

Capítulo 4. Mini Tutorial

El objetivo de este capítulo es mostrar paso a paso el desarrollo de una pequeña aplicación haciendo uso de JavaServer Faces, para que a través de esto se tenga una idea más clara acerca de cómo iniciar la implementación de este framework.

La aplicación a desarrollar consistirá de dos páginas: la primera para ingresar y capturar el nombre del usuario, y la segunda para desplegar un mensaje de bienvenida a dicho usuario.

Debido a la simplicidad de la aplicación, ésta incluirá sólo algunas funcionalidades de JavaServer Faces.

Para comenzar con este mini tutorial, se da por hecho que se ha instalado correctamente Tomcat, así como JavaServer Faces 1.1. A continuación se presentarán los pasos necesarios para el desarrollo mencionado.

1. Se crea una carpeta en Tomcat, dentro de *webapps*, quedando su estructura como se muestra en la Figura 4.1.

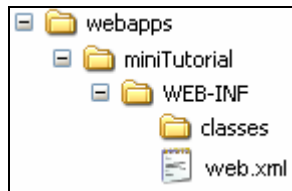


Figura 4.1 Estructura de Directorios

2. El archivo *web.xml* debe modificarse de tal manera que cada página con terminación *.faces* sea procesada por el servlet de JSF. El archivo *web.xml* se muestra en la Figura 4.2.

```

<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC
  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <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>*.faces</url-pattern>
  </servlet-mapping>
</web-app>

```

Figura 4.2 Contenido del Archivo *web.xml*

3. Se crea el archivo *faces-config.xml* dentro de la carpeta *web-inf*; este archivo jugará el papel de archivo de configuración para JSF. Inicialmente su contenido será como se muestra en la Figura 4.3; posteriormente se agregarán *beans* y reglas de navegación. Ahora la estructura de directorios queda como se muestra en la Figura 4.4.

```

<?xml version="1.0"?>

<!DOCTYPE faces-config PUBLIC
  "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.0//EN"
  "http://java.sun.com/dtd/web-facesconfig_1_0.dtd">

<faces-config>

</faces-config>

```

Figura 4.3 Contenido del Archivo *faces-config.xml*

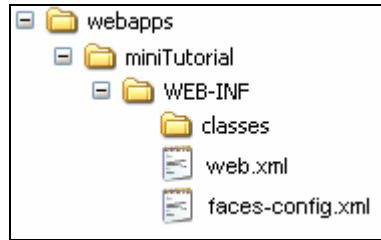


Figura 4.4 Estructura de Directorios Actualizada

4. Se crea el *bean* que se asociará al usuario para guardar su información; debido a que solamente se está pidiendo el nombre, el *bean* tendrá esta única propiedad, así como los correspondientes métodos *get* y *set*. La estructura de la clase que representa al *bean* descrito se muestra en la Figura 4.5, y la estructura de directorios hasta el momento se presenta en la Figura 4.6.

```

1 package mini;
2
3 public class Usuario
4 {
5     private String nombre;
6
7     public void setNombre(String nombre)
8     {
9         this.nombre = nombre;
10    }
11
12    public String getNombre()
13    {
14        return this.nombre;
15    }
16 }

```

Figura 4.5 Bean Usuario

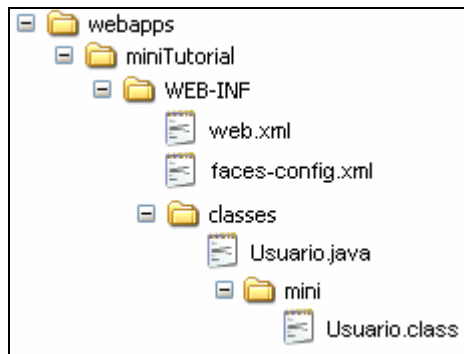


Figura 4.6 Estructura de Directorios Actualizada

5. Se registra el *bean* dentro del archivo de configuración de JSF (*faces-config.xml*) para poder hacer uso de él más adelante. El contenido del archivo *faces-config.xml* se muestra en la Figura 4.7.

```
<?xml version="1.0"?>

<!DOCTYPE faces-config PUBLIC
  "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.0//EN"
  "http://java.sun.com/dtd/web-facesconfig_1_0.dtd">

<faces-config>

  <managed-bean>
    <managed-bean-name>usuario</managed-bean-name>
    <managed-bean-class>mini.Usuario</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>

</faces-config>
```

Figura 4.7 Contenido del Archivo *faces-config.xml*

6. Se crea el paquete de mensajes para la aplicación y se guarda dentro de la carpeta “mini”. Los mensajes a utilizar en la aplicación se pueden observar en el contenido del archivo *.properties* que se muestra en la Figura 4.8.

```
tituloVentana = Mini Tutorial de JavaServer Faces
tituloPaginaUno = Págiano Uno: solicitar nombre al usuario
tituloPaginaDos = Págiano Dos: desplegar mensaje de bienvenida

ingresarNombre = Ingrese su nombre:
bienvenida = Bienvenido(a) a este Mini Tutorial de JavaServer Faces

aceptar = Aceptar
```

Figura 4.8 Archivo *mensajes.properties* a Manera de Paquete de Mensajes

7. Se crea la primer página; ésta deberá pedir el nombre al usuario, como se muestra en la Figura 4.9.

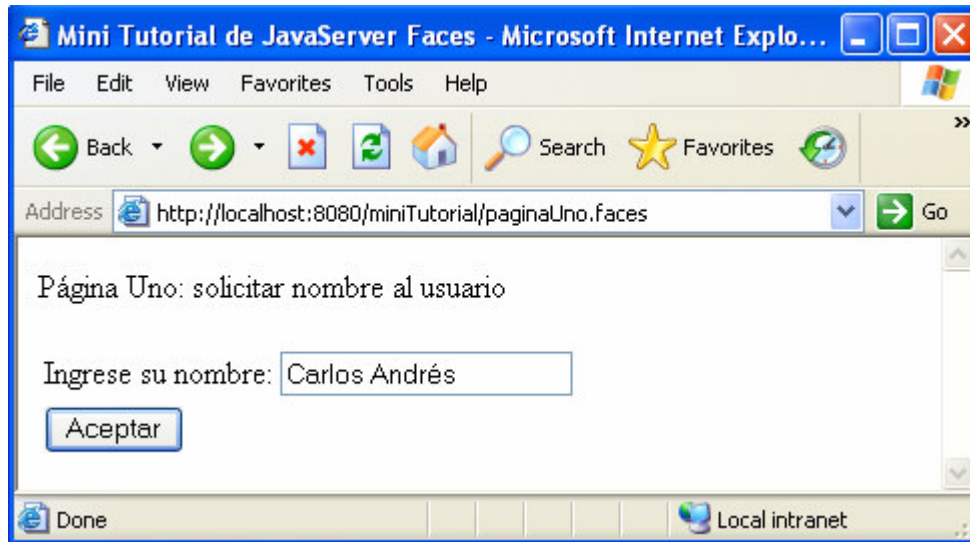


Figura 4.9 Diseño de la Página Uno

El código para la Página Uno se presenta en la Figura 4.10.

```

1 <html>
2 <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %> } a)
3 <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %> } b)
4 <f:view>
5 <head>
6 <f:loadBundle basename="mini.mensajes" var="msjs"/> } b)
7 <title>
8 <h:outputText value="#{msjs.tituloVentana}"/> } c)
9 </title>
10 </head>
11
12 <body>
13 <h:outputText value="#{msjs.tituloPaginaUno}" />
14 <h:form>
15 <h:panelGrid columns="3">
16 <h:outputText value="#{msjs.ingresarNombre}"/>
17 <h:inputText id="nombre" value="#{usuario.nombre}" required="true" /> } d) } e)
18 <h:message for="nombre" />
19
20 <h:commandButton value="#{msjs.aceptar}" action="exito" /> } f)
21 </h:panelGrid>
22 </h:form>
23 </body>
24 </f:view>
25 </html>

```

Figura 4.10 Código de la Página Uno

- Los puntos más importantes del código de la Página Uno se destacan en la Figura 4.10 y se comentan a continuación:
- a) se importan y se asignan a variables las librerías de JavaServer Faces;
 - b) se importa y se asigna a una variable el paquete de mensajes;
 - c) se hace uso del paquete de mensajes a través de la variable asignada;
 - d) se mapea el valor del componente `h:inputText` con la propiedad `nombre` del *bean* llamado `usuario` (previamente registrado); es decir, para esta página, el valor que se ingrese en el campo del nombre, será guardado en la variable `nombre` del bean `Usuario`;
 - e) se declara como obligatorio el campo del nombre;
 - f) se utiliza una navegación estática.
8. Se crea la segunda página; ésta desplegará la bienvenida al usuario, como se muestra en la Figura 4.11.

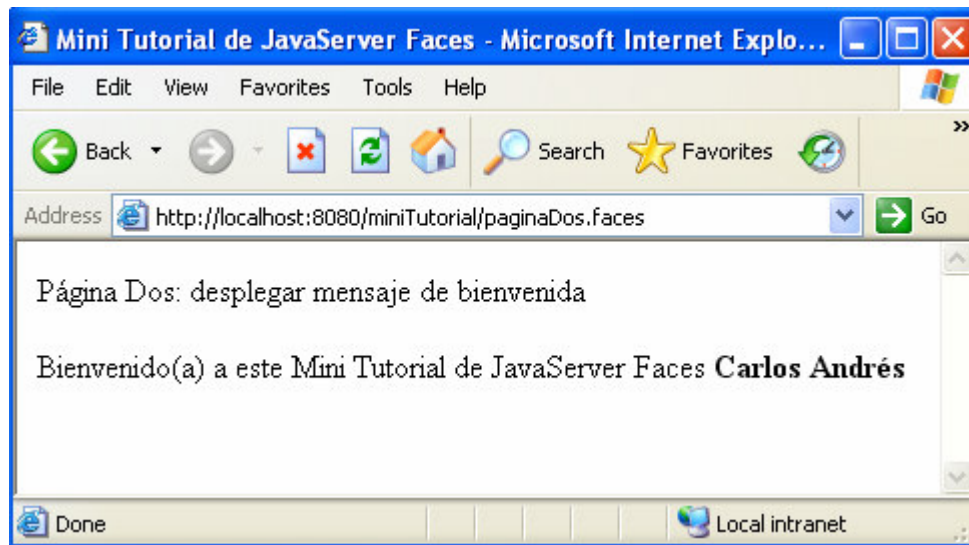


Figura 4.11 Diseño de la Página Dos

El código de la Página Dos se presenta en la Figura 4.12.

```

1 <html>
2   <%= taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
3   <%= taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
4   <f:view>
5     <head>
6       <f:loadBundle basename="mini.mensajes" var="msjs" />
7       <title>
8         <h:outputText value="#{msjs.tituloVentana}" />
9       </title>
10    </head>
11
12    <body>
13      <h:outputText value="#{msjs.tituloPaginaDos}" />
14      <h:form>
15        <h:outputText value="#{msjs.bienvenida}" />
16        <b><h:outputText value="#{usuario.nombre}" /></b>
17      </h:form>
18    </body>
19  </f:view>
20 </html>

```

Figura 4.12 Código de la Página Dos

Los puntos más importantes del código de la Página Dos se destacan en la Figura 4.12 y se comentan a continuación:

- a) Se mapea el valor del componente `h:outputText` con la propiedad `nombre` del *bean* llamado `usuario` (previamente registrado); es decir, para esta página, se tomará el valor almacenado en la variable `nombre` del *bean* `Usuario` y se desplegará.
9. Se diseña e implementa la navegación de la aplicación en el archivo `faces-config.xml`. Ésta consistirá en ir de la Página Uno a la Página Dos cuando se haga clic sobre el botón “Aceptar” de la primera. El archivo de configuración de JSF queda como se muestra en la Figura 4.13.

```

<?xml version="1.0"?>

<!DOCTYPE faces-config PUBLIC
    "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.0//EN"
    "http://java.sun.com/dtd/web-facesconfig_1_0.dtd">

<faces-config>

    <managed-bean>
        <managed-bean-name>usuario</managed-bean-name>
        <managed-bean-class>mini.Usuario</managed-bean-class>
        <managed-bean-scope>session</managed-bean-scope>
    </managed-bean>

    <navigation-rule>
        <from-view-id>/paginaUno.jsp</from-view-id>
        <navigation-case>
            <from-outcome>exito</from-outcome>
            <to-view-id>/paginaDos.jsp</to-view-id>
        </navigation-case>
    </navigation-rule>

</faces-config>

```

Figura 4.13 Contenido del Archivo *faces-config.xml*

10. La estructura de directorios final de la aplicación se presenta en la Figura 4.14.

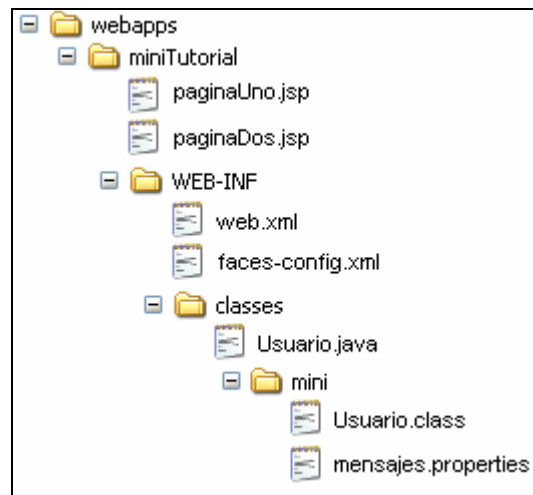


Figura 4.14 Estructura de Directorios Final.

11. Se abre un explorador y se ingresa la dirección <http://localhost:8080/miniTutorial/paginaUno.faces> para comenzar la aplicación.