	**Escuela Superior de Empresa, Ingeniería y Tecnología.** **Ingeniería Informática**
CICLO I	**GUIA DE LABORATORIO #8** **Ingeniería de Software** **JSP y JDBC**

I. OBJETIVOS

Que el estudiante:

- Pueda crear e implementar paginas JSP con Netbeans.
- Cree y manipule sesiones con JSP.

II. INTRODUCCION

JavaServer Pages (JSP) (http://java.sun.com/jsp) es una tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web. Los archivos JSP combinan HTML con etiquetas especiales y fragmentos de código Java.

Una página JSP es un tipo especial de servlet que puede poseer código HTML y que contiene ciertas etiquetas especiales (acciones) o fragmentos de código (scriptlets) para incluir contenido dinámico en dichas páginas. El código HTML se añade tal cual a la respuesta, mientras que los elementos dinámicos se evalúan con cada petición.

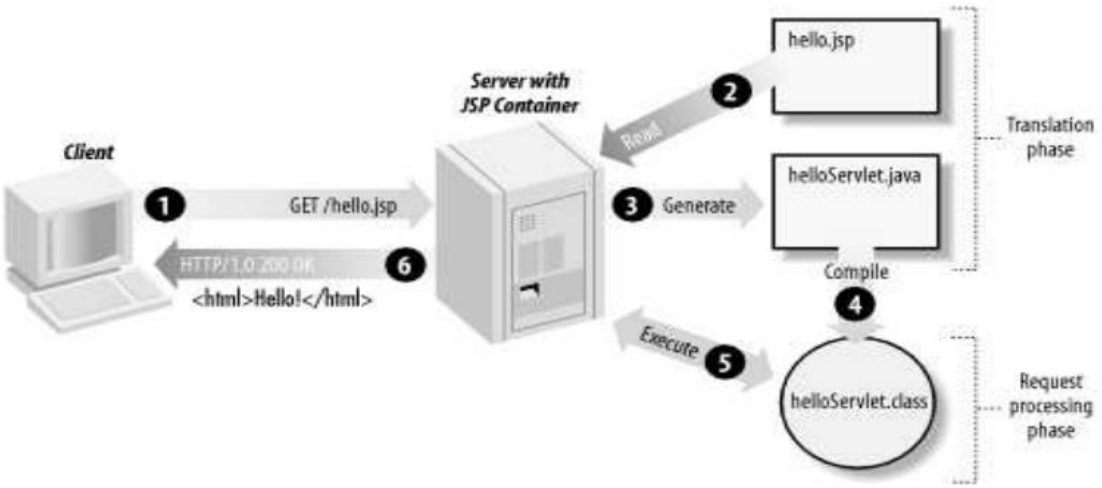


Figura 1: Petición de una página JSP

Cuando un cliente pide esa página al servidor de aplicaciones, se traduce a un fichero fuente de Java que implementa un Servlet, se compila y se ejecuta para devolver el resultado de la petición.

Una vez compilado el servlet, las peticiones posteriores se reducen a ejecutar dicho servlet en lugar de realizar el proceso una y otra vez. Esta es la razón por la que la primera petición tarda mucho

más en responderse que las siguientes.
Dado que una JSP no es más que un servlet, hay que recordar que todo el código se ejecuta en el servidor, las respuestas son puramente páginas HTML.

Código Java
Podemos insertar código Java dentro de JSP de tres formas: Expresiones, scriptlets y declaraciones.

Expresiones: Son fragmentos de código Java, con la forma **<%= expresión %>** que se evalúan y se muestran en la salida del navegador. En general, dentro de una expresión podemos usar cualquier cosa que usaríamos dentro de un System.out.print(expr);

Scriptlets: Son fragmentos de código Java con la forma **<% código %>**, en general, podemos insertar cualquier código que pudiéramos usar dentro de una función Java. Para acceder a la salida del navegador, usamos el objeto implícito out.

Declaraciones: Contienen declaraciones de variables o métodos, con la forma **<%! declaración %>.** Estas variables o métodos serán accesibles desde cualquier lugar de la página JSP. Hay que tener en cuenta que el servidor transforma la página JSP en un servlet, y éste es usado por múltiples peticiones, lo que provoca que las variables conserven su valor entre sucesivas ejecuciones.

Directivas: Las directivas son elementos que proporcionan información al motor JSP, e influirán en la estructura del servlet generado. Hay tres tipos de directivas: page, taglib e include.

page: Se indica con la forma <%@ page atributo="valor">. Tiene diversos usos, entre los cuales destacaremos:

- Importar clases. Importar código, de la misma forma que se realiza en un programa en Java, se indica con el atributo import.
  Ejemplo:

  ```
  <%@page import="java.io.*, miPackage.miClase"%>
  ```

- Indicar si la página tendrá acceso a la sesión. Se especifica con el atributo session.
  Ejemplo:

  ```
  <%@page session="true" %>
  ```

- Gestión de errores. Permite redireccionar a una página cuando se produzca un error, se indica con los atributos errorPage e isErrorPage.

  Ejemplos:

  ```
  <%@page errorPage="error.jsp">
  [...]
  <%@page isErrorPage="true">
  <html>
      <body>
          Error, contacte con el administrador [...]
      </body>
  </html>
  ```

Include: Permite incluir un archivo en el lugar donde se especifique, al contrario que con la acción <jsp:include> que veremos más adelante, la directiva include simplemente copia el contenido del

archivo byte a byte, siendo el resultado similar a si copiáramos el texto del archivo incluido y lo pegáramos en el JSP.

taglib: Se emplea para indicar que se van a emplear librerías de etiquetas. Se verá con más detalle en la siguiente sesión de clases.
Ejemplo:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

Acciones
Las acciones tienen la forma <jsp:accion [parámetros]/>, y tienen diversos usos, entre los que destacan la inclusión de páginas y transferencia de control.

Inclusión de páginas
Se realiza con la acción **<jsp:include page="pagina.jsp">.** Incluye la salida de otra página JSP en la actual, al contrario que con la directiva <%@include file="fichero.ext"%>, la página incluida se ejecuta y su salida se inserta en la página que la incluye, con la directiva se incluye el contenido del archivo (no su salida) y se ejecuta conjuntamente con la página principal.

La página incluida tiene acceso a los parámetros enviados a la principal, y podemos enviarle nuevos parámetros con la sub etiqueta <jsp:param name="nombre" value="valor"/>.

Transferencia de control
Se realiza con la acción <jsp:forward page="pagina.jsp"/>. La petición es redirigida a otra página, y la salida de la actual se descarta. Al igual que con la inclusión, la página a la que se redirige tiene acceso a los parámetros pasados a la actual, y es posible el envío de nuevos parámetros.
Ejemplo:

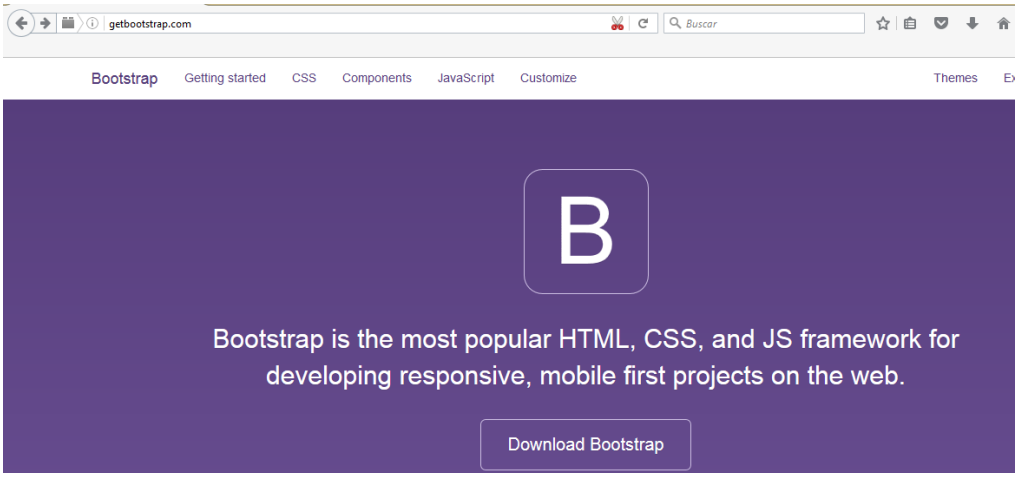
```
<jsp:forward page="principal.jsp">
    <jsp:param name="titulo" value="Principal"/>
</jsp:forward>
```

III. PROCEDIMIENTO

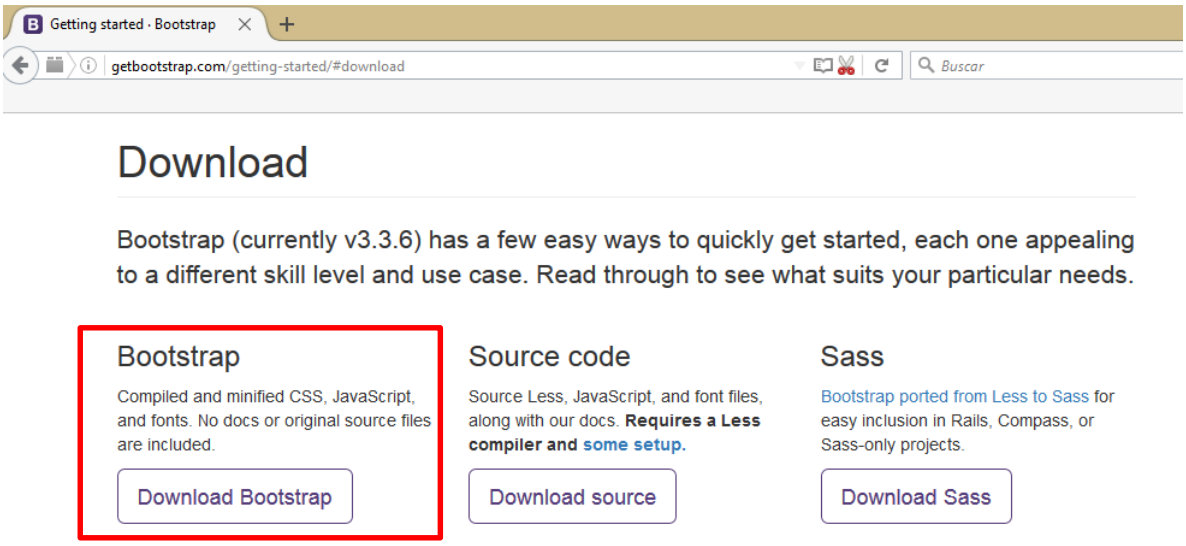
Creación de Pagina JSP y Manejo de Elementos Java.

1. Crear un nuevo proyecto web con el nombre de Guia8POO1.

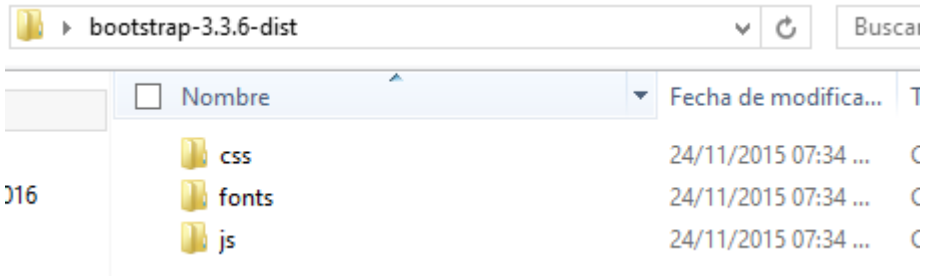
Para el desarrollo de esta guía haremos uso del framework Bootstrap que nos permitirá manejar de forma responsiva el diseño visual de nuestra aplicación. Lo primero que hay que hacer es descargar los archivos correspondientes de la página oficial: http://getbootstrap.com/



a. Damos click en el botón "**Download Bootstrap**" lo que nos redirigirá a una nueva página con 3 opciones de descarga, de las cuales seleccionaremos la primera: "**Download Bootstrap**".

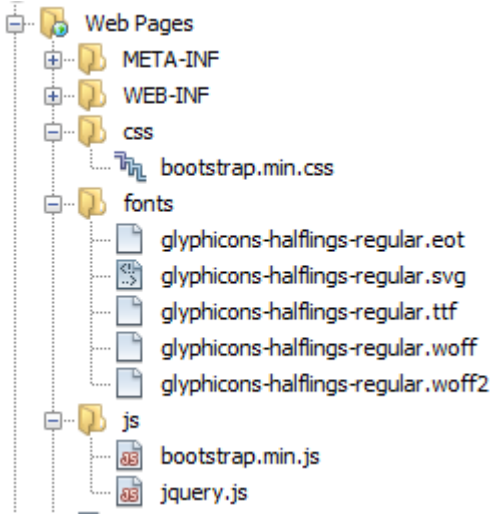


b. Luego de la descarga, descomprimimos la carpeta correspondiente, que contiene la siguiente estructura:



De los archivos disponibles, utilizaremos los siguientes: ***bootstrap.min.css*** (ubicado dentro de la carpeta css), ***bootstrap.min.js*** (ubicado dentro de la carpeta js) y todos los archivos de la carpeta fonts. Adicionalmente, utilizaremos el framework jquery para disponer de ciertas funcionalidades dinámicas de la parte visual, el cual se encuentra disponible desde el siguiente enlace: https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.js

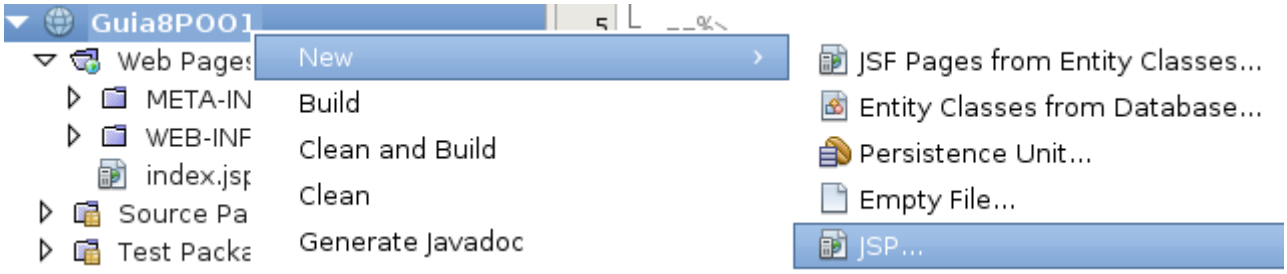
c. En el proyecto crearemos tres nuevas carpetas (Folder) dentro de **Web Pages**, una llamada **js**, **css**, y **fonts** en donde colocaremos los archivos mencionados, así.



Nota: También es posible incluir el framework de Bootstrap y todos sus recursos en nuestra aplicación desde un CDN (Red de Entrega de Contenidos), esta es una alternativa viable cuando se cuenta con una buena conexión a internet y no se desea incluir de manera local los archivos correspondientes. Lo anterior puede lograrse incluyendo dentro de las etiquetas **head** de nuestra página referencias a los archivos de forma remota en lugar de forma local, así:

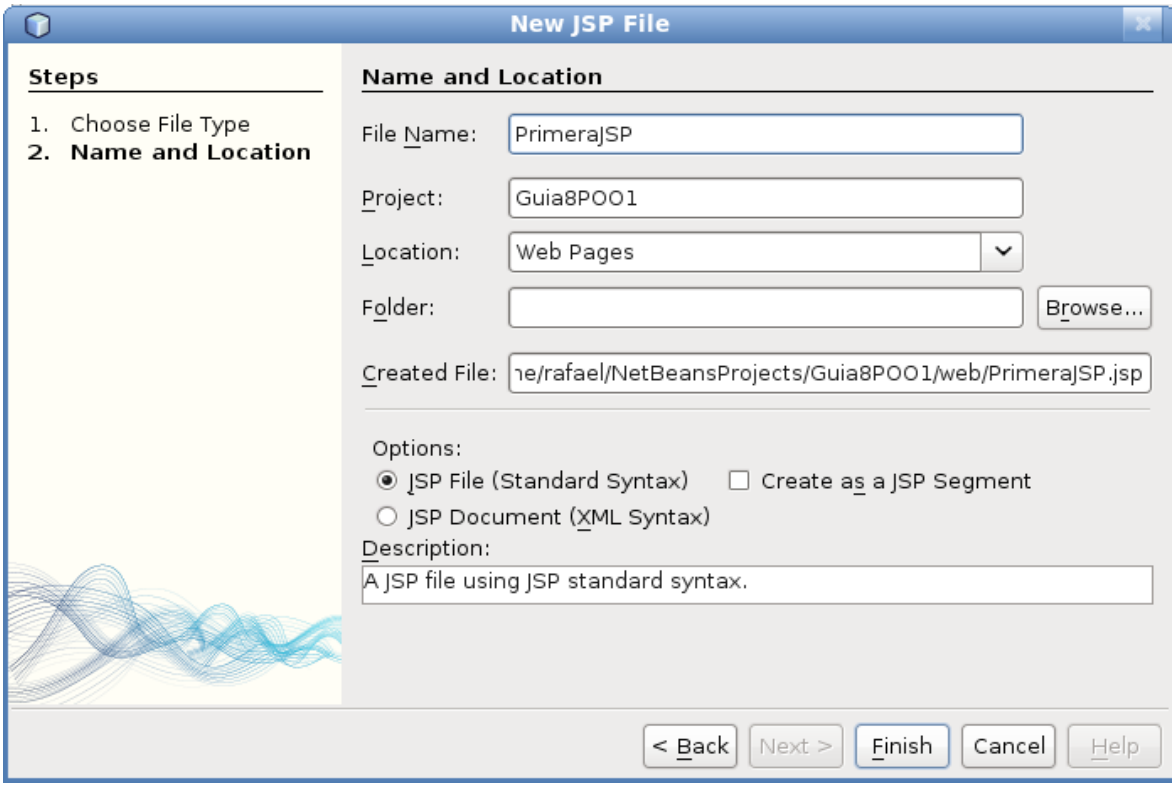
```
<head>
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css"
rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
</head>
```

2. Crear una nueva página JSP (ver la siguiente figura).



3. En el nombre del archivo ingresar **"PrimeraJSP"**, no agregar extensión ya que esta se

agrega por defecto.

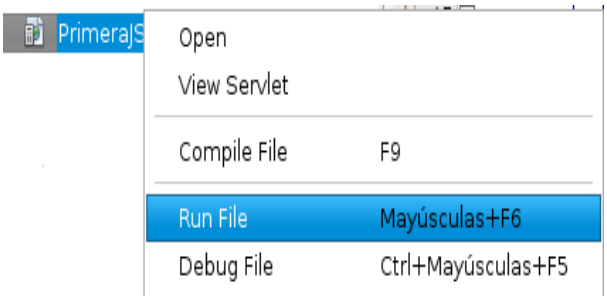


4. Digitar el siguiente código en la página creada.

```
<%@page import="java.util.*"%>
<%! String titulo = "Mi primer ejemplo con JSP";
   String cadena = "Primer ejemplo";
%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <title><%= titulo%></title>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="css/bootstrap.min.css">
   </head>
   <body>
      <div class="container">
         <div class="row">
            <h3><%= titulo%></h3>
         </div>
         <div class="panel panel-primary">
            <div class="panel-heading"><%= cadena%></div>
            <div class="panel-body">
               <%
                  out.println(new Date());
               %>
            </div>
         </div>
```

```
    </div>
  </body>
</html>
```

5. Correr la aplicación para ver el resultado (ver la figura siguiente)

Correr Pagina	**Resultado**
	

6. Ahora modificaremos un poco el código para poder utilizar Elementos de JSP, como son las expresiones, scriptlets y declaraciones, el código de su página debe quedar de la siguiente manera.

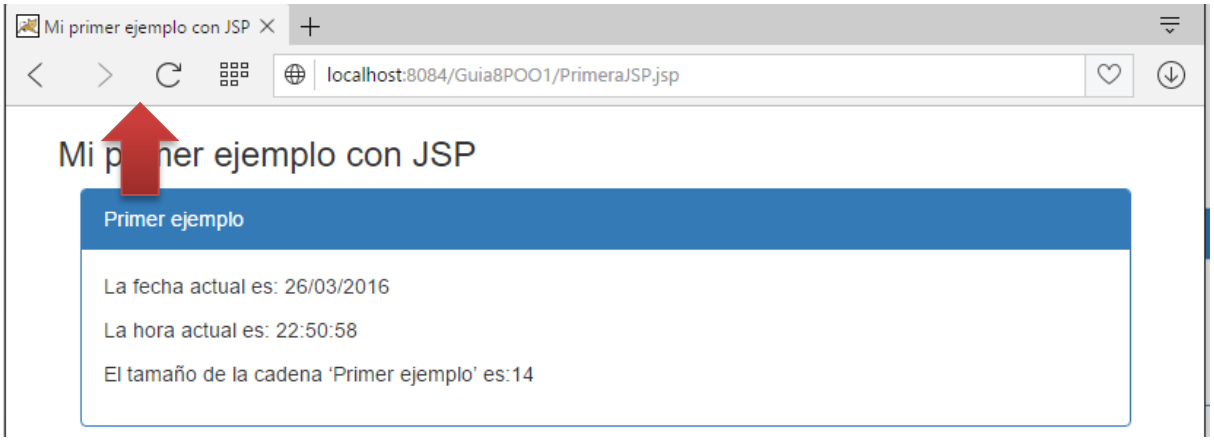
Codigo

```
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page import="java.util.*"%>
<%! String titulo = "Mi primer ejemplo con JSP";
   String cadena = "Primer ejemplo";
%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
      <title><%= titulo%></title>
          <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="css/bootstrap.min.css">
   </head>
<body>
      <%
         //Obteniendo la fecha actual
         Date fechaActual = new Date();
         //Formateando la fecha
         DateFormat formatoHora = new SimpleDateFormat("HH:mm:ss");
         DateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
      %>
       <div class="container">
         <div class="row">
            <h3><%= titulo%></h3>
```

```
        </div>
        <div class="panel panel-primary">
          <div class="panel-heading"><%= cadena%></div>
          <div class="panel-body">
  <p><%= "La fecha actual es: " + formatoFecha.format(fechaActual)%></p>
  <p><%= "La hora actual es: " + formatoHora.format(fechaActual)%></p>
  <p><%= " El tamaño de la cadena ‘"+ cadena + "’ es:" + cadena.length()%></p>
          </div>
        </div>
      </div>
  </body>
</html>
```

El resultado obtenido sería el siguiente:

Tip: No es necesario que usted vuelva a seleccionar Run File cuando hace pequeños cambios a las páginas JSP. Basta con refrescar el navegador con F5 o su correspondiente ícono de actualizar.



Tome en cuenta que para cambios de otro tipo como incluir recursos (imágenes, css, etc), agregar beans, incluir packages, agregar librerías, cambiar nombres a ficheros, entre otros, puede ser necesario que usted ejecute **clean and build** e incluso limpiar el caché del navegador en el caso que usted no pueda ver los cambios realizados. **Aunque en la mayoría de casos con ejecutar el proyecto o archivo desde Run File basta**, no olvide esta sugerencia.

7. Scriptlets en JSP. Para esta parte crear una nueva página llamada **“forJSP”,** y luego digitar el siguiente código dentro de las etiquetas **body** de su página (no olvide realizar el llamado a la hoja de estilo de Bootstrap y las etiquetas meta correspondientes dentro del header, las mismas del ejemplo anterior).

```
<div class="container">
    <div class="row">
        <h3>Tablas de multiplicar</h3>
```

```
    </div>
    <%
        for (int i = 1; i <= 10; i++) {
    %>
    <div class="panel panel-primary">
      <div class="panel-heading"><%= "Tabla del " + i%></div>
      <div class="panel-body">
      <%
         for (int j = 1; j <= 10; j++) {
           out.println("<p>"+i + " x " + j + " = " + (i * j) + "</p>");
         }//Cerrando el for más interno
      %>
      </div>
    </div>
        <% }//Cerrando el for más externo
        %>
</div>
```

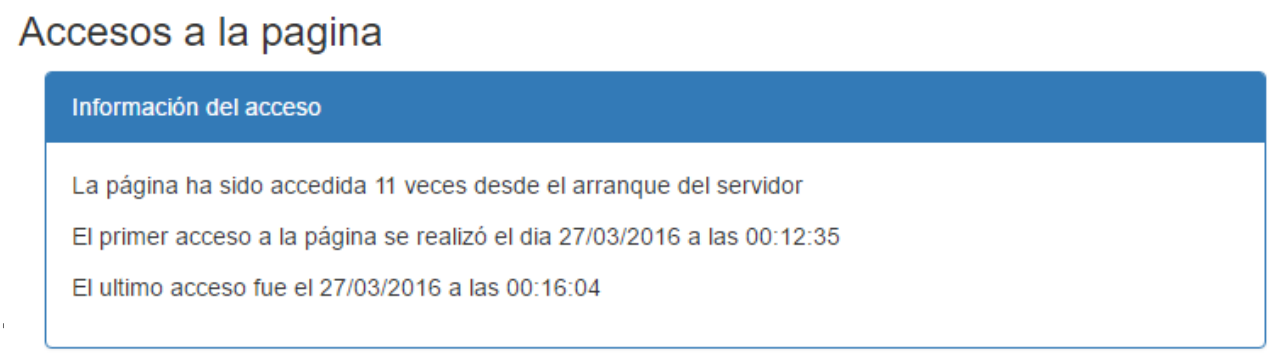
8. Declaraciones en JSP. Crear una nueva página llamada **"NumAcceso"** y luego digitar el siguiente código.

```
<%@page import="java.util.Date"%>
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.text.DateFormat"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%! int numeroAccesos = 0;
   DateFormat formatoHora = new SimpleDateFormat("HH:mm:ss");
   DateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
   java.util.Date primerAcceso = new java.util.Date();

   private Date ahora() {
      Date now = new Date();
      return now;
   }
%>

<!DOCTYPE html>
<html>
   <head>
      <title>Ejemplo</title>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="css/bootstrap.min.css">
   </head>
   <body>
      <div class="container">
         <div class="row">
            <h3>Accesos a la pagina</h3>
         </div>
         <div class="panel panel-primary">
            <div class="panel-heading">Información del acceso</div>
            <div class="panel-body">
```

```
            <p>
              <%="La página ha sido accedida " + (++numeroAccesos)
                  + " veces desde el arranque del servidor"%>
            </p>
            <p>
              <%="El primer acceso a la página se realizó el dia "
                  + formatoFecha.format(primerAcceso) + " a las "
                    + formatoHora.format(primerAcceso)%>
            </p>
            <p>
           <%= "El ultimo acceso fue el " + formatoFecha.format(ahora())
                  + " a las " + formatoHora.format(ahora())%>
            </p>
          </div>
        </div>
      </div>
    </body>
</html>
```



Ahora crearemos una aplicacion en la cual nos conectaremos a una base de datos, para ello seguir los siguientes pasos.

1. Cree la siguiente base de datos:

```
create database Guia8_POO1;

use Guia8_POO1;

create table empleados
(codigo int AUTO_INCREMENT PRIMARY KEY,
  nombre varchar(25) NOT NULL,
  apellidos varchar(25) NOT NULL,
  telefono  varchar(9) NOT NULL
);

Insert into empleados values(null,'Max', 'Planck', '78251231');
Insert into empleados values(null,' James Clerk ', 'Maxwell', '74561238');


create table tipo_usuarios(
id_tipo_usuario int AUTO_INCREMENT PRIMARY KEY,
tipo_usuario varchar(25));

create table usuarios(
```

```
id_usuario int AUTO_INCREMENT PRIMARY KEY,
nombres varchar(25),
apellidos varchar(25),
edad int,
id_tipo_usuario int,
usuario varchar(20),
password varchar(64),
UNIQUE (usuario),
constraint foreign key (id_tipo_usuario) references ti-
po_usuarios(id_tipo_usuario)
);

insert into tipo_usuarios values (null,'Tipo Usuario 1');
insert into tipo_usuarios values (null,'Tipo Usuario 2');

insert into usuarios values (null,'Root','No Last Name for
root',1,1,'root',SHA2('root',256));
insert into usuarios values (null,'Nikola','Tesla',34,2,'tesla',SHA2('corrienteAC',256));
```

2. Crear una página JSP que se llame **"conexion"** y digitar el siguiente código.

```
<%@ page language="java" import="java.sql.*" %>
<%
   Connection conexion = null;
   PreparedStatement st = null;
   ResultSet rs = null;
   //Leemos el driver de Mysql
   Class.forName("com.mysql.jdbc.Driver");

   // Se obtiene una conexión con la base de datos.
   conexion = DriverManager.getConnection(
         "jdbc:mysql://localhost/Guia8_POO1", "root", "");

%>
```

Nota: No olvide agregar el driver a las librerías de su proyecto.

3. Ahora crearemos una página llamada **"consulta"** y digitar el siguiente código, correr y observar el resultado.

```
<%@ include file="conexion.jsp"%>
<div class="row col-md-7">
   <table class="table table-striped table-bordered table-hover">
      <thead>
         <tr>
            <th>Nombres</th>
            <th>Apellidos</th>
            <th>Telefono</th>
            <th>Operaciones</th>
         </tr>
      </thead>
```

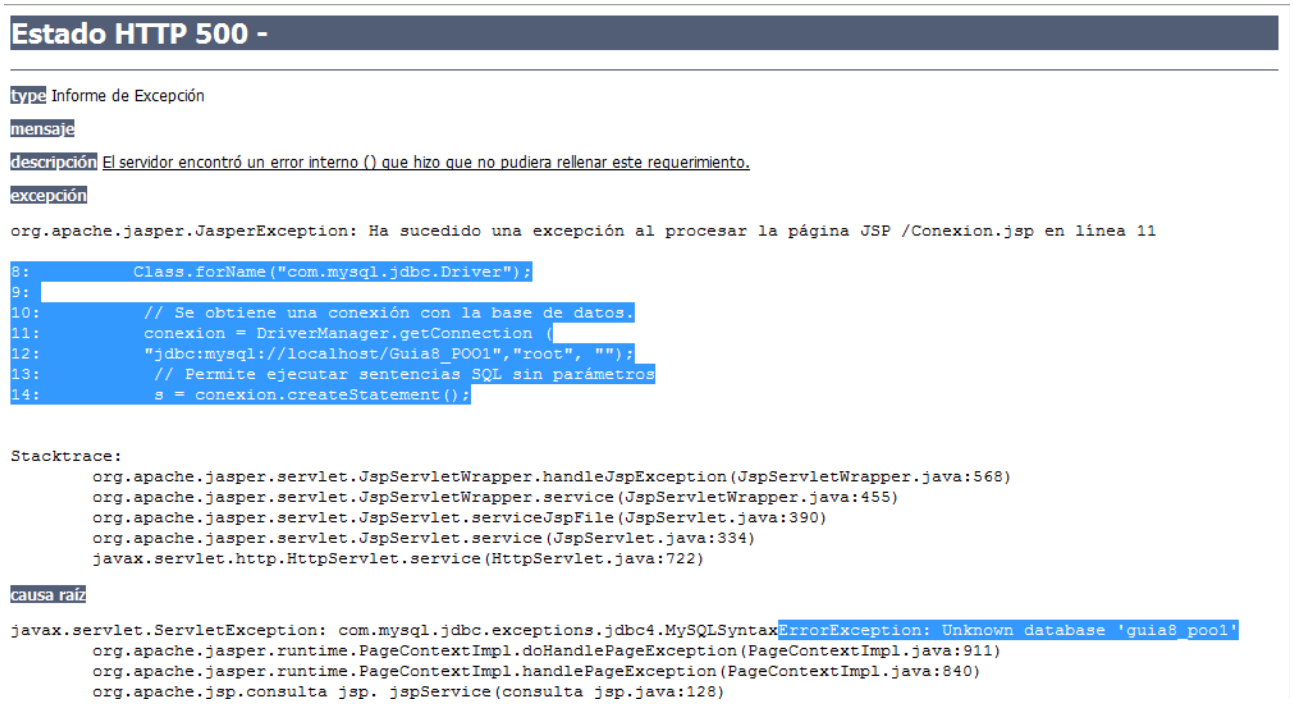
```
      <tbody>
        <%
            st = conexion.prepareStatement("select * from Empleados");
            rs = st.executeQuery();
            while (rs.next()) {
        %>
        <tr>
            <td><%=rs.getString("nombre")%></td>
            <td><%=rs.getString("apellidos")%></td>
            <td><%=rs.getString("telefono")%></td>
            <td><a class="btn btn-danger"
href='eliminar.jsp?id=<%=rs.getString("codigo")%>'>Eliminar</a></td>
        </tr>
        <%
            }
            conexion.close();
        %>

      </tbody>
    </table>
</div>
```

Vea el resultado de consulta.jsp en su navegador (los estilos del framework Bootstrap se aplicarán posteriormente).

Nombres	Apellidos	Telefono	Operaciones
Max	Planck	78251231	Eliminar
James Clerk	Maxwell	74561238	Eliminar

Nota: De ahora en adelante, la mayoría de errores en las páginas aparecerán en la traza de error que el contenedor web retorna (tomcat, glassfish, jboss, websphere, etc) y que son visibles en el navegador.



Cuando se encuentre con estos problemas, procure identificar la sección del código afectada y encontrar la causa raíz del error. Muchos de los errores se producen en tiempo de ejecución y no en la codificación, por lo que es importante que sepa leer estas trazas de errores.

4. Ingresaremos datos a la base, para ello debe crear una página JSP llamada **"ingresaremp"**, con el siguiente código.

```
<%@ page language="java" import="java.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>Empleados</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <h3>Empleados</h3>
      </div>
      <div class="row col-md-5" >
        <form role="form" action="ingresar.jsp" method="POST">
          <div class="col-md-10">
            <div class="well well-sm"><strong><span class="glyphicon glyphicon-
asterisk"></span>Campos requeridos</strong></div>
            <div class="form-group">
              <label for="nombre">Nombres:</label>
              <div class="input-group">
                <input type="text" class="form-control" name="nombre" id="nombre"
```

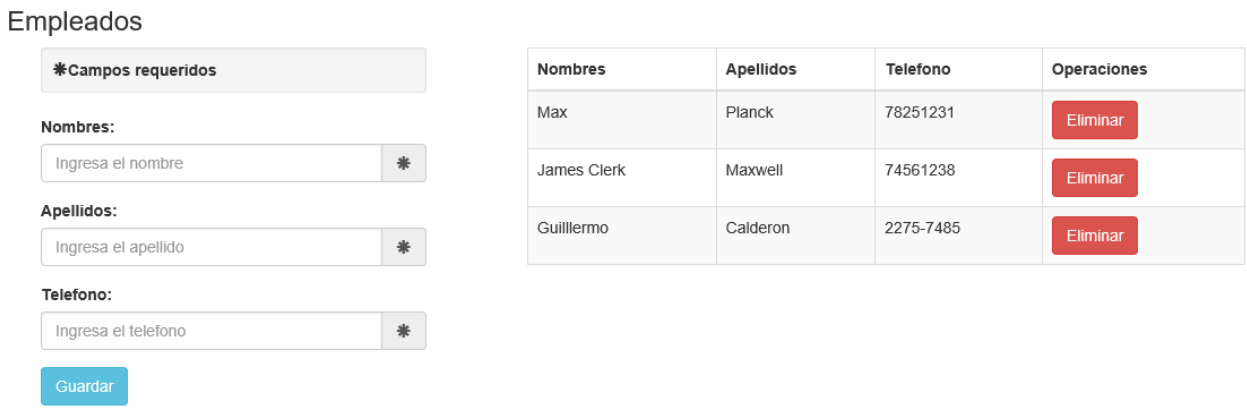
```
placeholder="Ingresa el nombre" required>
                        <span class="input-group-addon"><span class="glyphicon glyphicon-
asterisk"></span></span>
                    </div>
                </div>
                <div class="form-group">
                    <label for="apellido">Apellidos:</label>
                    <div class="input-group">
                        <input type="text" class="form-control" id="apellido" name="apellido"
placeholder="Ingresa el apellido" required>
                        <span class="input-group-addon"><span class="glyphicon glyphicon-
asterisk"></span></span>
                    </div>
                </div>
                <div class="form-group">
                    <label for="telefono">Telefono:</label>
                    <div class="input-group">
                        <input type="text" class="form-control" id="telefono" name="telefono"
placeholder="Ingresa el telefono" required>
                        <span class="input-group-addon"><span class="glyphicon glyphicon-
asterisk"></span></span>
                    </div>
                </div>
                <input type="submit" class="btn btn-info" value="Guardar">
            </div>
        </form>
         <%-- Verificando si la variable resultado esta vacia--%>
        <% if (request.getParameter("resultado")!=null){
         %>
          <div class="alert alert-success col-md-10">
              <b> <%= request.getParameter("resultado") %></b>
          </div>
        <%
        }
        %>
       </div>

       <!--notese el uso de jsp:include -->
       <jsp:include page="consulta.jsp" />

    </div> <!-- /container -->

  </body>
</html>
```

Su página debería lucir de la siguiente manera:



5. Ahora crearemos la página que realizará el query para ingresar los datos tomados del formulario de la página **"ingresaremp"**. Como se pudo observar en el action se digitó la página donde serán enviados los datos a ser capturados, está página será creada con el nombre **"ingresar.jsp"** y luego deberá digitar el siguiente código:

```
<%@ include file="conexion.jsp"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>

   <%
    //Capturando los datos por medio del request y el metodo getParameter
        String nombre = request.getParameter("nombre");
        String apellido = request.getParameter("apellido");
        String telefono = request.getParameter("telefono");
        //Sentencia sql para ingresar datos
        st=conexion.prepareStatement("INSERT INTO empleados values (null,?,?,?)");
        st.setString(1,nombre);
        st.setString(2,apellido);
        st.setString(3,telefono);
        st.executeUpdate();
        conexion.close();
     %>
     <%--Forward que se utiliza para redireccionar a la pagina de ingresaremp.jsp--%>
     <jsp:forward page="ingresaremp.jsp">
        <jsp:param name="resultado" value="Datos Ingresados Exitosamente"/>
     </jsp:forward>
  </body>
</html>
```

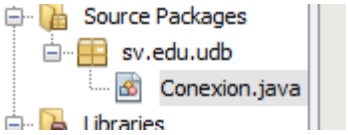
Manejo de sesiones con JDBC

Ejemplo práctico: Administración de usuarios.

Un caso práctico donde usar las sesiones es en las páginas a las que se debe acceder habiendo

introducido previamente un usuario y una clave. Si no se introducen estos datos no se podrán visualizar y de igual manera si alguien intenta entrar directamente a una de estas páginas sin haberse identificado; será redirigido a la página principal para que se identifique. No podrá acceder a los recursos del sitio de manera anónima.

Primero debe crear una clase Conexion dentro de un paquete **sv.edu.udb**, agregar el siguiente código:
Nota: Es la misma clase Conexion de la guía 6 no es necesario que digite todo el código, incluya el archivo o copie y pegue el contenido. Solo debe asegurarse de usar la base de datos correcta.




```
package sv.edu.udb;

import java.sql.*;

public class Conexion {
    private Connection conexion =null;
    private Statement s =null;
    private ResultSet rs=null;
    private String query="";

   //Contructor
   public Conexion() throws SQLException{
        try
       {
            //obtenemos el driver de para mysql
           Class.forName("com.mysql.jdbc.Driver");
           // Se obtiene una conexión con la base de datos. 2
           conexion = DriverManager.getConnection (
              "jdbc:mysql://localhost/Guia8_POO1","root", "");
            // Permite ejecutar sentencias SQL sin parámetros
            s = conexion.createStatement();
             }
       catch (ClassNotFoundException e1) {
              //Error si no puedo leer el driver de MySQL
   System.out.println("ERROR:No encuentro el driver de la BD: " +e1.getMessage());
       }
   }
   //Metodo que permite obtener los valores del resulset
   public ResultSet getRs() {
       return rs;
   }
   //Metodo que permite fijar la tabla resultado de la pregunta
   //SQL realizada
   public void setRs(String consulta)  {
       try {
           this.rs = s.executeQuery(consulta);
       } catch (SQLException e2) {
              System.out.println("ERROR:Fallo en SQL: "+e2.getMessage());
       }
   }
```

```
    //Metodo que recibe un sql como parametro que sea un update,insert.delete
    public void setQuery(String query) throws SQLException {
        this.s.executeUpdate(query);
    }

    //Metodo que cierra la conexion
    public void cerrarConexion() throws SQLException{
        conexion.close();
    }
}
```

login.jsp

La primera página de la aplicación JSP es en la que el usuario se debe identificar con un nombre de usuario y una clave.

Ésta página JSP contiene un formulario en donde se especifica la página destino luego de que el usuario pulse el botón de enviar los datos. Además, se ha añadido una comprobación en caso de recibir un parámetro llamado "error" y se muestra el mensaje que contenga. De esta forma el usuario ve qué tipo de error se ha producido.

```
<%@page session="true" language="java" import="java.util.*" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Inicio de sesión</title>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="css/bootstrap.min.css">
    </head>
    <body>
        <div class="container-fluid">
            <div class="row">
                <div class="col-sm-4 col-sm-offset-4">
                    <h2>Inicio de sesión</h2>
                    <%
                        if (request.getParameter("error") != null) {
                    %>
                    <div class="alert alert-danger">
                     <strong>Error!</strong> <%=request.getParameter("error")%>
                     <br>
                    </div>
                    <%
                        }//Fin del if
                        %>
                    <form action="checklogin.jsp"  method="post">
                        <div class="form-group">
                            <label for="usuario">Usuario</label>
                            <input type="text" class="form-control" id="usuario" placehold-
```

```
er="Usuario" name="usuario" required>
                    </div>

                    <div class="form-group">
                      <label for="clave">Password:</label>
                      <input type="password" class="form-control" id="clave" placehold-
er="Password" name="clave" required>
                    </div>

                    <div class="form-group">
                      <button class="btn btn-lg btn-primary btn-block" type="submit">Iniciar
sesión</button>
                    </div>
                  </form>
                </div>
              </div>
            </div>
          </body>
</html>
```

checklogin.jsp

Esta página es la encargada de recoger el usuario y la clave enviados desde el formulario. Una vez recibidos se almacenan en dos variables(“usuario” y “clave”) de tipo String. A continuación se comparan con los valores correctos contra los datos de la base de datos.

Si esta comprobación es correcta se crea un objeto de tipo session y se guarda el valor en la variable “usuario” en la sesión mediante el método setAttribute().

A continuación y mediante la opción estándar <jsp: fordward> se redirecciona al usuario a la página final en la que se encuentra el menú de opciones al que se accede después de haber completado de forma satisfactoria el proceso de identificación.

En caso que la comprobación de usuario y clave no se cumpla se redirecciona al usuario hacia la página de inicio, para que vuelva a identificarse incluyendo esta vez un parámetro llamado “error” con un mensaje que avisará qué es lo que le ha ocurrido.

NOTA: Los contraseñas almacenadas en la base de datos se han encriptado usando **SHA-2** el cual es considerado en la actualidad como uno de los algoritmos criptográficos más seguros. La encriptación de los datos es una práctica muy habitual en la industria del desarrollo del software y tiene por objetivo asegurar la confidencialidad de datos sensibles.

id_usuario	nombres	apellidos	edad	id_tipo_usuario	usuario	password
1	Root	No Last Name for root	1	1	root	4813494d137e1631bba301d5acab6e7bb7aa74ce1185d45656...
2	Nikola	Tesla	34	2	tesla	ddf6090b26ba1463b7a32dd7be10512f906dcd5e011d70a849...

Conceptualmente es imposible obtener el texto original a partir del hash (texto encriptado) generado por el algoritmo, debido a esto lo que se hace es encriptar el texto ingresado por el usuario en el formulario y comparar ese texto encriptado con el valor almacenado en la base de datos.

En este ejemplo se han generado claves criptográficas de 256 bits, es decir 64 caracteres hexadecimales.

```
<%@page session="true" language="java" import="java.util.*" %>
<%@ page import="sv.edu.udb.Conexion, java.sql.*" %>
```

```
<%
        String usuario=request.getParameter("usuario");
        String clave=request.getParameter("clave");
        Conexion con = new Conexion();
             //buscará una coincidencia (count usuario), si es correcto
        //podrá loguearse
        con.setRs("select count(usuario),nombres from usuarios"
              + " where usuario='"+usuario+"' and "
              + "password=SHA2('"+ clave +"',256)");
        //Se esta usando SHA2 con claves de 256 bits

        ResultSet rs = con.getRs();
        rs.next();
        if(rs.getInt(1)==1){ //solo una coincidencia es permitida
        //si se puede loguear, hay que evitar que quede una conexión
        //activa
        rs.close();
        con.cerrarConexion();

        //se asignan los parámetros de sesión
        HttpSession sesionOk = request.getSession();
        sesionOk.setAttribute("usuario", usuario);
           %>
           <jsp:forward page="menu.jsp" />
           <%
        }
        else{
           %>

            <jsp:forward page="login.jsp">
            <jsp:param name="error" value="Usuario y/o clave
            Incorrecto. Vuelve a intentarlo."/>
            </jsp:forward>

           <%
        }
        rs.close();
        con.cerrarConexion();
%>
```

menu.jsp

La página contiene la comprobación de que se ha realizado el proceso de login previamente. En caso de no ser así, es decir, que se ha entrado de forma “no correcta”, se redirige de nuevo al usuario a la página de login.jsp, para que se identifique. Para esta comprobación se recupera el valor que supuestamente ya estaría en la sesión. Si este valor es nulo, quiere decir que la sesión no se ha creado y por lo tanto es necesario que se identifique. El proceso para redireccionar al usuario a la página de login.jsp también lleva el parámetro “error” con un mensaje que será visto en la primera

página.

Si por el contrario, la sesión sí ha sido creada, quiere decir que el usuario ha sido identificado de forma correcta, por lo que la variable usuario creada previamente se almacena el nombre del usuario utilizado, para ser mostrado posteriormente.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page session="true" %>
<%
   String usuario = "";
   HttpSession sesionOk = request.getSession();
   if (sesionOk.getAttribute("usuario") == null) {
%>
<jsp:forward page="login.jsp">
   <jsp:param name="error" value="Es obligatorio identificarse"/>
</jsp:forward>
<%
   } else {
      usuario = (String) sesionOk.getAttribute("usuario");
   }
%>

<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Menu principal</title>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="css/bootstrap.min.css">
      <script src="js/jquery.js"></script>
      <script src="js/bootstrap.min.js"></script>
   </head>
   <body>
   <nav class="navbar navbar-default">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed"
              data-toggle="collapse" data-target="#navbar"
              aria-expanded="false" aria-controls="navbar">
          <span class="sr-only">Desplegar navegación</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">Ejemplo POO</a>
      </div>
      <div id="navbar" class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li class="active"><a href="#">Inicio</a></li>
          <li><a href="ingresaremp.jsp">Empleados</a></li>
          <li class="dropdown">
```

```
          <a href="#" class="dropdown-toggle" data-toggle="dropdown"
            role="button" aria-haspopup="true"
            aria-expanded="false">Usuarios <span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="opc1.jsp">Crear usuario</a></li>
            <li><a href="opc2.jsp">Borrar usuario</a></li>
            <li><a href="opc3.jsp">Cambiar clave</a></li>
          </ul>
        </li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="cerrarsesion.jsp">
          <%=usuario%> (cerrar sesión)</a></li>
      </ul>
    </div>
  </div>
</nav>

    <div class="container">
    <h3>PROCESO DE IDENTIFICACIÓN</h3>
    <p>
      <h5>Menú de administración</h5>
      <b>Usuario activo:</b> <%=usuario%>
    </p>
    </div>
</body>
</html>
```

cerrarsesion.jsp

La última opción que incorpora el menú es la de “Cerrar sesión”, que será de gran utilidad cuando se haya finalizado el trabajo y queremos estar seguro que nadie realiza ninguna acción con nuestro usuario y clave.

Al pulsar este enlace, se recupera de nuevo la sesión y mediante el método invalidate() se da por finalizada la sesión.

```
<%@ page session="true" %>
<%
    HttpSession sesionOk = request.getSession();
    sesionOk.invalidate();
%>
<jsp:forward page="login.jsp"/>
```

Intente acceder sin loguearse a http://localhost:8080/GUIA8/menu.jsp para comprobar el manejo de sesiones. Los usuarios disponibles son root con password root y tesla con password corrienteAC

IV. EJERCICIOS COMPLEMENTARIOS

1. **Tomando de base el proyecto del segundo periodo realizar el mantenimiento de una tabla y manejar los usuarios a partir de la base de datos.**

2. **Implementar la operación “eliminar” de la página ingresaremp.jsp.**

Hoja de cotejo.

HOJA DE EVALUACIÓN

Carnet:	Alumno:
Fecha:	Docente:
No.:	Título de la guía:

Actividad a evaluar	Criterio a evaluar	Cumplió		Puntaje
		SI	NO	
Desarrollo	Realizó los ejemplos de guía de práctica (40%)			
	Presentó todos los problemas resueltos (20%)			
	Funcionan todos correctamente y sin errores (30%)			
	Envió la carpeta comprimida y organizada adecuadamente en subcarpetas de acuerdo al tipo de recurso (10%)			
	PROMEDIO:			
Investigación complementaria	Entregó la investigación complementaria en la fecha indicada (20%)			
	Resolvió todos los ejercicios planteados en la investigación (40%)			
	Funcionaron correctamente y sin ningún mensaje de error a nivel de consola o ejecución (40%)			
	PROMEDIO:			