



Jakarta Faces Technology

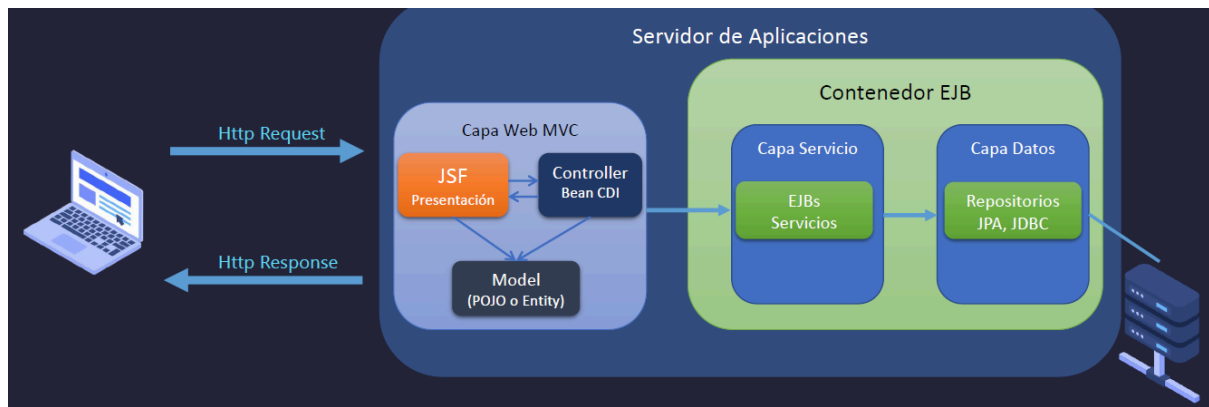
<https://jakarta.ee/learn/docs/jakartaee-tutorial/current/web/faces-intro/faces-intro.html>

Introducción

Es un framework para aplicaciones web que simplifica el desarrollo de interfaces de usuario en Jakarta EE mediante el uso de componentes gráficos y la vinculación del patrón MVC.

Permite la reutilización de componentes de la interfaz de usuario, hace uso de etiquetas JSF para vincular diferentes componentes gráficos así como validaciones a formularios directamente desde la página.

Las aplicaciones JSF son ejecutadas en un contenedor que permite realizar su despliegue y mantener con vida el sistema y sus componentes, contienen componentes de lógica de negocio (javaBeans), una biblioteca de etiquetas para componentes de interfaz gráfica, validadores, controladores de eventos y navegación entre otros.

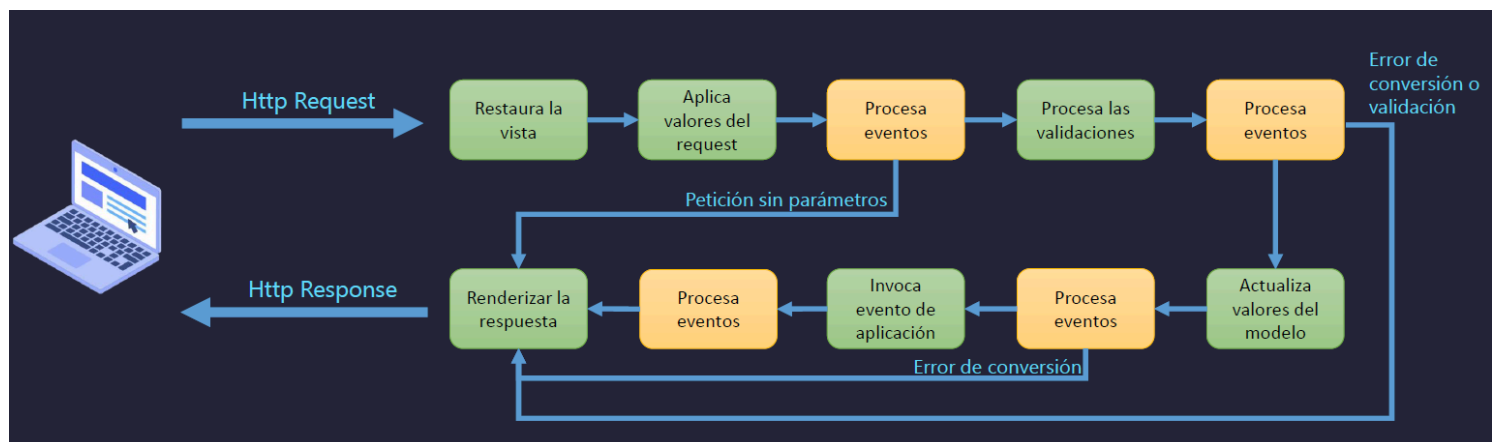


Características

- Modelo vista controlador MVC
- Componentes para interfaces de usuario
- Manejo eventos
- Validación de campos formulario
- Define reglas de navegación
- Soporte internacionalización y localización
- Integración CDI y EJB

Una **Facelet Composition Page** es una página creada usando el marco de trabajo **Facelets**, que es la tecnología predeterminada para definir vistas en aplicaciones web con **JavaServer Faces (JSF)**. Facelets es el motor de vistas de JSF que permite construir interfaces de usuario (UI) de manera modular, reutilizando componentes y plantillas de manera eficiente.

Ciclo de vida



El ciclo de vida estándar de JSF consta de **6 fases**, pero no siempre se ejecutan todas. Dependiendo de las acciones y la configuración, algunas fases pueden ser omitidas o completadas rápidamente.

1. Restaurar la vista (Restore View)
2. Aplicar los valores de la solicitud (Apply Request Values)
3. Procesar validaciones (Process Validations)
4. Actualizar los valores del modelo (Update Model Values)
5. Invocar la acción (Invoke Application)
6. Renderizar la respuesta (Render Response)

¿Qué es una Facelet Composition Page?

Una **Facelet Composition Page** es un archivo de vista con la extensión `.xhtml` que utiliza la capacidad de composición de Facelets para **reutilizar componentes comunes** entre varias páginas. La idea es que puedes crear una **plantilla base** con elementos comunes (como el encabezado, el pie de página, el menú de

navegación) y luego crear otras páginas que hereden o incluyan esos elementos comunes, lo que facilita la consistencia y el mantenimiento del diseño.

Conceptos clave:

1. **Plantillas (Templates):** Un archivo `.xhtml` que define una estructura general (como un diseño de página) que otras páginas pueden usar.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<h:head>
  <title><ui:insert name="title">Título por defecto</ui:insert></title>
</h:head>
<h:body>
  <h:panelGroup>
    <!-- Encabezado -->
    <ui:insert name="header">Encabezado por defecto</ui:insert>
    <!-- Contenido -->
    <ui:insert name="content">Contenido por defecto</ui:insert>
    <!-- Pie de página -->
    <ui:insert name="footer">Pie de página por defecto</ui:insert>
  </h:panelGroup>
</h:body>
</html>
```

2. **Páginas de composición:** Páginas que usan una plantilla base (como la del ejemplo anterior) y **rellenan** los huecos definidos en la plantilla (`<ui:insert>`).

Ejemplo de una Facelet Composition Page basada en la plantilla anterior:

```
<ui:composition template="/WEB-INF/templates/baseTemplate.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets">

    <ui:define name="title">Página de ejemplo</ui:define>
    <ui:define name="header">
        <h:outputText value="Este es mi encabezado personalizado" />
    </ui:define>
    <ui:define name="content">
        <h:outputText value="Aquí va el contenido de mi página" />
    </ui:define>
</ui:composition>
```

Beneficios de Facelet Composition:

- **Reutilización:** Permite reutilizar el diseño y los componentes comunes en varias páginas.
- **Mantenimiento más sencillo:** Cambios en la estructura general (como un cambio en el encabezado o pie de página) solo requieren cambios en un solo lugar (la plantilla).
- **Modularidad:** Se pueden dividir las páginas en pequeñas partes reutilizables, facilitando el desarrollo colaborativo y la claridad del código.

Componentes HTML más importantes en JSF

TIPO	ETIQUETA	DESCRIPCION
Agrupación	<h:form />	Define el formulario JSF - HTML
	<h:panelGrid />	Crea una tabla con los componentes incluidos en el panelGrid
	<h:panelGroup />	Permite agrupar dos o más componentes
Entrada	<h:inputText />	Campo de texto para entrada de datos
	<h:inputHidden />	Vincula un campo oculto en el formulario
	<h:inputSecret />	Vincula un campo de texto de datos de tipo contraseña (no muestra el contenido)
	<h:inputTextarea />	Componente área de texto (entrada de datos multilinea)
Salida	<h:outputText />	Etiqueta de texto
	<h:outputLabel />	Etiqueta de texto (se puede asociar a otro componente)
	<h:outputLink />	Permite crear un enlace o hipertexto
	<h:outputFormat />	Permite dar formato a un texto de salida
Errores y Mensajes	<h:message />	Permite imprimir un mensaje de error en la página
	<h:messages />	Permite imprimir varios mensajes de error en la página
Comandos – botones y enlaces	<h:commandButton />	Crea un botón para realizar acciones
	<h:commandLink />	Crea un enlace al que se le puede asociar a un botón
Selección	<h:selectOneMenu />	Lista de selección múltiple con única respuesta.
	<h:selectOneRadio />	Lista de radioButtons
	<h:selectManyListbox />	Lista de selección múltiple con múltiple respuesta
	<h:selectManyCheckbox />	Lista de checkbox
	<h:selectBooleanCheckbox />	Componente checkbox
	<h:selectManyMenu />	Componente de selección múltiple
Tablas de Datos	<h:dataTable />	Permite vincular una tabla de datos dinámicos, se obtienen de una lista de elementos.
Imágenes	<h:graphicImage />	Permite vincular imágenes

1. Texto y Salida

<h:outputText>: Muestra texto estático o dinámico.

<h:outputText value="Texto estático" />

<h:outputText value="#{bean.propiedad}" />

2. Entradas de Datos

<h:inputText>: Campo de entrada de texto.

```
<h:inputText value="#{bean.propiedad}" />
```

<h:inputSecret>: Campo de entrada de texto para contraseñas.

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

<h:inputTextarea>: Área de texto con varias líneas.

```
<h:inputTextarea value="#{bean.textoLargo}" rows="4" cols="50" />
```

3. Botones

<h:commandButton>: Botón que puede ejecutar una acción o enviar un formulario.

```
<h:commandButton value="Enviar" action="#{bean.metodo}" />
```

<h:commandLink>: Enlace que ejecuta una acción (similar a un botón).

```
<h:commandLink value="Haz clic aquí" action="#{bean.metodo}" />
```

4. Selección

<h:selectOneMenu>: Menú desplegable (solo una opción seleccionable).

```
<h:selectOneMenu value="#{bean.opcionSeleccionada}">
```

```
    <f:selectItem itemValue="opcion1" itemLabel="Opción 1" />
```

```
    <f:selectItem itemValue="opcion2" itemLabel="Opción 2" />
```

```
</h:selectOneMenu>
```

<h:selectManyCheckbox>: Conjunto de casillas de verificación (selección múltiple).

```
<h:selectManyCheckbox value="#{bean.opcionesSeleccionadas}">
    <f:selectItem itemValue="opcion1" itemLabel="Opción 1" />
    <f:selectItem itemValue="opcion2" itemLabel="Opción 2" />
</h:selectManyCheckbox>
```

<h:selectBooleanCheckbox>: Casilla de verificación de selección booleana.

```
<h:selectBooleanCheckbox value="#{bean.aceptaTerminos}" />
```

5. Etiquetas para Formularios

<h:outputLabel>: Etiqueta para un campo de formulario.

```
<h:outputLabel for="campoID" value="Etiqueta para el campo" />
```

6. Paneles de Agrupación

<h:panelGrid>: Crea un contenedor con disposición de tabla (filas y columnas).

```
<h:panelGrid columns="2">

    <h:outputLabel value="Nombre" />

    <h:inputText value="#{bean.nombre}" />

    <h:outputLabel value="Correo" />

    <h:inputText value="#{bean.correo}" />

</h:panelGrid>
```

<h:panelGroup>: Agrupa componentes sin generar un contenedor HTML específico.

```
<h:panelGroup>

    <h:outputText value="Texto agrupado" />

    <h:outputText value="Más texto" />

</h:panelGroup>
```

7. Mensajes y Errores

<h:messages>: Muestra todos los mensajes de validación y errores.

```
<h:messages globalOnly="true" />
```

<h:message>: Muestra un mensaje específico para un componente.

```
<h:message for="campoID" />
```

8. Archivos

<h:inputFile>: Campo de entrada para cargar archivos (JSF 2.2+).

```
<h:form enctype="multipart/form-data">

    <h:inputFile value="#{bean.archivo}" />

    <h:commandButton value="Subir"

action="#{bean.subirArchivo}" />

</h:form>
```

9. Enlaces y Navegación

<h:link>: Enlace estático que navega a otra página.

```
<h:link outcome="paginaDestino.xhtml" value="Ir a otra página"

/>
```

<h:outputLink>: Enlace que apunta a una URL externa.

```
<h:outputLink

value="http://www.google.com">Google</h:outputLink>
```

10. Scripts y Recursos

<h:outputScript>: Incluye un archivo de script (JavaScript).

```
<h:outputScript name="js/miscscript.js" />
```

<h:outputStylesheet>: Incluye un archivo CSS.

```
<h:outputStylesheet name="css/estilos.css" />
```

Componentes Core más importantes en JSF

JSF brinda una serie de etiquetas “core” usadas para manejo de eventos, atributos, validación, conversión de datos, administración de componentes entre otros.

ETIQUETA	DESCRIPCION
<f:view />	Permite crear una vista en la página
<f: subview />	Permite crear una sub vista en la página
<f: facet />	Permite añadir un elemento facet a un componente (para cabeceras y pies de página por ejemplo)
<f: attribute />	Añade un atributo a un componente mediante clave - valor
<f: param />	Añade un parámetro a un componente
<f: actionListener />	Añade una acción (listener) a un componente
<f: valueChangeListener />	Añade un nuevo valor al escuchador de un componente
<f: convertDateTime />	Agrega fecha a un componente
<f: convertNumber />	Añade conversión de un número a un componente
<f: validator />	Añade un validador a un componente
<f: validateDoubleRange />	Permite validar un rango de tipo double para datos ingresados en el componente
<f: validateLength />	Permite validar longitud del valores de ingresados en un componente
<f: validateLongRange />	Permite validar un rango de tipo long para valores ingresados en componentes
<f: loadBundle />	Carga origen de un elemento bundle
<f: selectitems />	Permite especificar los ítems que se van a cargar en un componente de selección (tomados por lo regular de un arreglo o listas)
<f: selectitem />	Permite especificar el ítem que se va a mostrar en un componente de selección (cuando se hace directamente)
<f: verbatim />	Añade una marca a una página jsf