

Ejercicio guiado

-Mantenimiento de Registros, usando
Capas, servlet, sesiones, bootstrap,
segmentos jspf y hojas de estilo



**Universidad
Tecnológica
del Perú**

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Ingrese sus datos

Usuario

Password

☐ Recordarme

Ingresar

a

Bienvenidos

localhost:8084/Ejemplo11/menuPrincipal.jsp

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR Cerrar Session

MENU PRINCIPAL

MANTENIMIENTO DE DATOS

- Usuario
- Area
- Puesto
- Epp
- Asignación Epp por Puesto

REGISTRO DE ENTREGA

- registroEntregaEpp

REPORTES

- Usuario
- Area

© Héctor Delgado

Cambiar de menú
(Utilizar un menú personalizado
Puede utilizar el código y modelo de
su agrado)

Mantenimiento de datos del

localhost:8084/Ejemplo11/mantenimientoUsuario.jsp

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR Cerrar Session

Mantenimiento de datos del usuario **Nuevo** **b.1**

Ingrese datos del Usuario a buscar: Usuario

Consultar **b.2**

Regresar

© Héctor Delgado

b.1.1

b.2.1

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR Cerrar Session

Codigo: 9

Usuario: HECTOR

Password: 123456

Nombres: HECTOR

Apellidos: DELGADO

Regresar

© Héctor Delgado

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR Cerrar Session

Ingrese datos del nuevo Usuario

Codigo

Usuario

Password

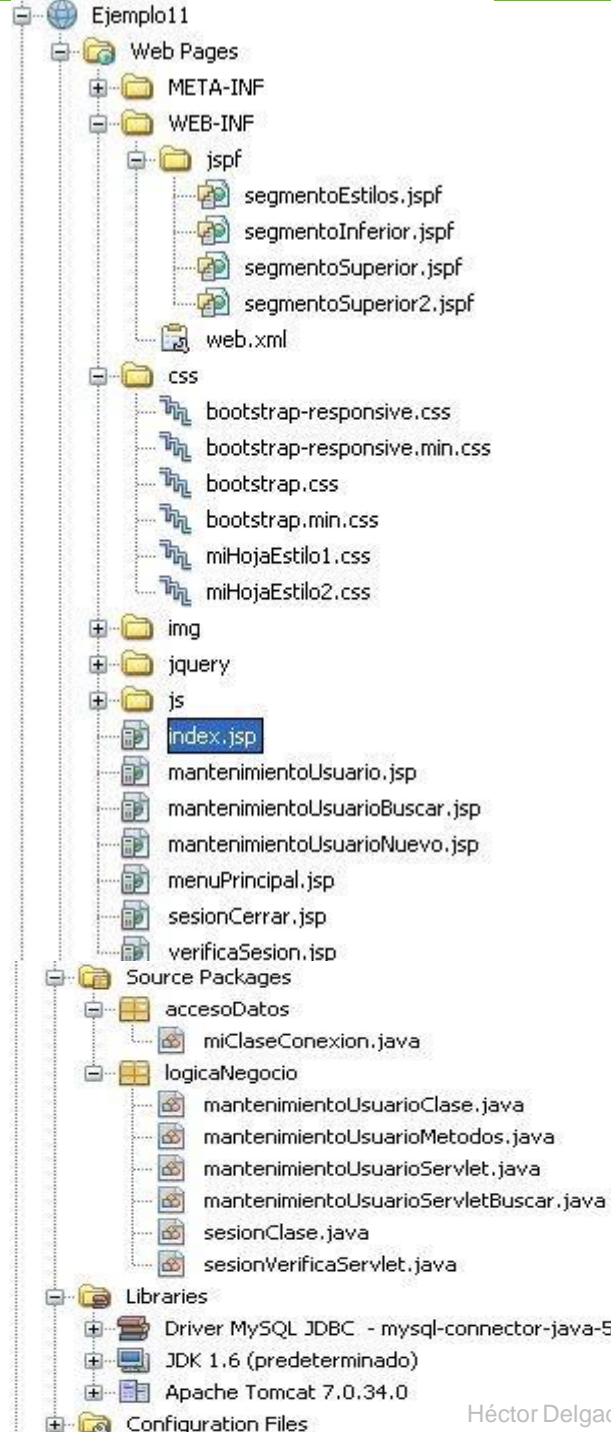
Nombres

Apellidos

Guardar Cancelar

© Héctor Delgado

Análisis



Encabezados y pie de pagina

Hojas de Estilo y Bootstrap

jquery

Capa presentación

Capa acceso a Datos

Capa lógica de negocios

Driver mysql

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

- 1 Jalar el contenido de la carpeta bootstrap al proyecto(Directorio Web Pages)



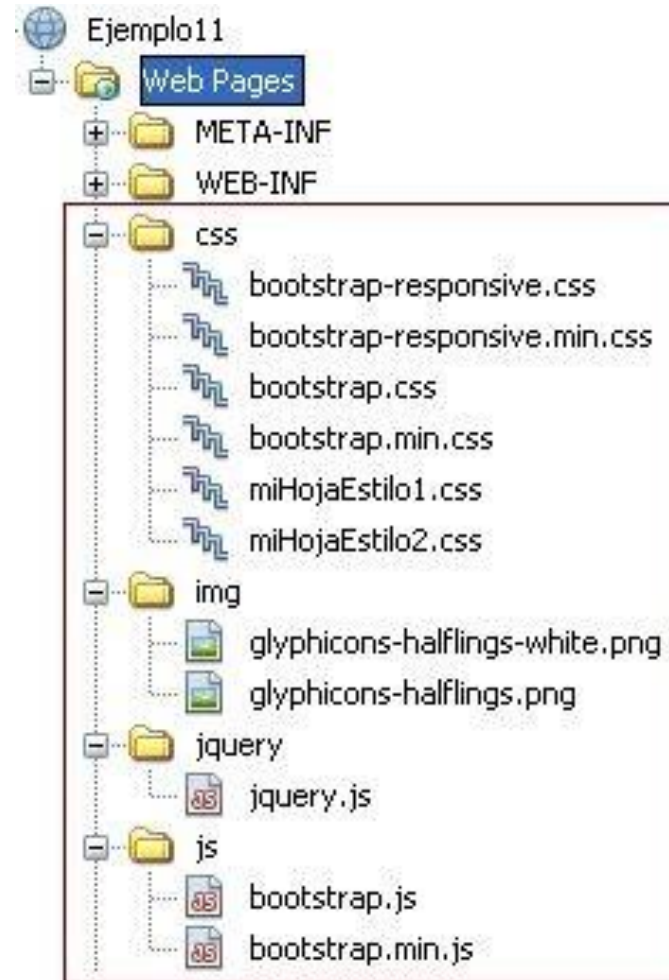
- 2 Jalar la carpeta jquery al proyecto (Directorio Web Pages)



Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

3 Observar que el contenido será el siguiente:



Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

- 4 Crear una carpeta dentro del directorio WEB-INF con el nombre: **jspf**
Dentro de la carpeta jspf, crear los siguientes archivos:



Nombre y ubicación

File Name:

Proyecto:

Ubicación: ▼

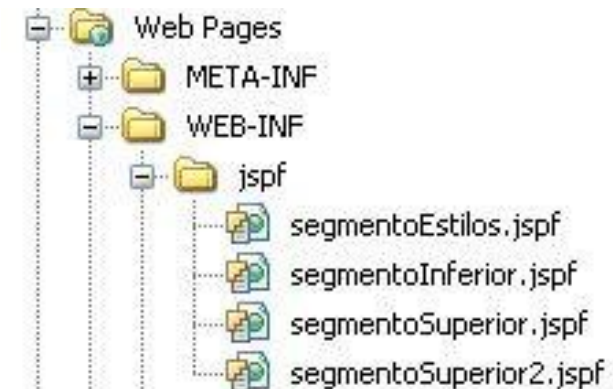
Carpeta:

Archivo creado:

Options:

☒ JSP File (Standard Syntax) ☒ Create as a JSP Segment

☐ JSP Document (XML Syntax)



5 Modificar el contenido de:

Encabezado

```
segmentoSuperior.jspf x
1 <!-- any content can be specified here e.g.: --%>
2 <%@ page pageEncoding="UTF-8" %>
3 <div>
4   <div class="miEncabezado">
5     Sistema de Registro de Equipos de Proteccion Personal(EPP)
6   </div>
7   <div class="miContenido">
8   </div>
```

Contenido de la página

6

Pie de
página

```
segmentoInferior.jspf x
1 <!-- any content can be specified here e.g.: --%>
2 <%@ page pageEncoding="UTF-8" %>
3   <div class="miPiePagina">
4     &COPY; Héctor Delgado
5   </div>
6 </div>
```

7 Modificar el contenido de:

2do Encabezado, usualmente es donde va el menú(En caso el usuario ingrese en sesión se mostrara el menú)



```
<%-- any content can be specified here e.g.: --%>
2  <%@ page pageEncoding="UTF-8" %>
3  <div>
4      <div class="miEncabezado">
5          Sistema de Registro de Equipos de Proteccion Personal(EPP)
6          <div class="miSubEncabezado a1">
7              <a href="menuPrincipal.jsp" ><b><em><font color="#FFFFCC">Inicio</font></em></b></a>
8          </div>
9      </div>
10     <div class="miContenido">
11     </div>
```

8 Para evitar colocar el mismo código en todas las paginas se crea un agrupador **segmentoEstilos.jspf**



```
<%-- any content can be specified here e.g.: --%>
2  <%@ page pageEncoding="UTF-8" %>
3  <link href="css/miHojaEstilo1.css" rel="stylesheet" type="text/css"/>
4  <link href="css/miHojaEstilo2.css" rel="stylesheet" type="text/css"/>
5  <link href="css/bootstrap.css" rel="stylesheet" type="text/css" />
6  <link href="css/bootstrap.min.css" rel="stylesheet" type="text/css"/>
7  <script src="jquery/jquery.js" type="text/javascript"> </script>
8  <script src="js/bootstrap.js" type="text/javascript"> </script>
9  <script src="js/bootstrap.min.js" type="text/javascript"> </script>
```


Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo

9 Análisis



9 Análisis

Sistema de Registro de Equipos

Ingrese sus datos

Usuario

Password

☐ Recordarme

Ingresar

segmentoSuperior.jspf



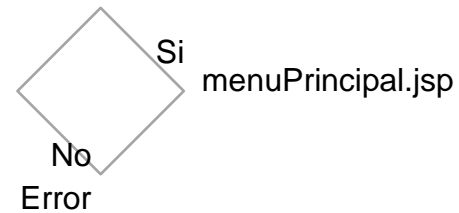
•Form

web.xml

logicaNegocio

sesionVerificaServlet.java

- txtUsuario
- Select



segmentoInferior.jspf

9 Modificar el contenido de Sistema de Registro de Equip

Ingrese sus datos

☐ Recordarme

```
index.jsp x
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <%@include file="WEB-INF/jspf/segmentoEstilos.jspf" %>
7      <title> Bienvenidos</title>
8  </head>
9  <body>
10     <%@include file="WEB-INF/jspf/segmentoSuperior.jspf" %>
11     <div class="ubicacion">
12         <form method="post" action="urlXMLServletVerificaSesion" class="form-signin">
13             <h3 class="form-signin-heading">Ingrese sus datos</h3>
14             <input type="text" name="txtUsuario" class="form-control" placeholder="Usuario" required autofocus/>
15             <input type="password" name="txtPassword" class="form-control" placeholder="Password" required/>
16             <label class="checkbox">
17                 <input type="checkbox" value="recordarme" id="recordarme" name="recordarme"> Recordarme
18             </label>
19             <button class="btn btn-small btn-primary btn-block" type="submit">Ingresar</button>
20         </form>
21     </div>
22     <%@include file="WEB-INF/jspf/segmentoInferior.jspf" %>
23 </body>
24 </html>
```

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

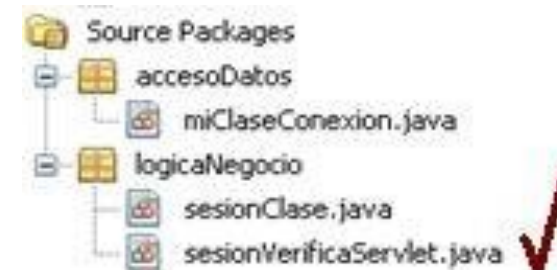
10 Insertamos un servlet con el nombre: sesionVerificaServlet



Nombre y ubicación

Nombre de Clase:

☒ Add information to deployment descriptor (web.xml)



Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo

- 10 Luego abrimos el archivo **WEB.XML** y modificamos:
el **name** : nombreXMLServletVerificaSesion
y por ultimo la **url**: urlXMLServletVerificaSesion



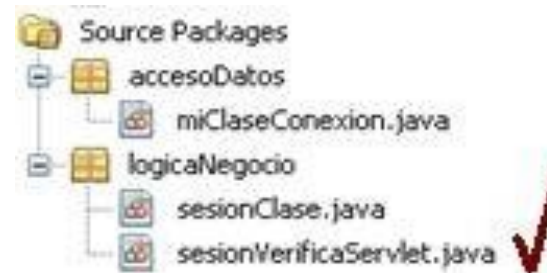
```
web.xml x
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi=
3  <servlet>
4      <servlet-name>nombreXMLServletVerificaSesion</servlet-name>
5      <servlet-class>logicaNegocio.sesionVerificaServlet</servlet-class>
6  </servlet>
7  <servlet-mapping>
8      <servlet-name>nombreXMLServletVerificaSesion</servlet-name>
9      <url-pattern>/urlXMLServletVerificaSesion</url-pattern>
10 </servlet-mapping>
11 <session-config>
12 <session-timeout>
13     30
14 </session-timeout>
15 </session-config>
16 </web-app>
```


Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo

11 Modificar el contenido de :

```
sesionVerificaServlet.java x
1  package logicaNegocio;
2
3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import javax.servlet.ServletException;
6  import javax.servlet.http.HttpServlet;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9  import javax.servlet.http.HttpSession;
10
```

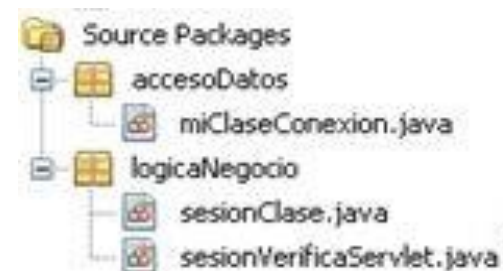


11 Modificar el contenido de :

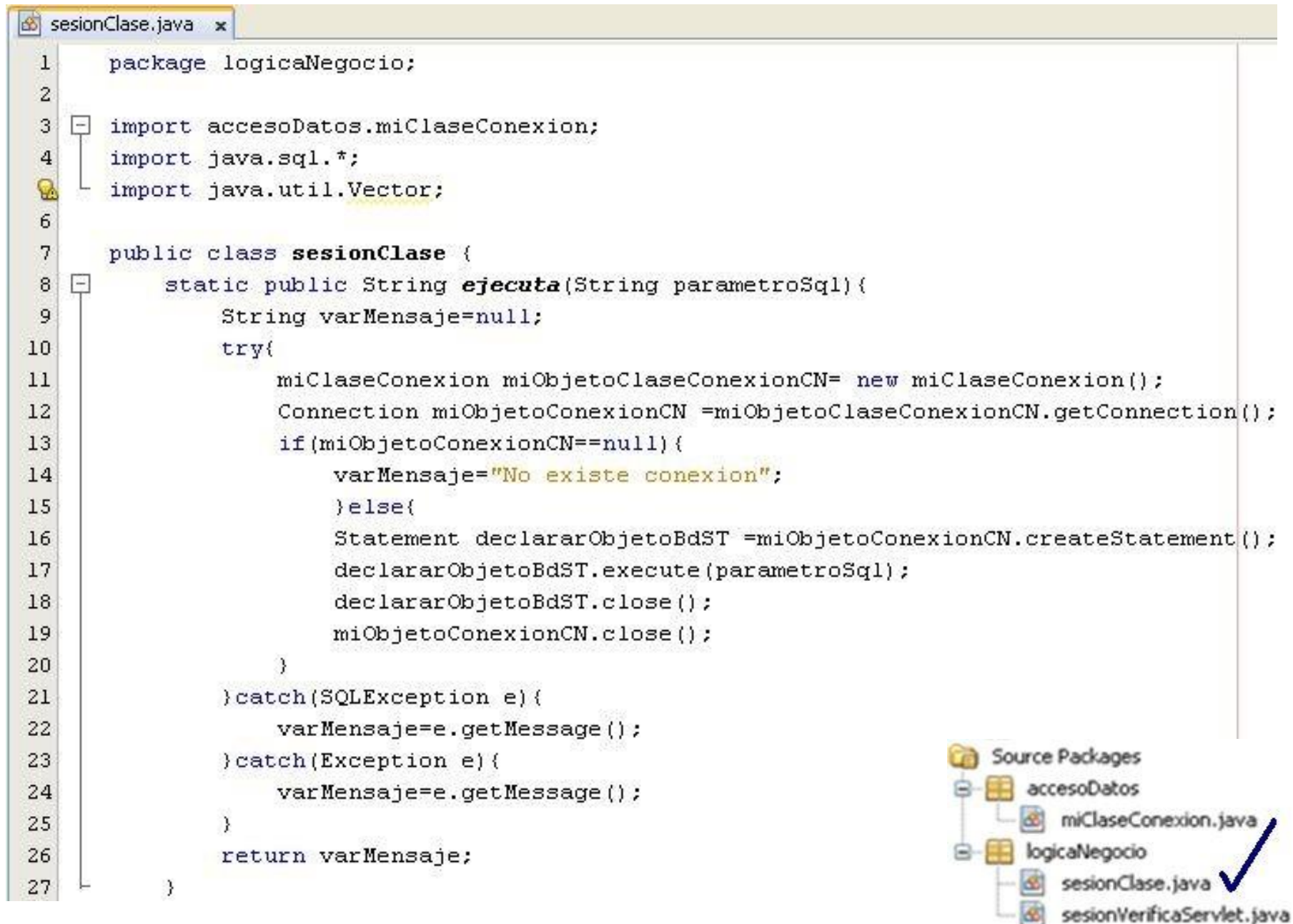
sesionVerificaServlet.java x

```
11 public class sesionVerificaServlet extends HttpServlet {
12     protected void processRequest(HttpServletRequest objetoPeticion1, HttpServletResponse objetoRespuesta1)
13     throws ServletException, IOException {
14         objetoRespuesta1.setContentType("text/html;charset=UTF-8");
15         PrintWriter out = objetoRespuesta1.getWriter();
16         try {
17             String varUsuario =objetoPeticion1.getParameter("txtUsuario");
18             String varPassword =objetoPeticion1.getParameter("txtPassword");
19             String [] datos=sessionClase.getFila("select * from mibase1.tbusuario2 where usuario='"+varUsuario+"' and password='"+varPassword+"'");
20             if (datos==null){
21                 out.println("<html>");
22                 out.println("Error, el usuario ");
23                 out.println("<b>"+varUsuario+"</b>");
24                 out.println(", no cumple con los datos solicitados, Para reingresar de clic<a href='index.jsp'>Aqui</a>");
25                 out.println("</html>");
26             }else{//Capturamos las variables
27                 String varCodigo=datos[0];
28                 String varNombres=datos[3];
29                 String varApellidos=datos[4];
30                 HttpSession session=objetoPeticion1.getSession(true);//Obtenemos los objetos a guardar en la sesion
31                 session.setAttribute("varCodigoSesion",varCodigo);
32                 session.setAttribute("varNombresSesion",varNombres);
33                 session.setAttribute("varApellidosSesion",varApellidos);
34                 objetoRespuesta1.sendRedirect("menuPrincipal.jsp");
35             } } finally {out.close(); }
36         }
37     }
77 }
```

HttpServlet methods. Click on the + sign on the left to edit the code.



12 Modificar el contenido de :



```
1 package logicaNegocio;
2
3 import accesoDatos.miClaseConexion;
4 import java.sql.*;
5 import java.util.Vector;
6
7 public class sessionClase {
8     static public String ejecuta(String parametroSql){
9         String varMensaje=null;
10        try{
11            miClaseConexion miObjetoClaseConexionCN= new miClaseConexion();
12            Connection miObjetoConexionCN =miObjetoClaseConexionCN.getConnection();
13            if(miObjetoConexionCN!=null){
14                varMensaje="No existe conexion";
15            }else{
16                Statement declararObjetoBdST =miObjetoConexionCN.createStatement();
17                declararObjetoBdST.execute(parametroSql);
18                declararObjetoBdST.close();
19                miObjetoConexionCN.close();
20            }
21        }catch(SQLException e){
22            varMensaje=e.getMessage();
23        }catch(Exception e){
24            varMensaje=e.getMessage();
25        }
26        return varMensaje;
27    }
28 }
```

Source Packages

- accesoDatos
 - miClaseConexion.java
- logicaNegocio
 - sessionClase.java ✓
 - sesionVerificaServlet.java

12 Modificar el contenido de :

sesionClase.java

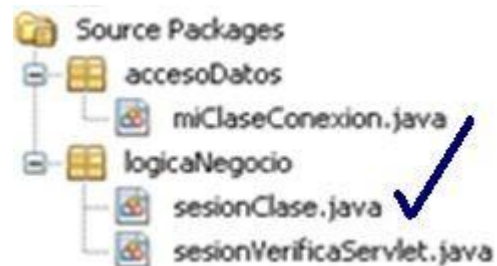
```
static public Vector consulta(String parametroSql){
    Vector registrosVector =new Vector();
    try{
        miClaseConexion miObjetoClaseConexionCN= new miClaseConexion();
        Connection miObjetoConexionCN =miObjetoClaseConexionCN.getConnection();
        if(miObjetoConexionCN==null){
            registrosVector=null;
        }else{Statement declararObjetoBdST =miObjetoConexionCN.createStatement();
            ResultSet datosObtenidosRs =declararObjetoBdST.executeQuery(parametroSql);
            ResultSetMetaData identificaDatosObtenidos=datosObtenidosRs.getMetaData();
            int numeroColumnas =identificaDatosObtenidos.getColumnCount();
            String[]tituloColumnas=new String[numeroColumnas];
            for(int i=0; i<numeroColumnas;++i)
                tituloColumnas[i]=identificaDatosObtenidos洗getColumnNane(i+1);
            registrosVector.add(tituloColumnas);
            while(datosObtenidosRs.next()){
                String []registroIdentificado=new String [numeroColumnas];
                for(int i=0; i<numeroColumnas;i++){
                    registroIdentificado[i]=datosObtenidosRs.getString(i+1);
                } registrosVector.add(registroIdentificado);}
            datosObtenidosRs.close();
            declararObjetoBdST.close();
            miObjetoConexionCN.close();
        }
    }catch(SQLException e){
        registrosVector=null;
    }catch(Exception e){
        registrosVector=null;    }
    return registrosVector;
}
```

Source Packages

- accesoDatos
 - miClaseConexion.java
- logicaNegocio
 - sesionClase.java ✓
 - sesionVerificaServlet.java

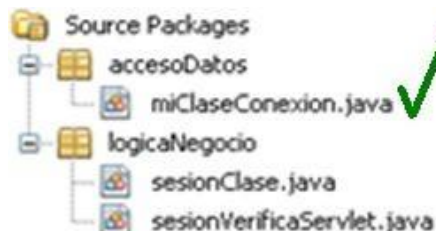
12 Modificar el contenido de :

```
sesionClase.java x
57 //
58 //RETORNA UNA SOLA FILA
59 static public String[] getFila(String parametroSql){
60     Vector registrosVectorConsultados=consulta(parametroSql);
61     String[]fila=null;
62     if(registrosVectorConsultados!=null)
63     if(registrosVectorConsultados.size()>1) //Si existen filas
64         fila=(String [])registrosVectorConsultados.get(1);//En 0 estan los titulos
65     return fila;
66 }
67
68 static public String getCampo(String parametroSql){
69     String []fila=getFila(parametroSql);
70     String campo=null;
71     if (fila!=null) //No hay campo
72         campo=fila[0];
73     return campo;
74 }
75 }
```



13 Modificar el contenido de :

```
miClaseConexion.java x
1 package accesoDatos;
2
3 import java.sql.*;
4 import java.sql.Connection;
5 import java.sql.DriverManager;
6
7 public class miClaseConexion {
8     private String driver="com.mysql.jdbc.Driver";
9     private String url="jdbc:mysql://localhost/mibase1";
10    private String login="root";
11    private String password="";
12
13    public Connection getConnection(){
14        Connection miConexionCN=null;
15        try{
16            Class.forName(driver).newInstance();
17            miConexionCN=DriverManager.getConnection(url, login, password);
18        }catch(SQLException e){
19            System.out.println(e.toString());
20            miConexionCN=null;
21        }catch(Exception e){
22            System.out.println(e.toString());
23            miConexionCN=null;
24        }
25        return miConexionCN;
26    }
27 }
```



Source Packages

- accesoDatos
 - miClaseConexion.java ✓
- logicaNegocio
 - sesionClase.java
 - sesionVerificaServlet.java

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

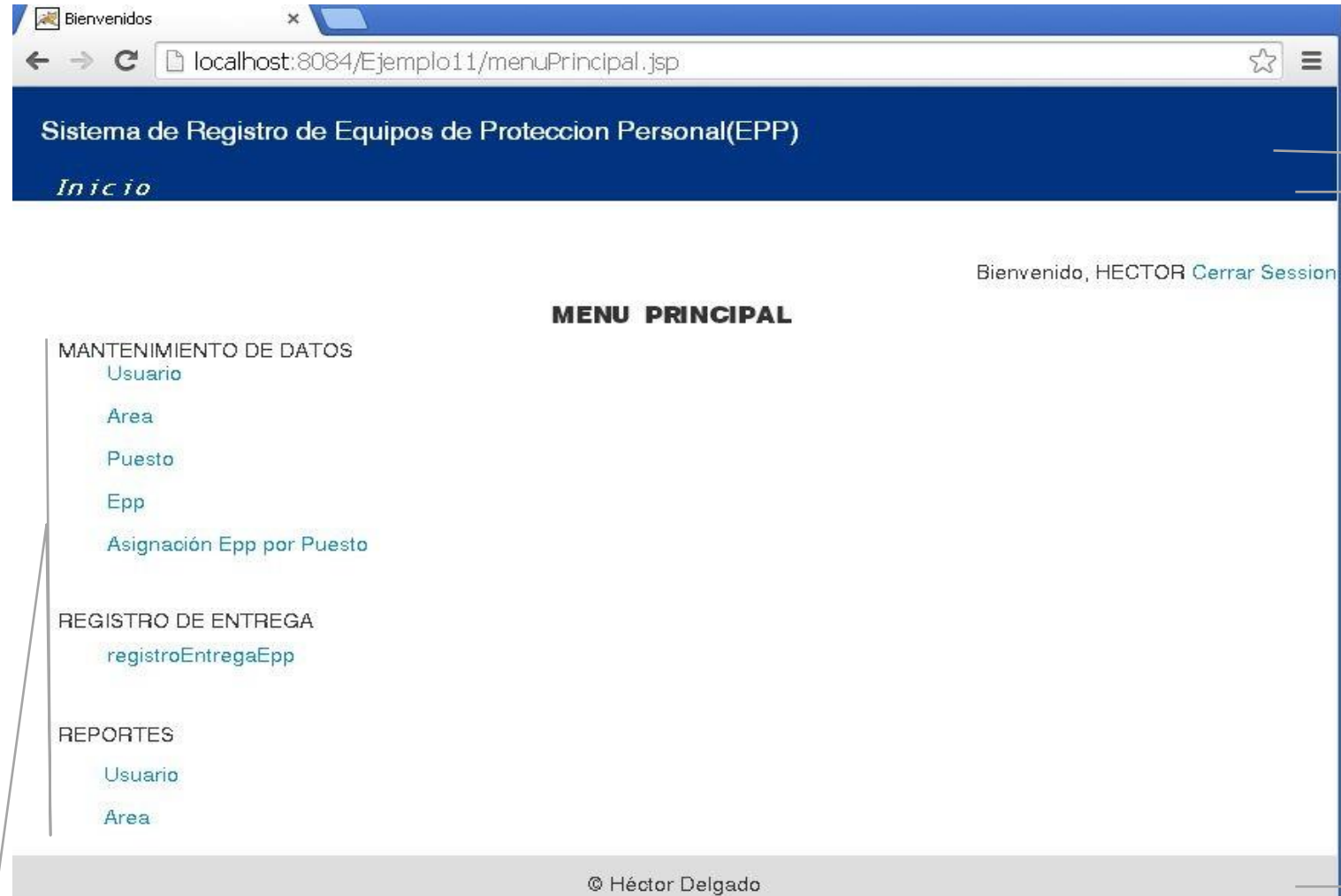
Análisis



The screenshot shows a web browser window with the address bar displaying 'localhost:8084/Ejemplo11/'. The page has a dark blue header with the text 'Sistema de Registro de Equipos de Protección Personal(EPP)'. Below the header, there is a white box with the title 'Ingresa sus datos'. Inside this box, there are two input fields: 'Usuario' and 'Password'. Below these fields is a checkbox labeled 'Recordarme'. At the bottom of the white box is a blue button labeled 'Ingresar'. The footer of the page is a gray bar with the text '© Héctor Delgado'.

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo



Cambiar de menu
(Utilizar un menu personalizado)

Segmentos
(Encabezado/Pie)

14 Modificar el contenido de :

```
menuPrincipal.jsp x
1  <%@page contentType="text/html" session="true" pageEncoding="UTF-8"%>
2  <%@include file="verificaSesion.jsp" %>
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7      <%@include file="WEB-INF/jspf/segmentoEstilos.jspf" %>
8      <title> Bienvenidos</title>
9  </head>
10 <body>
11     <%@include file="WEB-INF/jspf/segmentoSuperior2.jspf" %>
12     <P ALIGN="right">Bienvenido, <%=varNombresMostrar%>
13     <a href="sesionCerrar.jsp" >Cerrar Session</a></P>
14
15     <center><h4>MENU PRINCIPAL</h4></center>
16     MANTENIMIENTO DE DATOS<p>
17     <a href="mantenimientoUsuario.jsp" >Usuario</a><p>
18     <a href="" >Area</a><p>
19     <a href="" >Puesto</a><p>
20     <a href="" >Epp</a><p>
21     <a href="" >Asignación Epp por Puesto</a><p><BR>
22     REGISTRO DE ENTREGA<p>
23     <a href="" >registroEntregaEpp</a><p><BR>
24     REPORTE<p>
25     <a href="" >Usuario</a><p>
26     <a href="" >Area</a>
27     <%@include file="WEB-INF/jspf/segmentoInferior.jspf" %>
28 </body>
29 </html>
```


15 Modificar el contenido de :

```
verificaSesion.jsp x
1  <%@ page session="true" %>
2  <%
3      String varNombresMostrar="";
4      HttpSession varSesion2=request.getSession();
5      if(varSesion2.getAttribute("varNombresSesion")==null){
6          %>
7          <jsp:forward page="index.jsp">
8              <jsp:param name="error" value="Ingreses sus datos"/>
9          </jsp:forward>
10         <%
11     }else{
12         varNombresMostrar="" + session.getAttribute("varNombresSesion");
13     }
14     %>
```

16 Modificar el contenido de :

```
sesionCerrar.jsp x
1  <%
2      session.invalidate();//destruye la session
3      %>
4  <jsp:forward page="index.jsp"/>
```


Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo

The screenshot shows a web browser window with the address bar displaying 'localhost:8084/Ejemplo11/menuPrincipal.jsp'. The page has a dark blue header with the text 'Sistema de Registro de Equipos de Protección Personal(EPP)' and 'Inicio'. Below the header, there is a navigation bar with 'Bienvenido, HECTOR' and a 'Cerrar Session' link. The main content area is titled 'MENU PRINCIPAL' and contains three sections: 'MANTENIMIENTO DE DATOS' with links for 'Usuario', 'Area', 'Puesto', 'Epp', and 'Asignación Epp por Puesto'; 'REGISTRO DE ENTREGA' with a link for 'registroEntregaEpp'; and 'REPORTES' with links for 'Usuario' and 'Area'. A footer bar at the bottom contains the text '© Héctor Delgado'. Two callout boxes are present: one on the left pointing to the menu items with the text 'Cambiar de menu (Utilizar un menu personalizado)' and one on the right pointing to the header and footer with the text 'Segmentos (Encabezado/Pie)'.

Bienvenidos

localhost:8084/Ejemplo11/menuPrincipal.jsp

Sistema de Registro de Equipos de Protección Personal(EPP)

Inicio

Bienvenido, HECTOR Cerrar Session

MENU PRINCIPAL

MANTENIMIENTO DE DATOS

- Usuario
- Area
- Puesto
- Epp
- Asignación Epp por Puesto

REGISTRO DE ENTREGA

- registroEntregaEpp

REPORTES

- Usuario
- Area

© Héctor Delgado

Cambiar de menu
(Utilizar un menu personalizado)

Segmentos
(Encabezado/Pie)

14 Análisis

Sistema de Registro de Equipos

Inicio

MENU PRINCIPAL



Inicio

Nuevo

Usuario

Consultar

Regresar

Inicio

Codigo: 10

Usuario: HECTOR

Password: 123456

Nombres: HECTOR

Apellidos: DELGADO

Regresar

Inicio

Codigo

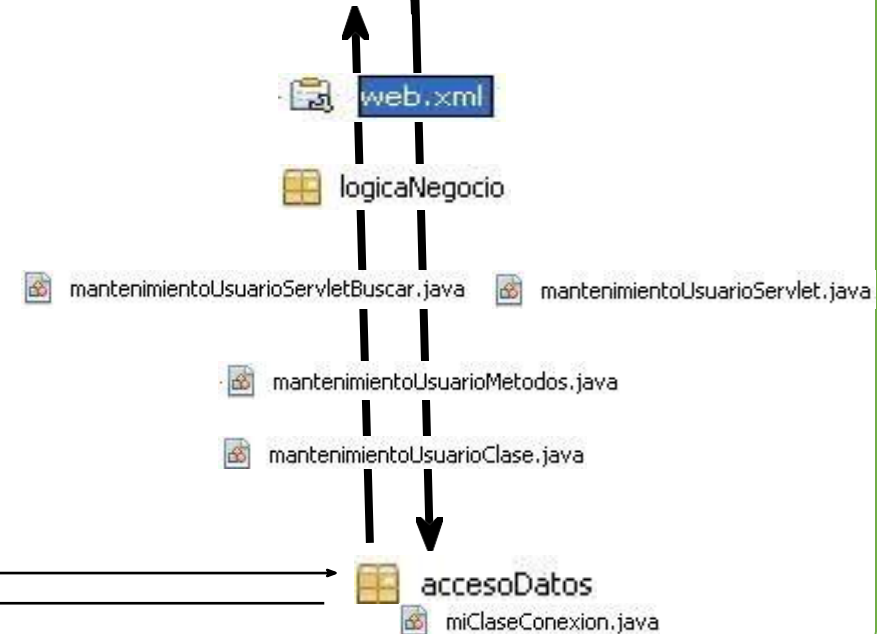
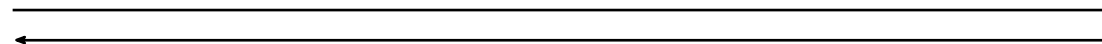
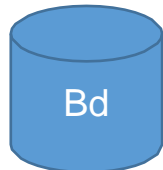
Usuario

Password

Nombres

Apellidos

Guardar Cancelar



Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el **mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

17 Desarrollar el contenido de la página mantenimientoUsuario.jsp

Mantenimiento de datos del usuario **Nuevo**

Bienvenido, HECTOR [Cerrar Session](#)

Mantenimiento de datos del usuario

Ingrese datos del Usuario a buscar: **Consultar**

Regresar


© Héctor Delgado

Muestra los resultados del código
ingresado

Abre el formulario nuevo

17 Desarrollar el contenido de la página mantenimientoUsuario.jsp

```
mantenimientoUsuario.jsp x
1  <%@page contentType="text/html" session="true" pageEncoding="UTF-8"%>
2  <%@include file="verificaSesion.jsp" %>
3  <!DOCTYPE html>
4  <html><head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <%@include file="WEB-INF/jspf/segmentoEstilos.jspf" %>
7      <title>Mantenimiento de datos del usuario</title>
8  </head>
9  <body>
10     <%@include file="WEB-INF/jspf/segmentoSuperior2.jspf" %>
11     <P ALIGN="right">Bienvenido, <%=varNombresMostrar%>
12     <a href="sesionCerrar.jsp" >Cerrar Session</a></P>
13     <div class="ubicacion">
14     <form class="form-signin">
15         Mantenimiento de datos del usuario
16         <a href="mantenimientoUsuarioNuevo.jsp" class="btn btn-small btn-success " >Nuevo</a>
17     </form> <br>
18     <form method="post" action="mantenimientoUsuarioServletBuscar" class="form-signin">
19         Ingrese datos del Usuario a buscar:
20         <input type="text" name="txtUsuarioB" class="input-small" placeholder="Usuario" required autofocus/><p>
21         <input type="submit" class="btn btn-small btn-success " value="Consultar"/>
22     </form> <br><center>
23     <a href="menuPrincipal.jsp" class="btn btn-small btn-danger " >Regresar</a></center>
24 </div>
25     <%@include file="WEB-INF/jspf/segmentoInferior.jspf" %>
26 </body>
27 </html>
```



Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

18 Desarrollar el contenido de la página mantenimientoUsuarioNuevo.jsp

localhost:8084/Ejemplo11/mantenimientoUsuarioNuevo.jsp

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR [Cerrar Session](#)

Ingrese datos del nuevo Usuario

© Héctor Delgado

Bootstrap

18 Desarrollar el contenido de la página mantenimientoUsuarioNuevo.jsp

```
mantenimientoUsuarioNuevo.jsp x
1  <%@page contentType="text/html" session="true" pageEncoding="UTF-8"%>
2  <%@include file="verificaSesion.jsp" %>
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7      <%@include file="WEB-INF/jspf/segmentoEstilos.jspf" %>
8      <title>Mantenimiento de datos del usuario</title>
9  </head>
10 <body>
11     <%@include file="WEB-INF/jspf/segmentoSuperior2.jspf" %>
12     <P ALIGN="right">Bienvenido, <%=varNombresMostrar%>
13     <a href="sesionCerrar.jsp" >Cerrar Session</a></P>
14     <div class="ubicacion">
15         <form method="post" action="mantenimientoUsuarioServlet" class="form-signin">
16             Ingrese datos del nuevo Usuario
17             <input type="text" name="txtCodigo" class="input-small" placeholder="Codigo" required autofocus/><p>
18             <input type="text" name="txtUsuario" class="input-small" placeholder="Usuario" required /><p>
19             <input type="text" name="txtPassword" class="input-small" placeholder="Password" required/><p>
20             <input type="text" name="txtNombres" class="input-small" placeholder="Nombres" required/><p>
21             <input type="text" name="txtApellidos" class="input-small" placeholder="Apellidos" required/><p>
22             <input type="submit" class="btn btn-small btn-success " value="Guardar"/>
23             <a href="mantenimientoUsuario.jsp" class="btn btn-small btn-danger ">Cancelar</a>
24         </form>
25     </div>
26     <%@include file="WEB-INF/jspf/segmentoInferior.jspf" %>
27 </body>
28 </html>
```

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

19 Crear un servlet



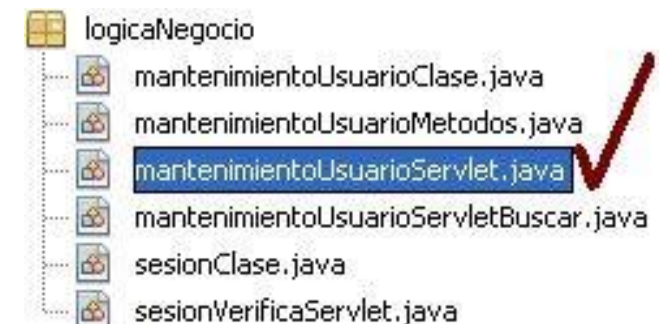
Nombre y ubicación

Nombre de Clase:

[Siguiente >](#)

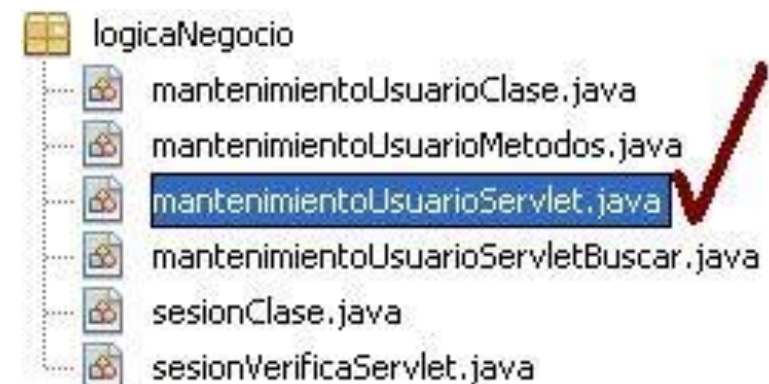
☒ Add information to deployment descriptor (web.xml)

[Terminar](#)



19 Desarrollar el contenido de la página mantenimientoUsuarioServlet

```
mantenimientoUsuarioServlet.java x
1  +  /*...*/
5      package logicaNegocio;
6
7  -  import java.io.IOException;
8      import java.io.PrintWriter;
9      import javax.servlet.ServletException;
10     import javax.servlet.http.HttpServlet;
11     import javax.servlet.http.HttpServletRequest;
12     import javax.servlet.http.HttpServletResponse;
13
14     import javax.servlet.annotation.WebServlet;
15     import logicaNegocio.mantenimientoUsuarioMetodos;
16     import logicaNegocio.mantenimientoUsuarioClase;
17
18  +  /**...*/
```



19 Desarrollar el contenido de la página mantenimientoUsuarioServlet

mantenimientoUsuarioServlet.java x

```
22 public class mantenimientoUsuarioServlet extends HttpServlet {
23     /**...*/
33     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
34         throws ServletException, IOException {
35         response.setContentType("text/html;charset=UTF-8");
36         PrintWriter out = response.getWriter();
37         try {
38             String varCodigo=request.getParameter("txtCodigo");
39             String varUsuario=request.getParameter("txtUsuario");
40             String varPassword=request.getParameter("txtPassword");
41             String varNombres=request.getParameter("txtNombres");
42             String varApellidos=request.getParameter("txtApellidos");
43             if(!varCodigo.equalsIgnoreCase("") && !varUsuario.equalsIgnoreCase("") && !varPassword.equalsIgnoreCase("")
44                 && !varNombres.equalsIgnoreCase("") && !varApellidos.equalsIgnoreCase("")){
45                 mantenimientoUsuarioClase busuario=new mantenimientoUsuarioClase(varCodigo, varUsuario, varPassword, varNombres, varApellidos);
46                 boolean sw=mantenimientoUsuarioMetodos.agregarUsuario(busuario);
47                 if(sw){
48                     request.getRequestDispatcher("mantenimientoUsuario.jsp").forward(request, response);
49                 }else{
50                     //PrintWriter out=response.getWriter();
51                     out.println("Error.");
52                     } } finally {
53                     out.close();
54                 }
55             }
56
57     }
97 }
```

HttpServlet methods. Click on the + sign on the left to edit the code.

logicaNegocio

- mantenimientoUsuarioClase.java
- mantenimientoUsuarioMetodos.java
- mantenimientoUsuarioServlet.java**
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

20 Crear una clase para adicionar los métodos



logicaNe

Nuevo



Java Class...

mantenimientoUsuarioMetodos.java x

```
1  +  /*...*/
5  package logicaNegocio;
6  -  import java.sql.*;
7  L  import accesoDatos.miClaseConexion;
8  +  /**...*/
12 public class mantenimientoUsuarioMetodos {
13  -  public static boolean agregarUsuario(mantenimientoUsuarioClase usuario){
14      boolean agregado=false;
15      try {
16          miClaseConexion c=new miClaseConexion();
17          Connection con=c.getConnection();
18          if(con!=null){
19              Statement st;
20              st = con.createStatement();
21              st.executeUpdate("INSERT INTO tbusuario2(`codigo`,`usuario`,`password`,`nombres`,`apellidos`) "
22                  + "VALUES ('"+usuario.getVarCodigo()+"','"+usuario.getVarUsuario()+"','"+usuario.getVarPassword()+"'
23                  + ",'" +usuario.getVarNombres()+"','"+usuario.getVarApellidos()+"')");
24              agregado=true;
25              st.close();
26          }
27          } catch (SQLException e) {
28              agregado=false;
29              e.printStackTrace();
30          }
31          return agregado;
32      }
33
34  }
```



logicaNegocio

- mantenimientoUsuarioClase.java
- mantenimientoUsuarioMetodos.java** ✓
- mantenimientoUsuarioServlet.java
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

21 Crear una clase para la estructura y construct

logicaNegocio Nuevo Java Class...

mantenimientoUsuarioClase.java

```
1  /*...*/
5  package logicaNegocio;
6
7  /**...*/
11 public class mantenimientoUsuarioClase {
12     private String varCodigo;
13     private String varUsuario;
14     private String varPassword;
15     private String varNombres;
16     private String varApellidos;
17
18 }
19
```

Insertar código... Alt+Insertar

- Generar
- Constructor...
- Logger...
- Getter...
- Setter...

Generar constructor

Seleccione los campos que serán ini...

- ☒ varCodigo : String
- ☐ varUsuario : String
- ☐ varPassword : String
- ☐ varNombres : String
- ☐ varApellidos : String

Generar Cancelar

logicaNegocio

- mantenimientoUsuarioClase.java**
- mantenimientoUsuarioMetodos.java
- mantenimientoUsuarioServlet.java
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

21 Crear una clase para la estructura y constructor

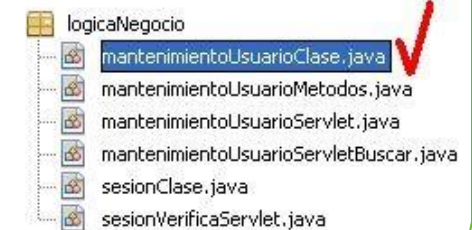
```
mantenimientoUsuarioClase.java x
1  /*...*/
5  package logicaNegocio;
6
7  /**...*/
11 public class mantenimientoUsuarioClase {
12
13     public mantenimientoUsuarioClase () {
14     }
15
16     private String varCodigo;
17     private String varUsuario;
18     private String varPassword;
19     private String varNombres;
20     private String varApellidos;
21
22 }
```

Se creó el constructor vacío

Insertar código... Alt+Insertar

Generar
Constructor...
Logger...
Getter...
Setter...

Insertar los **SET**



21 Crear una clase para la estructura y constructor

mantenimientoUsuarioClase.java x

```
1  /*...*/
5  package logicaNegocio;
6  /**...*/
10 public class mantenimientoUsuarioClase {
11     public mantenimientoUsuarioClase () {
12     }
13     public void setVarCodigo(String varCodigo) {
14         this.varCodigo = varCodigo;
15     }
16     public void setVarUsuario(String varUsuario) {
17         this.varUsuario = varUsuario;
18     }
19     public void setVarPassword(String varPassword) {
20         this.varPassword = varPassword;
21     }
22     public void setVarNombres(String varNombres) {
23         this.varNombres = varNombres;
24     }
25     public void setVarApellidos(String varApellidos) {
26         this.varApellidos = varApellidos;
27     }
28     private String varCodigo;
29     private String varUsuario;
30     private String varPassword;
31     private String varNombres;
32     private String varApellidos;
33 }
```

Se creó el Código set

Insertar los **GET**



Generar
Constructor...
Logger...
Getter... ✓

Insertar código... Alt+Insertar

logicaNegocio
mantenimientoUsuarioClase.java ✓
mantenimientoUsuarioMetodos.java
mantenimientoUsuarioServlet.java
mantenimientoUsuarioServletBuscar.java
sesionClase.java
sesionVerificaServlet.java

21 Observar como quedaría el código:

mantenimientoUsuarioClase.java x

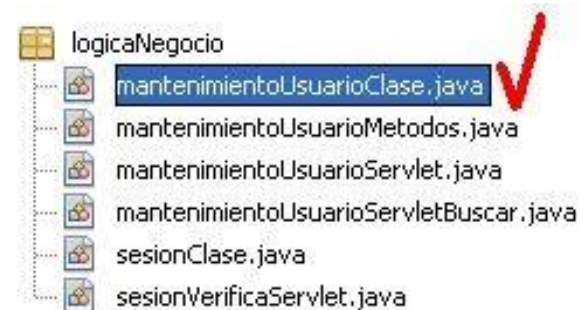
```
1  +  /*...*/
5  package logicaNegocio;
6  +  /**...*/
10 public class mantenimientoUsuarioClase {
11  -  public mantenimientoUsuarioClase() {
12      this.varCodigo="";
13      this.varUsuario="";
14      this.varNombres="";
15      this.varApellidos="";
16  }
17  -  public mantenimientoUsuarioClase(String varCodigo, String varUsuario, String varPassword, String varNombres, String varApellidos) {
18      this.varCodigo = varCodigo;
19      this.varUsuario = varUsuario;
20      this.varPassword = varPassword;
21      this.varNombres = varNombres;
22      this.varApellidos = varApellidos;
23  }
24  -  public void setVarCodigo(String varCodigo) {
25      this.varCodigo = varCodigo;
26  }
27  -  public void setVarUsuario(String varUsuario) {
28      this.varUsuario = varUsuario;
29  }
30  -  public void setVarPassword(String varPassword) {
31      this.varPassword = varPassword;
32  }
```

logicaNegocio

- mantenimientoUsuarioClase.java ✓
- mantenimientoUsuarioMetodos.java
- mantenimientoUsuarioServlet.java
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

21 Observar como quedaría el código:

```
mantenimientoUsuarioClase.java x
33 public void setVarNombres(String varNombres) {
34     this.varNombres = varNombres;
35 }
36 public void setVarApellidos(String varApellidos) {
37     this.varApellidos = varApellidos;
38 }
39 public String getVarCodigo() {
40     return varCodigo;
41 }
42 public String getVarUsuario() {
43     return varUsuario;
44 }
45 public String getVarPassword() {
46     return varPassword;
47 }
48 public String getVarNombres() {
49     return varNombres;
50 }
51 public String getVarApellidos() {
52     return varApellidos;
53 }
54 private String varCodigo;
55 private String varUsuario;
56 private String varPassword;
57 private String varNombres;
58 private String varApellidos;
59 }
```



22 Análisis

Sistema de Registro de Equipos

Inicio

MENU PRINCIPAL



Inicio

Nuevo

Usuario

Consultar

Regresar

Inicio

Codigo:

Usuario:

Password:

Nombres:

Apellidos:

Regresar

Inicio

Guardar Cancelar

web.xml

logicaNegocio

mantenimientoUsuarioServletBuscar.java

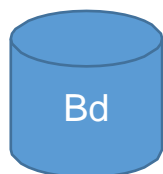
mantenimientoUsuarioServlet.java

mantenimientoUsuarioMetodos.java

mantenimientoUsuarioClase.java

accesoDatos

miClaseConexion.java



Bd

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita el **mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

22 Buscar datos

The screenshot shows a web browser window with the address bar displaying `localhost:8084/Ejemplo11/mantenimientoUsuario.jsp`. The page has a dark blue header with the text "Sistema de Registro de Equipos de Proteccion Personal(EPP)" and "Inicio". Below the header, there is a greeting "Bienvenido, HECTOR" followed by a link "Cerrar Session". The main content area contains two white boxes. The first box has the text "Mantenimiento de datos del usuario" and a green button labeled "Nuevo". The second box has the text "Ingrese datos del Usuario a buscar:" followed by a text input field containing the word "Usuario" and a green button labeled "Consultar". At the bottom of the main content area, there is a red button labeled "Regresar". The footer of the page is a grey bar with the text "© Héctor Delgado".

Bootstrap

22 Crear un SERVLET con el nombre mantenimientoUsuarioServletBus

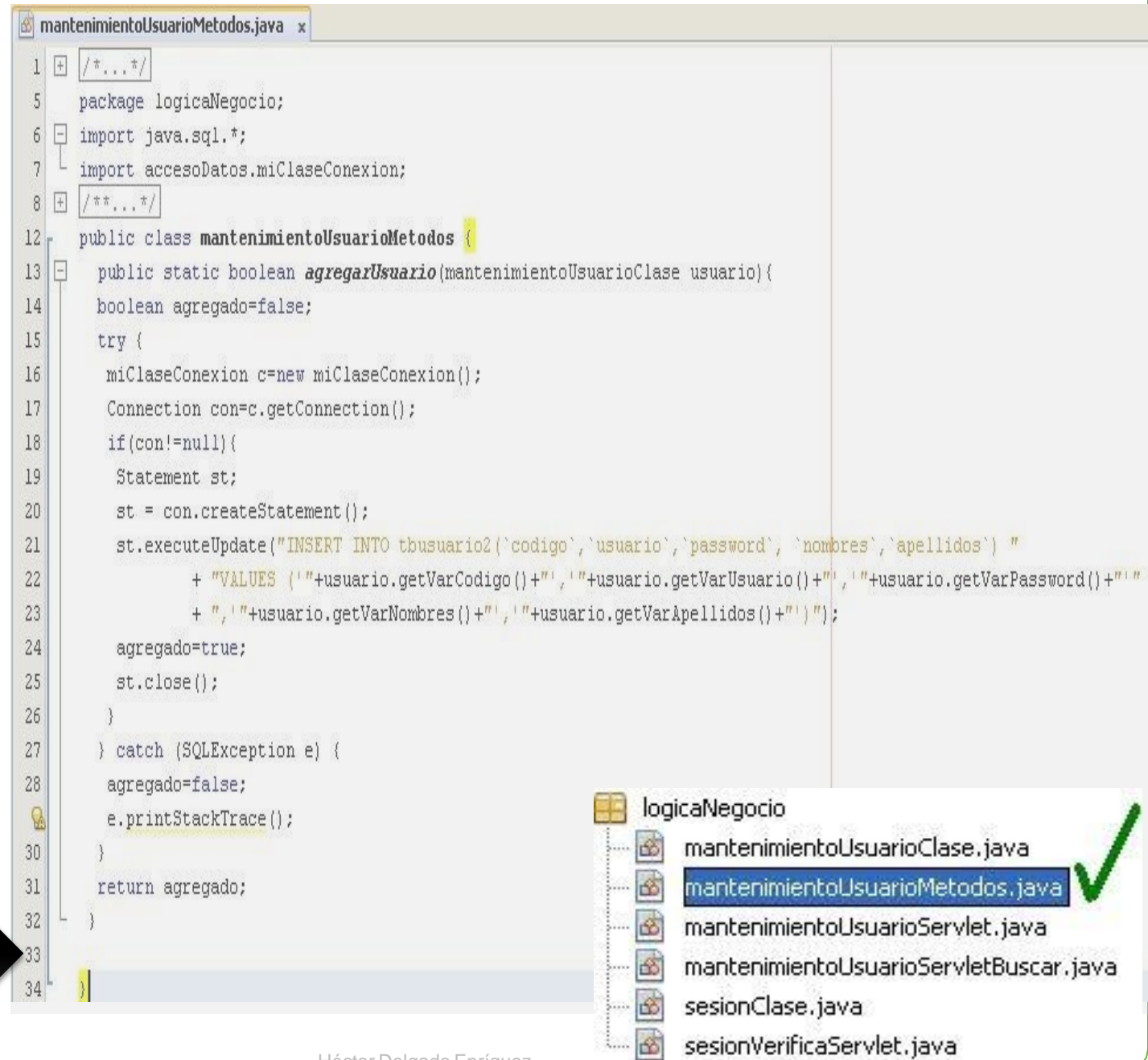
logicaNegocio

Nuevo

Servlet...

```
mantenimientoUsuarioServletBuscar.java x
1  /*...*/
5  package logicaNegocio;
6  import java.io.IOException;
7  import java.io.PrintWriter;
8  import javax.servlet.ServletException;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import javax.servlet.annotation.WebServlet;
13 import logicaNegocio.mantenimientoUsuarioMetodos;
14 import logicaNegocio.mantenimientoUsuarioClase;
15 /**...*/
19 public class mantenimientoUsuarioServletBuscar extends HttpServlet {
20     /**...*/
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         PrintWriter out = response.getWriter();
34         try {
35             String usuario=request.getParameter("txtUsuarioB");
36             mantenimientoUsuarioClase busuario=mantenimientoUsuarioMetodos.consultarUsuario(usuario);
37             if(busuario!=null){
38                 request.setAttribute("busuario", busuario);
39                 request.getRequestDispatcher("mantenimientoUsuarioBuscar.jsp").forward(request, response);
40             }else{
41                 out.println("Error, no se encontro el usuario.");
42             } } finally { out.close(); }
43     }
83 }
```

22 Adicionamos el método consultarUsuario del archivo mantenimientoUsuarioMetodos, en la línea 33



```
1  +  /*...*/
5  package logicaNegocio;
6  -  import java.sql.*;
7  -  import accesoDatos.miClaseConexion;
8  +  /**...*/
12 public class mantenimientoUsuarioMetodos {
13 -  public static boolean agregarUsuario(mantenimientoUsuarioClase usuario){
14     boolean agregado=false;
15     try {
16         miClaseConexion c=new miClaseConexion();
17         Connection con=c.getConnection();
18         if(con!=null){
19             Statement st;
20             st = con.createStatement();
21             st.executeUpdate("INSERT INTO tbusuario2(`codigo`,`usuario`,`password`,`nombres`,`apellidos`) "
22                 + "VALUES ('"+usuario.getVarCodigo()+"','"+usuario.getVarUsuario()+"','"+usuario.getVarPassword()+"'
23                 + "','"+usuario.getVarNombres()+"','"+usuario.getVarApellidos()+"')");
24             agregado=true;
25             st.close();
26         }
27     } catch (SQLException e) {
28         agregado=false;
29         e.printStackTrace();
30     }
31     return agregado;
32 }
33
34 }
```

logicaNegocio

- mantenimientoUsuarioClase.java
- mantenimientoUsuarioMetodos.java** ✓
- mantenimientoUsuarioServlet.java
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

22 Adicionamos el método consultarUsuario del archivo mantenimientoUsuarioMetodos, en la línea 33

```
mantenimientoUsuarioMetodos.java x
32  }
33  public static mantenimientoUsuarioClase consultarUsuario(String usuario){
34      mantenimientoUsuarioClase busuario=null;
35      try{
36          miClaseConexion c=new miClaseConexion();
37          Connection con=c.getConnection();
38          Statement st=con.createStatement();
39          ResultSet rs=st.executeQuery("Select * from tbusuario2 where usuario='"+usuario+"'");
40          while(rs.next()){
41              busuario=new mantenimientoUsuarioClase(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5));
42              //rs.getInt(3)
43              rs.close();
44              st.close();
45          }catch(SQLException se){
46              se.printStackTrace();
47          }
48          return busuario;
49      }
50  }
```

logicaNegocio

- mantenimientoUsuarioClase.java
- mantenimientoUsuarioMetodos.java** ✓
- mantenimientoUsuarioServlet.java
- mantenimientoUsuarioServletBuscar.java
- sesionClase.java
- sesionVerificaServlet.java

Ejemplo 11 – Mantenimiento Registros/Capas

Desarrollar una página que permita **el mantenimiento de registros usando Servlet, Capas, sesiones, bootstrap, segmentos jspf y hojas de estilo**

22 Buscar datos

Sistema de Registro de Equipos de Proteccion Personal(EPP)

Inicio

Bienvenido, HECTOR [Cerrar Session](#)

Codigo:

Usuario:

Password:

Nombres:

Apellidos:

© Héctor Delgado

Bootstrap

22 Crear un JSP con el nombre mantenimientoUsuarioBuscar

mantenimientoUsuarioBuscar.jsp

```
1  <%@page import="logicaNegocio.mantenimientoUsuarioClase"%>
2  <%@page contentType="text/html" session="true" pageEncoding="UTF-8"%>
3  <%@include file="verificaSesion.jsp" %>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8      <%@include file="WEB-INF/jspf/segmentoEstilos.jspf" %>
9      <title>Busqueda de datos del usuario</title>
10 </head>
11 <body>
12     <%@include file="WEB-INF/jspf/segmentoSuperior2.jspf" %>
13     <p ALIGN="right">Bienvenido, <%=varNombresMostrar%>
14     <a href="sesionCerrar.jsp" >Cerrar Session</a></p>
15     <div class="ubicacion">
16         <%mantenimientoUsuarioClase busuario=(mantenimientoUsuarioClase)request.getAttribute("busuario"); %>
17     <div align="center">
18         <table>
19             <tr><td>Codigo: </td><td><input type="text" size="20" name="nombre" value="<%= busuario.getVarCodigo() %>"> </td></tr>
20             <tr><td>Usuario: </td><td><input type="text" size="20" name="apellido" value="<%= busuario.getVarUsuario() %>"> </td></tr>
21             <tr><td>Password: </td><td><input type="text" size="20" name="edad" value="<%= busuario.getVarPassword() %>"> </td></tr>
22             <tr><td>Nombres: </td><td><input type="text" size="20" name="correo" value="<%= busuario.getVarNombres() %>"> </td></tr>
23             <tr><td>Apellidos: </td><td><input type="text" size="20" name="usuario" value="<%= busuario.getVarApellidos() %>"> </td></tr>
24         </table>
25     </div><br><center><a href="mantenimientoUsuario.jsp" class="btn btn-small btn-danger ">Regresar</a></center></div>
26     <%@include file="WEB-INF/jspf/segmentoInferior.jspf" %>
27 </body>
28 </html>
```

22 Análisis

Sistema de Registro de Equipos

Inicio

MENU PRINCIPAL



Inicio

Nuevo

Usuario

Consultar

Regresar

Inicio

Codigo: 10

Usuario: HECTOR

Password: 123456

Nombres: HECTOR

Apellidos: DELGADO

Regresar

Inicio

Codigo

Usuario

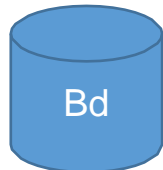
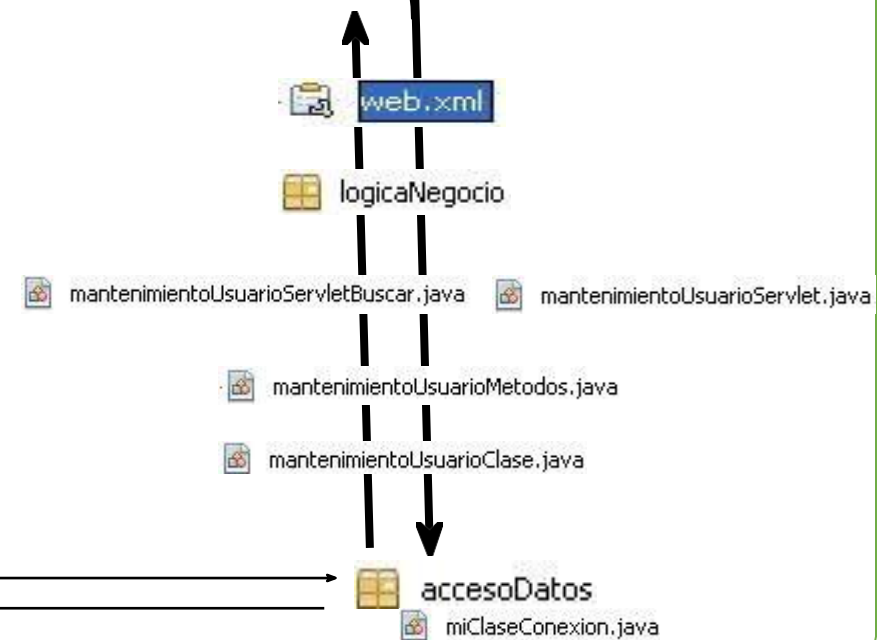
Password

Nombres

Apellidos

Guardar

Cancelar



Actividad Grupal

1.- Desarrollar el ejercicio guiado:

-Mantenimiento de Registros, usando

**Capas, servlet, sesiones, bootstrap,
segmentos jspf y hojas de estilo**



**Universidad
Tecnológica
del Perú**

- ❖ Una página que permita el ingreso de un usuario mediante **CAPAS**, sesiones, los datos del usuario deberán de estar almacenados en una tabla con la siguiente estructura:

- | | | |
|--|--|---------------------|
| • ItemAi(Auto incrementable) | • Elinea(Indica si esta en sesión o no - uno o | • Hora_Creacion |
| • IdUsuario(Correlativo desde 10000000x) | cero) | • Hora_Modificacion |
| • CodUsuario | • Num_Ingresos(Cantidad de ingresos a la | • Hora_Eliminacion |
| • Usuario | fecha) | • Hora_UltimoAcceso |
| • Password | • Fec_Creacion | |
| • Nombres | • Fec_Eliminacion | |
| • Apellidos | • Fec_Modificacion | |
| • Email | • Fec_UltimoAcceso | |
| • Permisos(Administrador/Usuario normal) | • Creado_Por | |
| • Estado(Activo-desactivo "Eliminado"- | • Modificado_Por | |
| Bloqueado) | • Eliminado_Por | |

*Adicionar más campos en caso de ser necesario (Nombre de la Pc, ip, lugar, ciudad, etc)

Validaciones:

- Deberá salir un mensaje de ERROR en caso:
 - El usuario sea incorrecto
 - La clave sea incorrecta
 - El usuario y clave sean incorrectos
- Deberá CERRAR la ventana ACTIVA en caso se equivoque 3 veces
- Deberá bloquearse el usuario en caso se equivoque 3 veces (solo el administrador puede desbloquearlo desde la pagina mantenimiento de usuarios)
- Deberá de estar conectado a una tabla y base de datos
- Deberá controlar los permisos de usuario
 - Si es **administrador** mostrara en el menú principal la opción MANTENIMIENTO DE USUARIOS
 - Si es **Usuario normal NO** mostrara en el menú principal la opción MANTENIMIENTO DE USUARIOS
- Cuando ingrese el usuario correcto el campo ENLINEA, deberá mostrar el valor 1 en la tabla(activo) y cero cuando no haya iniciado sesión.