Java Server Faces

Introducción

JavaServer Faces (JSF) es una tecnología para aplicaciones web desarrolladas en Java que tiene como objetivo simplificar el desarrollo de interfaces de usuario.

JSF usa JavaServer Pages (JSP), pero también se puede acomodar a otras tecnologías como XUL (XML para la interfaz de usuario).

JSF incluye:

Un conjunto por defecto de componentes para la interfaz de usuario.

Bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.

Un modelo de eventos en el lado del servidor.

Administración de estados.

Beans administrados (Managed Beans).

La especificación de JSF fue desarrollada por la Java Community Process como JSR 127, que definía JSF 1.0 y 1.1, JSR 252 que define JSF 1.2 y JSR 314 para JSF 2.0

Managed Beans

Los Managed Beans (o Beans Administrados) son objetos Java cuyo ciclo de vida (es decir, cuándo son creados y eliminados) corre por cuenta del Servlet Container. Obviamente, nosotros podemos configurar este ciclo de vida en el archivo faces-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
   <managed-bean>
   <managed-bean-name>miCuentaBean</managed-bean-name>
   <managed-bean-class>ar.com.educacionit.jsf.MiCuentaBean</managed-bean-class>
   <managed-bean-scope>session</managed-bean-scope>
   </managed-bean>
```

```
</faces-config>
```

Los beans siempre tendrán un nombre que los identifique, una clase y un ámbito en el cual pueden vivir, en este caso, "session" significa que este bean se crea cuando el usuario ingresa al sitio y se destruye cuando se destruya su sesión de usuario.

La clase Java:

```
package ar.com.educacionit.jsf;
import java.io.Serializable;
public class MiCuentaBean implements Serializable {
     private static final long serialVersionUID = 1L;
     private String name;
     public String getName() {
         return name;
     }
     public void setName(String name) {
         this.name = name;
     }
}
```

Managed Properties

Los Managed Beans pueden tener propiedades que, a su vez, sean Managed Beans también. Esta capacidad de poder administrar Beans cuyas propiedades pueden ser otros beans se denomina "Managed Properties" o

"Propiedades Administradas".

Se configuran a través del archivo faces-config.xml al igual que los Managed Beans:

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
 <managed-bean>
     <managed-bean-name>miCuentaBean</managed-bean-name>
     <managed-bean-class>ar.com.educacionit.jsf.MiCuentaBean</managed-bean-class>
     <managed-bean-scope>session</managed-bean-scope>
 <managed-property>
  cproperty-name>banco/property-name>
  cproperty-class>ar.com.educacionit.jsf.BancoBean/property-class>
  <value>#{bancoBean}</value>
</managed-property>
  </managed-bean>
  <managed-bean>
     <managed-bean-name>bancoBean</managed-bean-name>
     <managed-bean-class>ar.com.educacionit.jsf.BancoBean</managed-bean-class>
     <managed-bean-scope>application</managed-bean-scope>
  </managed-bean>
</faces-config>
```

Páginas JSF

Las páginas JSF son muy parecidas a las páginas JSP. Éstas definen dos bibliotecas al comienzo que son las que contienen un set de componentes a nivel visual.

Este set de componentes tienen una representación visual (HTML + Javascript) y una representación en el servidor (Java) de modo tal que, en caso de necesitarlo, nosotros mismos podríamos crear componentes debiendo, para ello, programar una clase Java y la representación en HTML para ese elemento.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</p>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<a href="http://www.w3.org/1999/xhtml" http://www.w3.org/1999/xhtml" |
   xmlns:f="http://java.sun.com/jsf/core"
   xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>JSF Hola Mundo</title>
  </h:head>
  <h:body>
       <h3>JSF Ejemplo Hola Mundo</h3>
       <h:form>
         <h:inputText value="#{holaBean.nombre}"></h:inputText>
         <h:commandButton value="Bienvenido" action="saludo"></h:commandButton>
       </h:form>
  </h:body>
</html>
```

El valor del campo de texto va a ser cargado en la propiedad "nombre" del bean holaBean que ejecutará la acción "saludo".

Navegación entre páginas

Las reglas de navegación (navigation-rule) nos permite definir a qué páginas puedo llegar mediante distintas acciones desde una página específica

En este caso, la acción "saludo" nos enviará directamente a la página saludo.xhtml.

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
xmlns="http://java.sun.com/xml/ns/javaee"</pre>
```

```
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">
  <managed-bean>
     <managed-bean-name>holaBean</managed-bean-name>
     <managed-bean-class>ar.com.educacionit.holajsf.beans.HolaBean</managed-bean-class>
     <managed-bean-scope>
  </managed-bean>
  <navigation-rule>
    <from-view-id>index.xhtml</from-view-id>
    <navigation-case>
      <from-outcome>saludo</from-outcome>
      <to-view-id>/saludo.xhtml</to-view-id>
   </navigation-case>
  </navigation-rule>
</faces-config>
```