# **JSF Tag Library**

## Introducción

En el capítulo anterior hicimos mención las bibliotecas de etiquetas (Tag Libraries) que forman parte de JSF. Éstas representan un conjunto de tags que nos evitan tener que escribir "snipplets" de código como hacíamos en las páginas JSP.

A continuación, vamos a presentar las más utilizadas presentando ejemplos de su uso.

### outputText

En este caso, este tag mostrará el valor de la propiedad text (llamando al método getText()) del managed bean "user"

<h:outputText value="#{user.text}"/>

#### textbox

Al enviar el formulario, este tag asignará el valor de "value" a la propiedad userName (llamando al método setUserName(String))

<h:inputText value="#{user.userName}" />

### password

<h:inputSecret value="#{user.password}" />

#### hidden value

<h:inputHidden value="#{user.answer}" />

#### checkboxes

```
<h:graphicImage library="images" name="sofa.png" />
```

que es el equivalente, en HTML, a:

<img src="/JavaServerFaces/faces/javax.faces.resource/sofa.png?ln=images" alt="" />

donde "images" responde al concepto de "biblioteca de recursos", estos deben estar en la carpeta "resources/images" dentro de nuestra aplicación web.

#### **button**

Al presionar el botón, este nos redireccionará a "% APP\_WEB%//faces/login.xhtml"

# El Tag DataTable

DataTable es una estructura más compleja que nos permite desplegar una lista de datos en un formato de filas y columnas.

```
<h:dataTable value="#{order.productos}" var="o">
    <h:column>
      <!-- column header -->
           <f:facet name="header">Número de orden</f:facet>
           <!-- row record -->
           #{o.numeroOrden}
      </h:column>
    <h:column>
           <f:facet name="header">Producto</f:facet>
           #{o.producto}
      </h:column>
    <h:column>
           <f:facet name="header">Precio</f:facet>
          #{o.precio}
      </h:column>
    <h:column>
           <f:facet name="header">Cantidad</f:facet>
           #{o.cantidad}
      </h:column>
</h:dataTable>
```

## **Validators**

JSF nos permite validar los campos de nuestros formularios a través de una serie de "Validators" que vienen incluídos con las librerías "core".

Los mensajes de las validaciones pueden ser modificados a través del archivo de recursos message.properties.

Además, nos permite crear nuestros propios Validators.