



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

Información general	
Actualizada por:	Carlos Andrés Flórez Villarraga
Duración estimada en minutos:	60
Docente:	Christian Andrés Candela y Einer Zapata
Guía no.	14
Nombre de la guía:	Configuración de JSF (Java Server Faces) 2.3

Información de la Guía

OBJETIVO

Aprender crear un proyecto web, y configurarlo para usar JSF 2.3.

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 web) que permitan la interacción con el usuario. JSF es un framework basado en componentes, 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 (o JavaFX) 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, 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.

PRECAUCIONES Y RECOMENDACIONES

Tenga en cuenta que las páginas deben ser guardadas con extensión xhtml. De igual forma debe recordar que cada etiqueta debe tener un cierre.

ARTEFACTOS

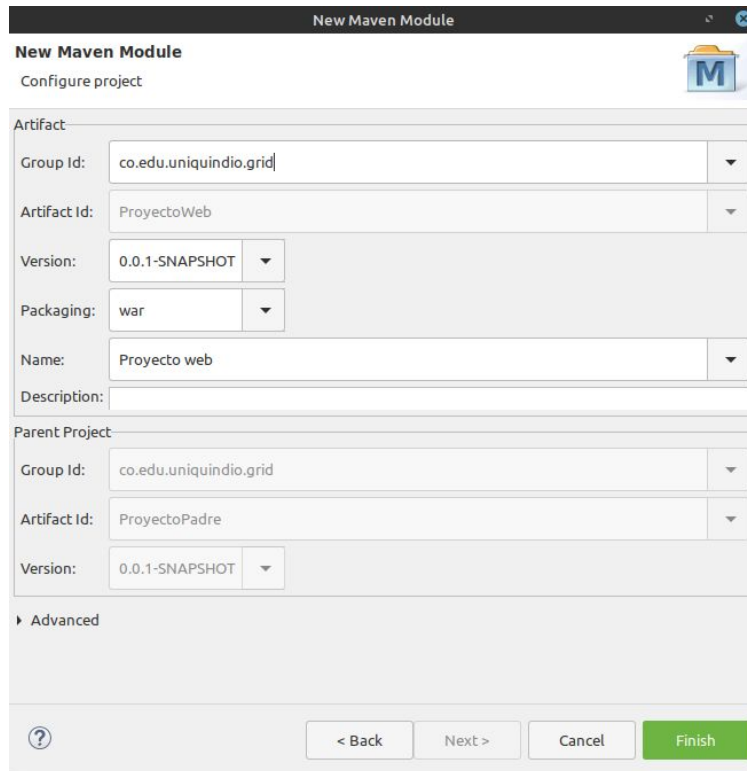
Se requiere tener instalado el JDK y un IDE para el desarrollo de aplicaciones (Eclipse JEE en su última versión), un servidor de aplicaciones que cumpla con las especificaciones de JEE, para esta práctica Glassfish.

EVALUACIÓN O RESULTADO

Se espera que el alumno pueda crear y desplegar exitosamente un proyecto con soporte de JSF 2.3.

Procedimiento

1. Para el desarrollo de esta guía necesitará una base de datos en MySQL, un proyecto de tipo Maven – POM, el cual debe contener un proyecto Maven con soporte para el uso de JPA, uno más con soporte para EJB 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. Adicione un nuevo módulo a su Proyecto padre. El nuevo módulo Maven será el Proyecto web. Para ello debe dar clic derecho sobre el Proyecto padre y en la opción maven seleccionar new Maven Module Project. Para este módulo se creará la configuración por defecto. Seleccione como tipo de empaquetamiento war.



3. Adicione las siguientes dependencias a su Proyecto web.

```
<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>primefaces</artifactId>
  <version>6.2</version>
</dependency>
<dependency>
  <groupId>javax</groupId>
  <artifactId>javaee-api</artifactId>
  <version>8.0</version>
  <scope>provided</scope>
</dependency>
```

```
<dependency>
  <groupId>javax.faces</groupId>
  <artifactId>javax.faces-api</artifactId>
  <version>2.3</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>com.sun.faces</groupId>
  <artifactId>jsf-impl</artifactId>
  <version>2.2.17</version>
  <scope>compile</scope>
</dependency>
```

4. Ahora modifique el archivo POM con la información perteneciente a su proyecto. También debe añadir como dependencias el Proyecto de persistencia y el de negocio. Tenga en cuenta que el *scope* debe ser *provided*.
5. A su Proyecto EAR adicione como dependencia el Proyecto WEB. Así:

```
<dependency>
  <groupId>co.edu.uniquindio.grid</groupId>
  <artifactId>ProyectoWeb</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <type>war</type>
</dependency>
```

6. De clic derecho sobre el Proyecto padre y acceda a la opción Maven – Update project.
7. Verifique que exista la carpeta `src/main/webapp/WEB-INF`. También lo puede encontrar en:



8. Verifique que exista el archivo `src/main/webapp/WEB-INF/web.xml`. Y que se vea así:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>ProyectoWeb</display-name>
  <welcome-file-list>
    <welcome-file>index.xhtml</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
```

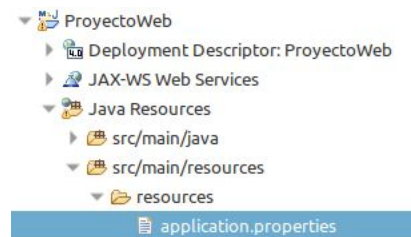
```
</servlet>
<servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
</web-app>
```

9. Verifique que exista el archivo `src/main/webapp/WEB-INF/faces-config.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_3.xsd"
    version="2.3">
    <application>

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

10. Cree un archivo de propiedades con el nombre de `application.properties` en el paquete `resources`



11. Adicione la siguiente llave a su archivo de propiedades.

```
saludo=Hola Mundo JEE
```

12. En su archivo `faces-config.xml` dentro de las etiquetas `application` adicione lo siguiente:

```
<resource-bundle>
    <base-name>resources.application</base-name>
    <var>mensajes</var>
</resource-bundle>
```

13. Diríjase a la página web de nombre `index.xhtml` en la carpeta `src/main/webapp` en donde encontrará el siguiente código.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
```

```
<title>Insert title here</title>
</head>
<body>
  <h1>#{holaMundoBean.mensaje}</h1>
</body>
</html>
```

Es importante recordar que en xhtml y JSF cada etiqueta debe tener un cierre.

Note que la letra inicial en `#{holaMundoBean.mensaje}` está en minúscula, a diferencia del nombre de la clase `HolaMundoBean`, donde la letra inicial aparece en mayúscula. Esto se debe a que en la página se hace referencia a una instancia de la clase `HolaMundoBean` no a la clase en sí misma.

14. Si no existe la página `index.xhtml` créela en dicha carpeta, para ello, de clic derecho al proyecto, luego en New y HTML File, asígnele el nombre `index.xhtml` (con extensión) y Template HTML 5. Además cree un Managed Bean que funcionará como el controlador de dicha página así:

```
package co.edu.uniquindio.prueba.beans;

import javax.enterprise.context.ApplicationScoped;
import javax.inject.Named;

@Named("holaMundoBean")
@ApplicationScoped
public class HolaMundoBean {

    private String mensaje;

    public String getMensaje() {
        return "Hola Mundo";
    }

    public void setMensaje(String mensaje) {
        this.mensaje = mensaje;
    }

}
```

15. De clic derecho sobre el Proyecto EAR y seleccione la opción Run – Run on Server. Luego abra un navegador e intente acceder a la página que acaba de crear usando la siguiente URL <http://localhost:8080/nombreproyecto/index.xhtml>.
16. Como podrá ver, hemos accedido al método `getMensaje` de forma directa desde nuestra página web.
17. En la página `index.xhtml`, reemplace el texto `#{holaMundoBean.mensaje}` por `#{mensajes.saludo}`.
18. Despliegue la aplicación nuevamente en el servidor para observar los resultados.
19. Modifique el método `getMensaje` de tal forma que su retorno haga referencia al texto almacenado en el archivo de propiedades `resources.mensajes` bajo la llave `saludo`.