



UNIVERSIDAD DEL QUINDÍO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Información general	
Actualizado por:	Carlos Andrés Flórez Villarraga
Duración estimada en minutos:	60
Docente:	Christian Andrés Candela y Einer Zapata
Guía no.	16
Nombre de la guía:	Navegación en JSF 2.3

Información de la Guía

OBJETIVO

Aprender a usar diferentes formas de navegar entre las páginas JSF.

CONCEPTOS BÁSICOS

Manejo de Eclipse, Java, archivos de propiedades, xml y Glassfish.

CONTEXTUALIZACIÓN TEÓRICA

JSF (Java Server Faces) es un *framework* de presentación, usado para la construcción de interfaces web (páginas) que permitan la interacción con el usuario. JSF es un *framework* basado en componentes y hace uso del patrón de diseño MVC (Modelo Vista Controlador).

Uno de los principales objetivos de JSF es separar la presentación (diseño de la página) de la lógica asociada a dicha presentación. Se puede pensar en JSF como en la librería *swing*, con la diferencia que *swing* lo usamos para crear las ventanas de nuestras aplicaciones de escritorio, mientras que JSF nos permite la creación de páginas web dentro de nuestras aplicaciones empresariales. Teniendo en cuenta esto, es importante decir que al igual que en *Spring* (*framework* para el desarrollo de aplicaciones) en JSF se pueden crear componentes propios con el fin de reutilizarlos posteriormente, o se puede hacer uso de componentes de terceros.

JSF ha sido incluido dentro de la especificación JEE como su capa de presentación. Siendo JSF un estándar, posee múltiples implementaciones. De igual forma gracias a su inclusión como estándar, JSF está en continua evaluación y actualización.

Dentro de JSF se tiene dos tipos de navegación, una navegación **implícita** y **definida por el usuario** (o programador). Cada una de las cuales se puede realizar: desde una página hacia otra, desde una página hacia un `ManagedBean`, o desde un `ManagedBean` hacia otra página.

Ver : <https://docs.oracle.com/javaee/7/tutorial/jsf-intro005.htm>

1. Navegación implícita.

1.1. Desde una página hacia otra.

La navegación entre páginas se puede realizar con diversos tipos de componentes JSF o HTML, y se puede cambiar la vista o redireccionar la página. El cambio de vista no afecta la URL de la página, por otra parte, en la

redirección se realiza una solicitud GET que cambia la URL de la página.

<h:link>: Permite generar hipervínculos, no requiere el uso de formulario. En los siguientes ejemplos se muestran dos diferentes formas de crear elementos que permiten navegar hacia la página index. El link genera una redirección.

```
<h:link outcome="index">link 1</h:link>
<h:link outcome="index" value="link 2" />
```

<a>: Al igual que el link, permite crear hipervínculos. En los siguientes ejemplos se muestran 3 formas de navegar a la página index. En las dos primeras, se requiere el uso del formulario porque se está haciendo uso del atributo `jsf:action`, el tercero es un uso normal de la etiqueta `<a>` y no requiere formulario. Por otra parte, entre los dos primeros elementos, el primero es un cambio de vista, mientras que el segundo genera una redirección.

```
<h:form>
  <a jsf:action="index" >index</a>
  <a jsf:action="index?faces-redirect=true" >index</a>
</h:form>

<a href="index.xhtml">index</a>
```

<h:outputLink>: Este componente funciona de forma similar a la etiqueta `<a>` que no ha sido enriquecida con el atributo `jsf:action`. Se recomienda su uso para crear vínculos hacia recursos y páginas externas o para enlaces internos en la misma página.

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

<h:commandLink>: Este componente funciona de forma similar a la etiqueta `<a>` cuando dicha etiqueta ha sido enriquecida con el uso de del atributo `jsf:action`. Este componente requiere el uso de formularios para su funcionamiento. En el primer ejemplo genera un cambio de vista que no afecta la URL de la página. El segundo ejemplo genera redirección hacia la página index afectando la URL.

```
<h:commandLink action="index" value="cambio de vista" />
<h:commandLink action="index?faces-redirect=true" value="redireccion" />
```

<input type="submit">: Este componente crea un botón que genera la acción de envío (`submit`) del formulario. Requiere estar incluido en un formulario para su funcionamiento y estar enriquecido con el atributo `jsf:action`. En el primer ejemplo genera un cambio de vista que no afecta la URL de la página. El segundo ejemplo genera redirección hacia la página index afectando la URL.

```
<h:form>
  <input type="submit" jsf:action="index" value="enviar" />
  <input type="submit" jsf:action="index?faces-redirect=true" value="enviar" />
</h:form>
```

<button>: Este componente crea un botón que genera la acción de envío (`submit`) del formulario. Requiere estar incluido en un formulario para su funcionamiento y estar enriquecido con el atributo `jsf:action`. En el

primer ejemplo general un cambio de vista que no afecta la URL de la página. El segundo ejemplo genera redirección hacia la página index afectando la URL.

```
<h:form>
    <button jsf:action="index">Enviar</button>
    <button jsf:action="index?faces-redirect=true">Enviar</button>
</h:form>
```

<h:commandButton>: Este componente crea un botón que genera la acción de envío (submit) del formulario. De forma similar a como lo hace la etiqueta `<input type="submit">` y al `<button>`. Requiere estar incluido en un formulario para su funcionamiento. En el primer ejemplo genera un cambio de vista que no afecta la URL de la página. El segundo ejemplo genera redirección hacia la página index afectando la URL.

```
<h:form>
    <h:commandButton action="index" value="enviar" />
    <h:commandButton action="index?faces-redirect=true" value="enviar" />
</h:form>
```

<h:button>: Este componente genera un botón, el cual permite realizar una redirección.

```
<h:button outcome="index" value="button" />
```

1.2. Desde un ManagedBean hacia una página

Cuando se invoca un método de un `ManagedBean` y dicho método retorna un `String`, JSF interpreta dicha cadena como la página a la que debe ir o la acción que debe realizar, similar al `action` de un botón o hipervínculo. En el primer ejemplo se genera un cambio de vista que no afecta la URL de la página. El segundo ejemplo genera redirección hacia la página index cambiando la URL.

```
@Named("bean")
@ApplicationScoped
public class Bean {
    ...
    public String ir(){
        return "index";
    }
}
```

```
@Named("bean")
@ApplicationScoped
public class Bean {
    ...
    public String ir(){
        return "index?faces-redirect=true";
    }
}
```

NOTA: si el método retorna `null`, o un `String` cuyo nombre no corresponda a una acción registrada o a una página existente en el proyecto JSF entonces permanecerá en la página actual.

2. Navegación definida por el programador

La navegación explícita en JSF se realiza a través de la configuración en el archivo `faces-config` que asocian constante de tipo `String` a la acción que permite navegar de una página a otra. Este tipo de navegación era ampliamente usada antes de la incorporación de la navegación implícita en JSF.

Ver:

<https://docs.oracle.com/javaee/7/tutorial/jsf-intro005.htm#BNAQL>

<https://docs.oracle.com/javaee/7/tutorial/jsf-configure009.htm#BNAXF>

PRECAUCIONES Y RECOMENDACIONES

Debe tener en cuenta que varios de los componentes usados para navegación entre páginas requieren estar incluidos dentro de un formulario.

ARTEFACTOS

Se requiere tener instalado el JDK y un IDE para el desarrollo de aplicaciones (Eclipse JEE en su última versión), un servidor de aplicaciones que cumpla con las especificaciones de JEE, para esta práctica Glassfish.

EVALUACIÓN O RESULTADO

Se espera que el alumno pueda emplear satisfactoriamente la navegación entre páginas JSF.

Procedimiento

1. Para el desarrollo de esta guía necesitará una base de datos en MySQL, un proyecto de tipo Maven – pom, el cual contenga un proyecto Maven con soporte para el uso de JPA, uno con soporte para EJB, uno para web y otro proyecto Maven configurado para la realización de pruebas. Y una conexión a la base de datos para ser usada en la generación de las tablas.
2. Cree una página que le permita al usuario registrar un nuevo producto. Tenga en cuenta lo siguiente para capturar la fecha límite del producto:

```
<h:inputText id="productoFecha" value="#{productoBean.fechaLimite}">
    <f:convertDateTime pattern="yyyy-MM-dd"></f:convertDateTime>
</h:inputText>
```

3. Adicione a su página un botón que invoque un método de un `ManagedBean` que permita registrar un producto, si el proceso es exitoso el usuario debe ser redireccionado a una página de éxito donde se muestren todos sus datos del producto registrado. En caso de que se presente algún error, el usuario deberá permanecer en la página de creación del producto y mostrar un mensaje que indique la causa del error.
4. Ejecute la aplicación en el servidor de aplicaciones para verificar su correcto funcionamiento.
5. Cree una página de búsqueda de productos que solicite al usuario uno o más parámetros del producto.
6. Adicione a la página un botón que invoque un método del `ManagedBean` el cual debe buscar el producto por los parámetros dados, en caso de encontrarlo deberá redireccionar a otra página que muestre los datos del producto encontrado. En caso de que no encuentre nada, deberá permanecer en la página de búsqueda



Guía de laboratorio
Área de Programación y Algoritmia



e indicar al usuario por medio de un mensaje que no se encontró.

7. Ejecute la aplicación en el servidor de aplicaciones para verificar su correcto funcionamiento.
8. Cree una página de bienvenida que contenga un menú que permita al usuario ir a la página de registro de producto o búsqueda, así como las demás opciones (lista de compras, lista de favoritos, edición de perfil, etc).
9. Ejecute la aplicación en el servidor de aplicaciones para verificar su correcto funcionamiento.