo de vida
  - Skins
  - Componentes Ajax
  - Redibujado de componentes
  - Algunos ejemplos



# Internacionalización

- Internacionalización: diseñar software que pueda adaptarse a diferentes idiomas y regiones cambios de ingeniería de código.
- Localización: adaptar el software para una región específica mediante la adición de componentes específicos de un locale y la traducción de los textos.
- Buena práctica: colocar todos los mensajes en una misma localización central.
- Los textos se guardan en un fichero `.properties`

```
src/es/ua/jtech/i18n/messages.properties
```



# Indicar fichero de recursos (I)

- En el `faces-config.xml`

```
<?xml version="1.0"?>
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-
  facesconfig_2_0.xsd"
  version="2.0">
  <application>
    <resource-bundle>
      <base-name>es.ua.jtech.messages</base-name>
      <var>msgs</var>
    </resource-bundle>
  </application>
</faces-config>
```



## Indicar fichero de recursos (II)

- En cada vista/plantilla

```
<f:loadBundle basename="es.ua.jtech.messages"  
var="msgs" />
```

- En cualquier caso, podemos acceder al fichero de recursos mediante expresiones EL:

```
#{msgs.login_nombre}
```



# Placeholders en los mensajes

- Muchos mensajes tienen partes variables en el mismo.

```
tuPuntuacion=Tienes {0} puntos.
```

- Los placeholders se numeran partiendo de cero.
- Uso de `f:param` para rellenar los placeholders

```
<h:outputFormat value="#{msgs.tuPuntuacion}">  
  <f:param value="#{marcadorBean.puntos}" />  
</h:outputFormat>
```



# Acceder a los recursos desde el código Java

```
public String checkDni(){
    if("11222333".equals(dni.getNumero()) && "N".equalsIgnoreCase(dni.getLetra())){
        FacesContext facesContext = FacesContext.getCurrentInstance();
        ResourceBundle msgs =
facesContext.getApplication().getResourceBundle(facesContext, "msgs");
        facesContext.getExternalContext().getFlash().put("message",
msgs.getString("reset_pass_ok"));
        return "OK";
    }

    return "ERROR";
}
```



# Múltiples idiomas

- Indicar en el faces-config.xml los idiomas soportados

```
<faces-config>
  <application>
    <locale-config>
      <default-locale>en</default-locale>
      <supported-locale>de</supported-locale>
    </locale-config>
  </application>
</faces-config>
```

- Idioma por defecto de la aplicación: el del navegador
  - Cabecera Accept-Language





# Cambiar el idioma

- De manera programada

```
UIViewRoot viewRoot = FacesContext.getCurrentInstance().getViewRoot();  
viewRoot.setLocale(new Locale("de"));
```

- Usando el componente `f:view`

```
<f:view locale="de">
```

```
<f:view locale="#{langController.lang}">
```



# Mensajes Flash

- Mensajes que se declaran en una petición y se usan en la siguiente.
- Tomado prestado de Ruby on Rails.
- Uso habitual: paso de mensajes.

```
ExternalContext.getFlash().put("message", "Campo actualizado correctamente");
```

- Recuperamos el mensaje de dos maneras:

```
#{flash.message}
```

```
#{flash.keep.message}
```



# Librerías de componentes profesionales

- JSF Matrix: <http://www.jsfmatrix.net/>

Others: [Ajax4jsf](#) [ILOG](#) [Backbase](#) [G4JSF](#) [jenia4faces](#) [VisualWebPack](#) [DynaFaces](#) [IBM Impl](#)

[Mojarra](#) [Dinamica](#) [jQuery](#) [Echo 3](#)

[QuipuKit](#) [ZK](#) [Simplica](#) [ExtJS](#) [RCFaces](#) [WebGalileoFaces](#) [BluePrints](#) [Woodstock](#) [Netadvantage](#)

Hide All	Hide	Hide	Hide	Hide	Hide	Hide
	Tomahawk	Tobago	Trinidad	ICEfaces	JBoss RichFaces	Oracle ADF
<b>Technology</b>	JSF	JSF	JSF	JSF	JSF	JSF
<b>Homepage</b>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>
<b>Demo URL</b>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>
<b>Demo WAR</b>						
<b>Docu Quantity</b>						
<b>Docu URL</b>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	
<b>Forum URL</b>			<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>	<a href="#">URL</a>
<b>Free</b>						
<b>Job Hits</b>	2	?	2	1	4	15
<b>Google Hits</b>	81.300	17.800	46.700	53.100	53.700	64.400
Components	Tomahawk	Tobago	Trinidad	ICEfaces	JBoss RichFaces	Oracle ADF
<b>Datatable</b>		AJAX	AJAX	AJAX	AJAX	
<b>Tree</b>			AJAX	AJAX	AJAX	
<b>Treetable</b>			AJAX		AJAX	
<b>Livescrolling</b>				AJAX	AJAX	



# Características de RichFaces

- Proyecto de JBoss (RedHat)
- Licencia LGPL v2.1
- Librería completa de componentes
- Framework maduro, formado por 2 librerías: Ajax4jsf `<a4j:>` y componentes RichFaces `<rich:>`
- Los componentes Ajax pueden usarse con componentes normales JSF sin modificar casi nada las páginas existentes
- Posibilidad de gestionar recursos en la propia aplicación web: imágenes, código javascript y hojas de estilo CSS
- Facilidades para generar recursos en caliente: gráficas, imágenes, sonido



## Versión 4.2

- Versión 4.2 en el mercado desde abril de 2012
  - Componentes nuevos
    - rich:editor
    - rich:pickList
    - rich:orderingList
    - rich:notify
  - Mejoras en componentes ya existentes (drag and drop, fileUpload, ...)
  - Uso de HTML5 para mayor compatibilidad con dispositivos móviles
  - *Resource packaging*



# *Demo*

<http://showcase.richfaces.org>

(RichFaces 4.2.2)

<http://www.jtech.ua.es/richfaces-showcase>

(RichFaces 4.0)



# Peticiones Ajax

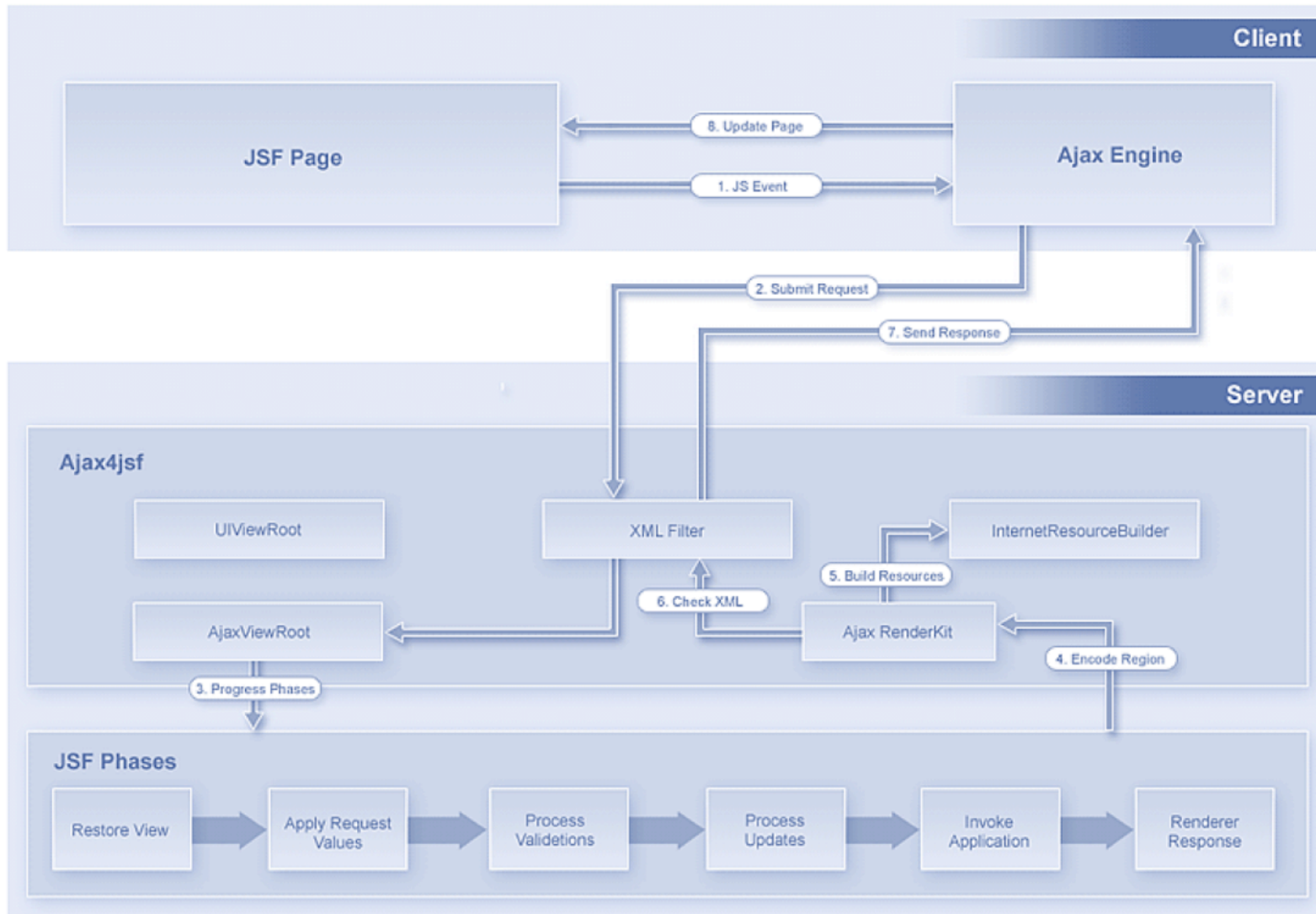
- Con el soporte Ajax es posible realizar peticiones al servidor y hacer que sólo se redibujen los componentes que nos interesan.
- También es posible ligar eventos javascript de los componentes HTML con llamadas a métodos en el servidor.

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>
```



# Funcionamiento de las peticiones Ajax







# Skins

- En RichFaces podemos configurar el aspecto de la aplicación mediante *skins*
- Se define mediante un parámetro en el fichero `web.xml`

```
<context-param>  
  <param-name>org.richfaces.SKIN</param-name>  
  <param-value>wine</param-value>  
</context-param>
```

```
DEFAULT  
plain  
emeraldTown  
blueSky  
wine  
japanCherry  
ruby  
classic  
deepMarine
```



# Componentes Ajax

- a4j:commandButton

```
<a4j:commandButton value="Enviar" action="#{bean.doAction}  
                    rerender="miBloque" />
```

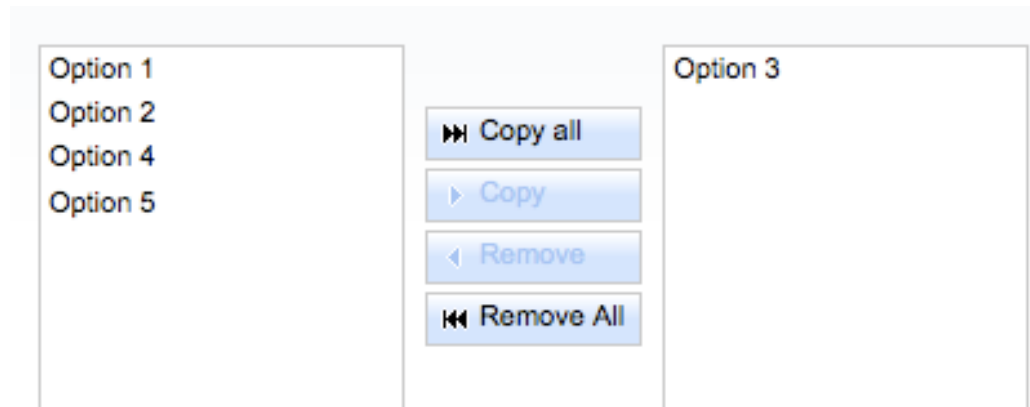
- a4j:support

```
<h:inputText value="#{user.name}">  
  <a4j:support event="onkeyup" action="#{bean.doAction}  
              rerender="miBloque" />  
</h:inputText>
```

- eventos: **onblur**, **onchange**, **onclick**,  
**ondblclick**, **onfocus**, **onkeydown**, **onkeypress**,  
**onkeyup**, **onmousedown**, **onmousemove**,  
**onmouseout**, **onmouseover**, **onmouseup**, **onselect**

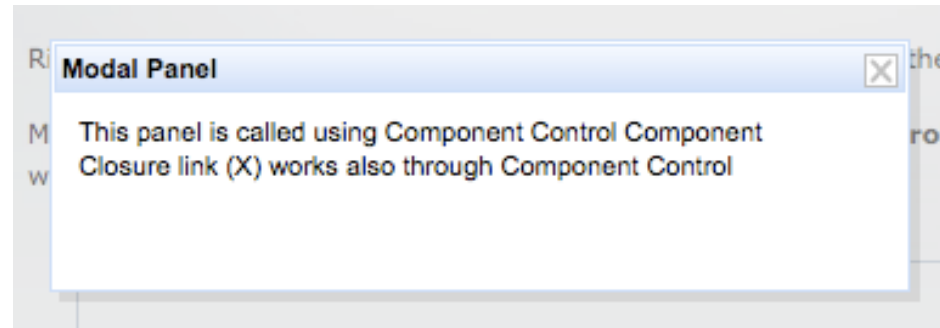


# Componentes: pickList



```
<rich:pickList>
  <f:selectItem itemLabel="Option 1" itemValue="1"/>
  <f:selectItem itemLabel="Option 2" itemValue="2"/>
  <f:selectItem itemLabel="Option 3" itemValue="3"/>
  <f:selectItem itemLabel="Option 4" itemValue="4"/>
  <f:selectItem itemLabel="Option 5" itemValue="5"/>
</rich:pickList>
```

# Componentes: Panel modal



```
<rich:modalPanel id="panel" width="350" height="100">
  <f:facet name="header">
    <h:panelGroup>
      <h:outputText value="Modal Panel"></h:outputText>
    </h:panelGroup>
  </f:facet>
  <f:facet name="controls">
    <h:panelGroup>
      <h:graphicImage value="/images/modal/close.png"
        styleClass="hidelink" id="hidelink" />
      <rich:componentControl for="panel" attachTo="hidelink"
        operation="hide" event="onclick" />
    </h:panelGroup>
  </f:facet>
  <h:outputText
    value="This panel is called using Component Control Component"/>
  <br />
  <h:outputText
    value="Closure link (X) works also through Component Control"/>
</rich:modalPanel>
```



# Componentes: DataGrid

Chevrolet Corvette	Chevrolet Corvette	Chevrolet Corvette
Price: 45816 Mileage: 33761.0 VIN: URVKACNCUWLKGNH Stock: UAVFHNI	Price: 44463 Mileage: 21131.0 VIN: ZMGBHYUGFHAOZO Stock: FSLHYPQ	Price: 40996 Mileage: 19277.0 VIN: NHOTACHIPWPWBCF Stock: WTWGWA
Price: 18730 Mileage: 7154.0 VIN: ZMZCKZRYFMXJXT Stock: FVXHEO	Price: 23933 Mileage: 9642.0 VIN: TFUEBHMVQBJPHH Stock: MPJUGIP	Chevrolet Malibu Price: 50935 Mileage: 9093.0 VIN: IWAMDLTGMZMVGR Stock: JCFLFO
Chevrolet Malibu Price: 31832 Mileage: 53452.0 VIN: HOVLLAHRCRBLNC Stock: CCPAGNC	Chevrolet Malibu Price: 38114 Mileage: 58194.0 VIN: DJALBWTVMWABXO Stock: PJWBJX	Chevrolet Malibu Price: 17472 Mileage: 61349.0 VIN: BQBSEZQKTDLWGP Stock: LBG0YGW

```
<rich:dataGrid value="#{dataTableScrollerBean.allCars}"
  var="car" columns="3" elements="9" width="600px">
  <rich:panel bodyClass="pbody">
    <f:facet name="header">
      <h:outputText value="#{car.make} #{car.model}" />
    </f:facet>
    <h:panelGrid columns="2">
      <h:outputText value="Price:" styleClass="label" />
      <h:outputText value="#{car.price}" />
      <h:outputText value="Mileage:" styleClass="label" />
      <h:outputText value="#{car.mileage}" />
      <h:outputText value="VIN:" styleClass="label" />
      <h:outputText value="#{car.vin}" />
      <h:outputText value="Stock:" styleClass="label" />
      <h:outputText value="#{car.stock}" />
    </h:panelGrid>
  </rich:panel>
  <f:facet name="footer">
    <rich:datascroller></rich:datascroller>
  </f:facet>
</rich:dataGrid>
```

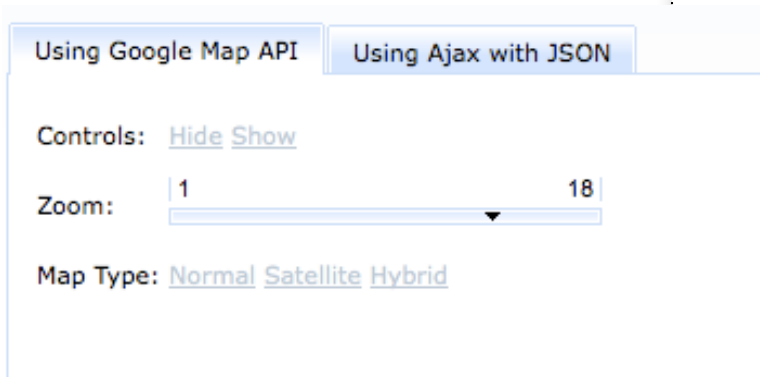
# Componentes: Google map



```
<rich:gmap
  gmapVar="map" zoom="16" style="width:400px;height:400px"
  gmapKey="ABQIAAAA5SzcCLDOLK2VPhx3P-poFxQDT1fstRCWND9TPh4
          hnvi3n3eSLhQH-hQAsES9VPnDb0M9QRvXK83_Lw"
  lat="38.38463"
  lng="-0.51287" />
```



# Componentes: Tab panel



```
<rich:tabPanel switchType="ajax" width="350" height="400">
  <rich:tab label="Using Google Map API">
    <h:panelGrid columns="2" columnClasses="optionList">
      <h:outputText value="Controls:" />
      <h:panelGroup>
        <a href="javascript: void 0" onclick= map.hideControls();>Hide</a>
        <a href="javascript: void 0" onclick= map.showControls();>Show</a>
      <br/>
    </h:panelGroup>
    <h:outputText value="Zoom:" />
    <rich:inputNumberSlider id="zoom" showInput="false"
      minValue="1" maxValue="18" value="{gmBean.zoom}"
      onchange="map.setZoom(this.value)" />
    <h:outputText value="Map Type:" />
    <h:panelGroup>
      <a href="javascript: void 0"
        onclick=map.setMapType(G_NORMAL_MAP);>Normal</a>
      <a href="javascript: void 0"
        onclick=map.setMapType(G_SATELLITE_MAP);>Satellite</a>
      <a href="javascript: void 0"
        onclick=map.setMapType(G_HYBRID_MAP);>Hybrid</a>
    <br/>
    </h:panelGroup>
  </h:panelGrid>
</rich:tab>
  <rich:tab label="Using Ajax with JSON">
    <rich:dataGrid var="place" value="{gmBean.point}" columns="2">
      <h:graphicImage onclick="showPlace('{place.id}')"
        style="cursor:pointer" value="resource://{place.pic}" />
    </rich:dataGrid>
  </rich:tab>
</rich:tabPanel>
```



# Referencias

- *Página principal de RichFaces en JBoss*
- *Documentación de RichFaces*
- *RichFaces*, documento sobre RichFaces en el Wiki de JBoss, con bastantes enlaces interesantes.
- *Servidor Hudson de RichFaces*





# ¿Preguntas?