



Universidad de
Oviedo



REGLAS DE NAVEGACIÓN JSF + MANAGED BEANS

Enol García González
Universidad de Oviedo
18 de septiembre de 2023

CONTENIDOS

- 1 Introducción
- 2 Navegación
- 3 Ciclo de vida
- 4 Beans

Java Server Faces

Es un framework contruido sobre JSP para facilitar el desarrollo de interfaces de usuario. La primera versión 1.0 se lanzó en 2004.

Una de las ventajas de JSF es que se puede ampliar con librerías como:

- **RichFaces**, **ICEFaces** y **OpenFaces**, que agregan componentes visuales y soporte para AJAX.
- **jQuery4jsf**, que añade soporte para jQuery.
- **PrimeFaces**, que simplifica la configuración de JSF.

VERSIONES

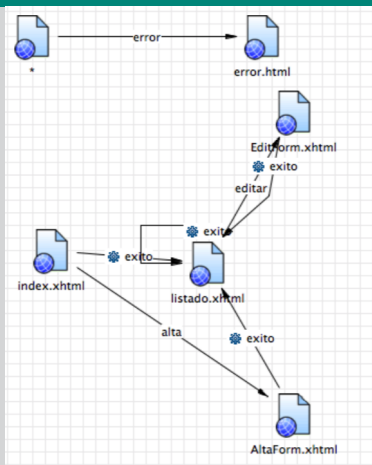
Las versiones más populares a día de hoy son:

- 2.2 (2013). Porque es la primera que soporta HTML5
- 2.3 (2017). Porque es la primera que soporta WebSockets y validación de Beans

¿QUÉ NOS VAMOS A ENCONTRAR?

- Un nuevo conjunto de etiquetas: `<h:...>` y `<f:...>`
- Un fichero de descriptor web que ya conocemos: `web.xml`
- Un nuevo fichero de recursos y reglas de navegación: `faces-config.xml`
- Managed Beans para proporcionar atributos y funciones a la presentación

NAVEGACIÓN



NAVEGACIÓN

```
<navigation-rule>  
  <from-view-id>/index.xhtml</from-view-id>  
  <navigation-case>  
    <from-outcome>alta</from-outcome>  
    <to-view-id>/alta.xhtml</to-view-id>  
  </navigation-case>  
</navigation-rule>
```

```
<navigation-rule>  
  <from-view-id>*</from-view-id>  
  <navigation-case>  
    <from-outcome>error</from-outcome>  
    <to-view-id>/error</to-view-id>  
  </navigation-case>  
</navigation-rule>
```


CICLO DE VIDA

- 1 **Restore view.** la vista que tiene el cliente. Se recrea en memoria.
- 2 **Apply Request Values.** Se extraen los valores de la vista a unas variables locales
- 3 **Process Validations.** Se validan los datos y en caso de que no sean válidos se recrea la vista con los mensajes de errores.
- 4 **Update Model Values.** Se actualizan los datos validados en el objeto Bean
- 5 **Invoke Application.** Se llama al método del controlador que responde a la acción
- 6 **Render Response.** El controlador le da la respuesta al cliente. Típicamente una nueva vista.

¿QUÉ ES UN BEAN?

Definición

Componente software reusable para proveer de atributos y comportamiento.

¿Y un Bean Gestionado?

Es un Bean del que no nos vamos a preocupar de gestionar. JSF lo gestionará y nosotros solo tendremos que utilizarlo para proveer a las vistas xhtml de atributos y métodos

MANAGED BEAN – EJEMPLO

@ManagedBean

```
public class HealthPlanHandler {  
    private String employeeName="Joe_User";  
    private String employeeId="a1234"  
    private String healthPlanName="CareFirst";  
  
    public String getEmployeeName() { /** ... */ }  
    public void setEmployeeName(String employeeName) { /** ... */ }  
    public String getEmployeeId() { /** ... */ }  
    public void setEmployeeId(String employeeId) { /** ... */ }  
    public String getHealthPlanName() { /** ... */ }  
    public void setHealthPlanName(String healthPlanName) { /** ... */ }  
    public void doRegistration() { /** ... */ }
```

MANAGED BEAN – EJEMPLO

```
[...]
<div align="center">
<h1 class="title">Health Plan Signup</h1>
<br/>
<h:form>
  Employee Name:
  <h:inputText value="#{healthPlanHandler.employeeName}"/><br/>
  Employee ID:
  <h:inputText value="#{healthPlanHandler.employeeId}"/><br/>
  Health Plan Name:
  <h:inputText value="#{healthPlanHandler.healthPlanName}"/><br/>
  <h:commandButton value="Register" action="#{healthPlanHandler.doRegistration
    }"/>
</h:form>
[...]
```

MANAGED BEAN – CONSIDERACIONES

- Todas las propiedades a las que se accede tienen que tener getters y setters.
- Los getters y setters se DEBEN llamar get... y set... con la primera letra mayúscula.
- Para las propiedades booleanas se puede usar tanto el get como el is.
- PROBLEMA: Los métodos getter se llaman múltiples veces durante el ciclo de vida de JSF.
- No incluir código pesado en los getters y setters. Por ejemplo, acceso a base de datos.

MANAGED BEAN – REGISTRO

Con la anotación `@ManagedBean` sobre la clase (JSF \geq 2.0) o con el siguiente código XML en el fichero `faces-config.xml`

```
<!-- faces-config.xml -->
```

```
<managed-bean>  
  <managed-bean-name>healthPlanHandler</managed-bean-name>  
  <managed-bean-class>com.tew.beans.HealthPlanHandler</managed-bean-class>  
</managed-bean>
```

MANAGED BEAN – ÁMBITOS

- Request
- Session
- Application
- View. Debe implementar serializable. Cambia con la vista, así que se suele utilizar junto con Ajax.
- Custom. El Bean se guarda en un map y se puede gestionar programáticamente.
- None. Para Beans referenciados por otros Beans.

En las versiones 1.x sólo existían Request, Session y Application.

Típicamente utilizaremos Session y Application con las anotaciones @SessionScope y @ApplicationScope