

Sesiones y Permisos

I- Objetivos

- Implementar la lógica de negocio en programas de inicio de sesión
- Crear una estructura de datos que permita administrar usuarios.
- Establecer opciones con permisos para usuarios de la aplicación

II- Contenido

Introducción

El desarrollo de aplicaciones requiere que se dedique tiempo para el manejo de usuarios y permisos que estos tendrán con respecto a las operaciones que el sistema ejecutará. De manera que los accesos se manejen en función de perfiles previamente definidos por el administrador de la aplicación. Esto se traduce en la creación de tablas dentro de la base de datos que almacenaran información sobre los usuarios, roles, opciones del menú y permisos que los usuarios tendrán con respecto a las opciones.

Por otro lado el manejo de las operaciones con la base de datos (Lógica de Negocio), debe estructurarse de manera que satisfaga las exigencias de los procesos.

Nota: para continuar deberá estar la base de datos aerolínea proporcionada. Y el proyecto llamado Aerolínea, el mismo creado en la guía anterior con las clases funcionando.

Bugs guía anterior

Corregir clase Persistencia.java agregando las siguientes líneas marcadas. El error ocurre en resultsets vacíos

```
String [][] resultados=null;
try {
    PreparedStatement stmt =
    conexion.getConnection().prepareStatement(sqlQuery,
    ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
    ...
    if(rs.next()) {
        rs.last();
        ResultSetMetaData rsmd = rs.getMetaData();
        int numCols = rsmd.getColumnCount();
        int numFils =rs.getRow();
        resultados=new String[numCols][numFils];
        int j = 0;
        rs.beforeFirst();
```

```
while (rs.next())
{
    for (int i=0;i<numCols;i++)
    {
        resultados[i][j]=rs.getString(i+1);
    }
    j++;
}
return resultados;
}
return resultados;
```

Tablas de menú y permisos

```
create table menu(
    idmenu int NOT NULL identity,
    menu varchar(50) NOT NULL,
    descripcion varchar(100),
    url varchar(300),
    idpadre int,
    PRIMARY KEY(idmenu),
    FOREIGN KEY (idpadre) REFERENCES menu (idmenu)
);
create table permiso(
    idpermiso int NOT NULL identity,
    idmenu int NOT NULL,
    idrol int NOT NULL,
    PRIMARY KEY (idpermiso),
    FOREIGN KEY (idmenu) REFERENCES menu (idmenu),
    FOREIGN KEY (idrol) REFERENCES rol (idrol)
);
```

Insertar datos de prueba

Tabla: rol

idrol	rol
1	Admin
2	User

Tabla: país

idpais	pais
1	El Salvador
2	Estados Unidos

Tabla: Usuarios

Idusuario	Nombres	Apellidos	Correo	Teléfono	Contraseña	idpais	idrol
admin	Alicia	Perez	alicia@gmail.com	7777 7771	admin	1	1
beto	Beto	Perez	beto@gmail.com	7777 7772	beto	1	2

Ejemplo:

```
INSERT INTO USUARIO VALUES('admin','Alicia','Perez','alicia@usonsonate.edu.sv', '7777
7771', lower(CONVERT(VARCHAR(64),HashBytes('SHA2_256', 'admin'),2)), 1,1);
```

Tabla: menu

Idmenu	Menu	Descripcion	url	idpadre
1	Inicio		/Principal	
2	Usuarios		/Principal	
3	Aeropuertos		/Principal	
4	Aviones		/Principal	
5	Vuelos		/Principal	
6	Países		/Principal	
7	Roles		/Rol	2
8	Menu		/Menu	2
9	Permisos		/Permisos	2
10	Registrarse		/Usuarios	2
11	Reservaciones		/Reservaciones	5
12	Registro vuelos		/Vuelos	5

Tabla: Permisos

Rol: admin {1,2,3,4,5,6,7,8,9,10,11,12}

Rol: User {1,2, 5, 10, 11}

SQLs

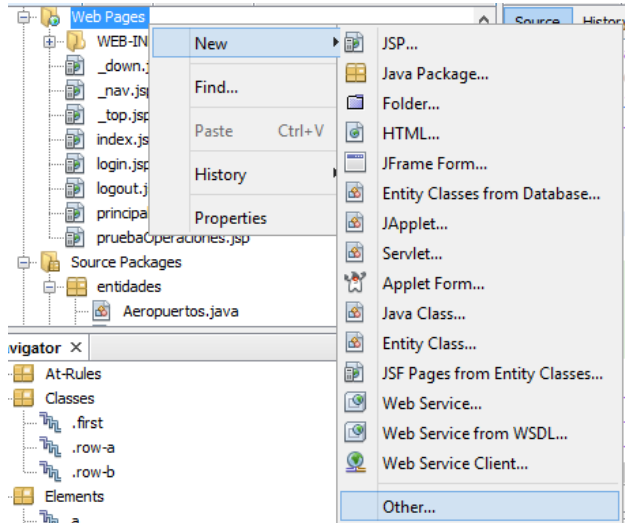
```

set identity_insert menu on;
insert into menu(idmenu, menu, descripcion, url, idpadre) values(1, 'Inicio', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(2, 'Usuarios', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(3, 'Aeropuertos', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(4, 'Aviones', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(5, 'Vuelos', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(6, 'Países', '',
'/Principal', null);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(7, 'Roles', '',
'/Rol', 2);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(8, 'Menu', '',
'/Menu', 2);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(9, 'Permisos', '',
'/Permisos', 2);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(10, 'Registrarse', '',
'/Usuarios', 2);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(11, 'Reservaciones',
'', '/Reservaciones', 5);
insert into menu(idmenu, menu, descripcion, url, idpadre) values(12, 'Registro Vuelos',
'', '/Vuelos', 5);
insert into permiso(idmenu, idrol) select idmenu, 1 from menu;
insert into permiso(idmenu, idrol) select idmenu, 2 from menu where idmenu in
(1,2,5,10,11);

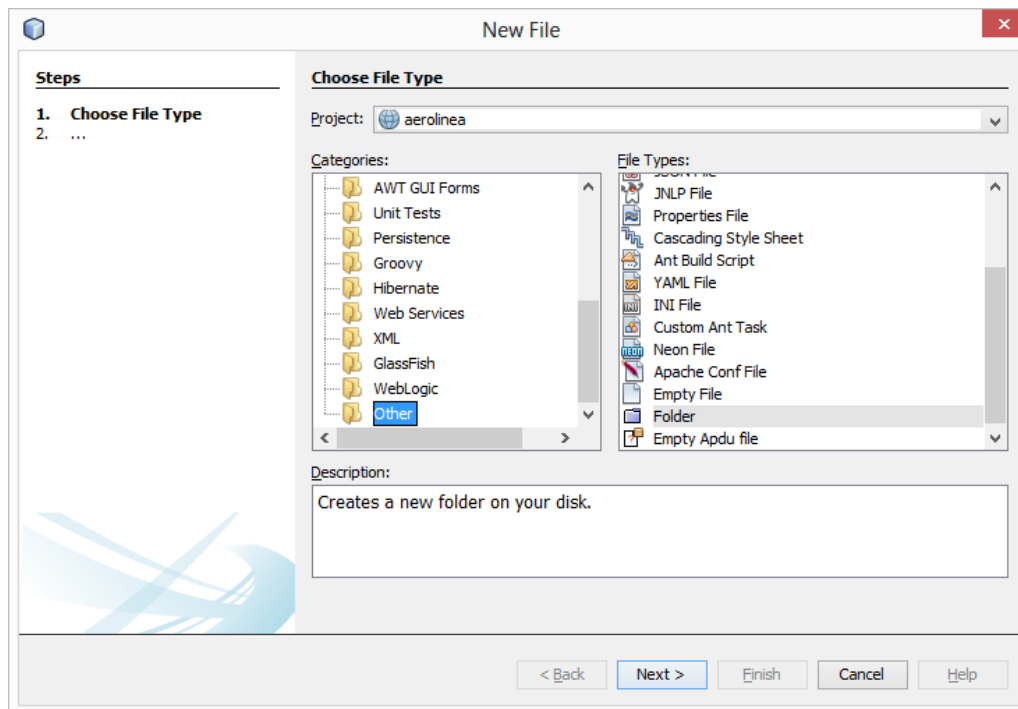
```

Preparando el proyecto.

Se creara dos carpetas en **Web Pages** para agregar los recursos proporcionados para el estilo de la página.



En la ventana que aparece clic en otros y buscar Folder.



Las 3 carpetas a crear serán **css**, **iconos** y **font** en las cuales agregaremos los archivos proporcionados en el aula virtual.

Página Inicio. index.jsp

Cuando se crea un sitio web la primera página que se muestra, por lo general, es la llamada Index sin importar la extensión que esta tenga. En el caso de nuestro proyecto deberá tener la extensión jsp por que tendrá incluido en el lenguaje HTML un poco de JSP.

index.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>Aerolinea Project</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css" />
    <link rel="stylesheet" type="text/css" href="css/main.css" media="screen" />
    <style type="text/css">
      .login {margin: 0 auto}
      .login td{padding:5px 0px}
      #header >h1{margin-left: 0px;padding-top: 110px;}
      #content >h1{text-align: center}
    </style>
  </head>
  <body>
    <div id="header">
      <h1>Aerolínea DSi</h1>
    </div>
    <div id="content">
      <hr/>
      <h1>Inicio de Sesión</h1><br/>
      <center>
        <c:if test="${error!=null}">
          <c:if test="${error==2}">
            <p><strong style="color: red">Usuario y/o contraseña incorrectos</strong></p>
          </c:if>
        </c:if>
      </center>

      <form name="main" action="Login?accion=login" method="POST">
        <table class="login">
          <tr><td>Usuario</td></tr>
          <tr><td><input type="text" name="txtUsuario" size="30px"/></td></tr>
          <tr><td>Contraseña</td></tr>
          <tr><td><input type="password" name="txtClave" size="30px"/></td></tr>
          <tr><td>
            <div class="buttons">
              <ul>
                <li><input type="submit" value="Entrar" name="btnEntrar"/></li>
              </ul>
            </div>
          </td></tr>
        </table>
      </form>
      <br/><br/><br/><br/>
    </div>
  </body>
</html>
```

Servlet: Login

Ya creada la página de inicio crearemos el controlador que se encargará de verificar que el usuario y contraseña son correctos. **Antes debemos mapear las tablas: usuario y menú, por lo menos.**

Creamos un servlet llamado **Login.java** y agregamos el código:

```

1  package com.aerolinea.control;
2
3  import com.aerolinea.conexion.*;
4  import com.aerolinea.entidad.Menu;
5  import com.aerolinea.entidad.Usuario;
6  import com.aerolinea.operaciones.Operaciones;
7  import com.aerolinea.utilerias.Hash;
8  import java.io.IOException;
9  import java.sql.SQLException;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import javax.servlet.ServletException;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16 import javax.servlet.http.HttpSession;
17 import java.io.PrintWriter;
18 import java.util.ArrayList;
19 import java.util.List;
20 import javax.servlet.annotation.WebServlet;
21
22 @WebServlet(name = "Login", urlPatterns = {"/Login"})
23 public class Login extends HttpServlet {
24
25     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
26         throws ServletException, IOException {
27         String accion = request.getParameter("accion");
28         if (accion == null) {
29             request.getRequestDispatcher("index.jsp").forward(request, response);
30         } else if (accion.equals("login")) {
31             try {
32                 iniciarSesion(request, response);
33             } catch (SQLException ex) {
34             }
35         }
36     }
37
38     private void iniciarSesion(HttpServletRequest request, HttpServletResponse response) throws IOException, SQLException {
39         String usuario = request.getParameter("txtUsuario");
40         String clave = request.getParameter("txtClave");
41         PrintWriter io = response.getWriter();
42         if (usuario == null) {
43             usuario = "";
44         }
45         if (clave == null) {
46             clave = "";
47         }
48         try {
49             Conexion conn = new ConexionPool();
50             conn.conectar();
51             Operaciones.abrirConexion(conn);
52             Operaciones.iniciarTransaccion();
53             if (conn.getConnection() == null) {
54                 request.setAttribute("error", 1);
55                 request.getRequestDispatcher("index.jsp").forward(request, response);
56             } else {
57                 HttpSession session = request.getSession();
58                 Usuario u = Operaciones.get(usuario, new Usuario());

```


```

57         if (u.getIdUsuario() != null) {
58             if (u.getIdClave().equals(Hash.generarHash(clave, Hash.SHA256))) {
59                 session.setAttribute("Usuario", u.getIdUsuario());
60                 session.setAttribute("Nombre", u.getIdNombres() + " " +
61                     u.getIdApellidos());
62                 session.setAttribute("Rol", u.getIdRol());
63                 List<Menu> permisos = getPermisos(u.getIdRol());
64                 session.setAttribute("Permisos", permisos);
65                 response.sendRedirect("Principal");
66             } else {
67                 request.setAttribute("error", 2);
68                 request.getRequestDispatcher("index.jsp").forward(request, response);
69             }
70         } else {
71             request.setAttribute("error", 2);
72             request.getRequestDispatcher("index.jsp").forward(request, response);
73         }
74     }
75     Operaciones.commit();
76 } catch (Exception ex) {
77     Operaciones.rollback();
78     io.print(ex);
79 } finally {
80     try {
81         Operaciones.cerrarConexion();
82     } catch (SQLException ex) {
83         Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
84     }
85 }
86
87 private List<Menu> getPermisos(Integer idrol) throws SQLException {
88     List<Menu> permisos = new ArrayList();
89     try {
90         String sql = "select * from menu where idmenu in (select idmenu from permiso where idrol = ?)";
91         List<Object> param = new ArrayList();
92         param.add(idrol);
93         String[][] rs = Operaciones.consultar(sql, param);
94         for (int i = 0; i < rs[0].length; i++) {
95             Menu m = new Menu(Integer.parseInt(rs[0][i]), rs[1][i], rs[2][i], rs[3][i], Integer.parseInt(rs[4][i]) == null ? "0" : rs[4][i]);
96             permisos.add(m);
97         }
98     } catch (Exception ex) {
99         Operaciones.rollback();
100     }
101     return permisos;
102 }
103
104 @Override
105 protected void doGet(HttpServletRequest request, HttpServletResponse response)
106     throws ServletException, IOException {
107     processRequest(request, response);
108 }
109
110 @Override
111 protected void doPost(HttpServletRequest request, HttpServletResponse response)
112     throws ServletException, IOException {
113     processRequest(request, response);
114 }
115
116 }
117

```

Agregar archivo web.xml

Dentro establecer el punto de entrada del proyecto, en este caso el servlet Login, para ello agregamos las siguientes líneas al web.xml.



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
4      <session-config>
5          <session-timeout>
6              30
7          </session-timeout>
8      </session-config>
9      <welcome-file-list>
10         <welcome-file>Login</welcome-file>
11     </welcome-file-list>
12 </web-app>

```

Página principal

La página principal estará diseñada, para reutilizar, por una página llamada **_top.jsp** y **_down.jsp** que tendrán el código de Encabezado y menú, y el pie de página respectivamente. Y un archivo llamado **principal.jsp** que se encargará de incluir los dos archivos.

Código _top.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8" session="true"%>
<% HttpSession sesion = request.getSession(); %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Aerolinea Project</title>
        <link rel="stylesheet" type="text/css" href="css/reset.css" />
        <link rel="stylesheet" type="text/css" href="css/main.css"
media="screen" />
    </head>
    <body>
        <div id="header">
            <h1>Aerolínea DSi</h1><br/>
            <h2>A Example Project for Learning</h2>
            
        </div>
        <div id="sesion">
            <h2>Usuario: <%= sesion.getAttribute("Nombre") %>
                <strong><%= sesion.getAttribute("Usuario") %></strong> |
                <a href="Principal?accion=logout">Cerrar Sesión</a>
            </h2>
        </div>
        <div id="menu">
            <ul>
                <c:forEach var="menu" items="${MenuPrincipal}">
                    <li><a
href="${pageContext.servletContext.contextPath}${menu.url}?op=${menu.idmenu}">${menu.menu}</a></li>
                </c:forEach>
            </ul>
        </div>
        <div id="content">
```

Código _down.jsp

```
</div>
    <div id="footer">
        <hr/>
        <br/>
        <p>
            <a href="">Home</a> |
            <a href="">Usuarios</a> |
            <a href="">Aeropuertos</a> |
```



```

        <a href="">Aviones</a> |
        <a href="">Países</a> |
        <a href="">Roles</a> |
        <a href="">Vuelos</a>

    </p>
    <br/>
    <p>&copy; Copyright 2018 | Design by: | Developed by: </p>
</div>
</body>
</html>

```

Código principal.jsp

```

1  <%%taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
2  <%%include file="_top.jsp" %>
3  <style>
4      #opciones li a{
5          text-decoration: none;
6          font-size: 14px
7      }
8  </style>
9  <ul id="opciones">
10     <c:forEach var="opcion" items="{PermisosAsignados}">
11         <li><a href="{pageContext.servletContext.contextPath}${opcion.url}?op={opcion.idmenu}">${opcion.menu}</a></li>
12     </c:forEach>
13 </ul>
14 <%%include file="_down.jsp" %>
15

```

Principal.java

Este Servlet se encargará de cerrar la sesión cuando el usuario así lo desea y otros eventos relacionados con el menú. Escuchar indicaciones.

```

1  package com.aerolinea.control;
2
3  import com.aerolinea.entidad.Menu;
4  import java.io.IOException;
5  import java.io.PrintWriter;
6  import java.util.List;
7  import java.util.stream.Collectors;
8  import javax.servlet.ServletException;
9  import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15 @WebServlet(name = "Principal", urlPatterns = {"/Principal"})
16 public class Principal extends HttpServlet {
17
18     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         PrintWriter io = response.getWriter();
21         String accion = request.getParameter("accion");
22         if (accion == null) {
23             HttpSession s = request.getSession();
24             List<Menu> per = (List<Menu>) s.getAttribute("Permisos");
25             List<Menu> MenuPrincipal = per.stream().filter(field -> field.getIdpadre() == 0).collect(Collectors.toList());
26             request.setAttribute("MenuPrincipal", MenuPrincipal);
27             String op = request.getParameter("op");
28             if (op != null) {
29                 List<Menu> PermisosAsignados = per.stream().filter(field -> field.getIdpadre() == Integer.parseInt(op)).collect(Collectors.toList());
30                 request.setAttribute("PermisosAsignados", PermisosAsignados);
31             }
32             request.getRequestDispatcher("principal.jsp").forward(request, response);
33         } else if (accion.equals("logout")) {
34             logout(request, response);
35         }
36     }
37

```

```
38 private void logout(HttpServletRequest request, HttpServletResponse response) throws IOException{
39     HttpSession session = request.getSession();
40     session.removeAttribute("Usuario");
41     session.removeAttribute("Nombre");
42     session.removeAttribute("Rol");
43     session.invalidate();
44     response.sendRedirect("Login");
45 }
46
47 @Override
48 protected void doGet(HttpServletRequest request, HttpServletResponse response)
49     throws ServletException, IOException {
50     processRequest(request, response);
51 }
52
53 @Override
54 protected void doPost(HttpServletRequest request, HttpServletResponse response)
55     throws ServletException, IOException {
56     processRequest(request, response);
57 }
58 }
59
```

III- Tarea

1. Crear el formulario para el registro del usuario sólo permitirá registrar usuarios con rol 1 y 2.
2. Validar la página principal para que al cerrar la sesión o al no estar logeado no permita mostrarla así mismo al querer ir hacia atrás no muestre la página, sino la de validación.
3. Validar url por rol denegando el recurso
4. Modificar el menú de para que funcione con N niveles, actualmente funciona con N=2