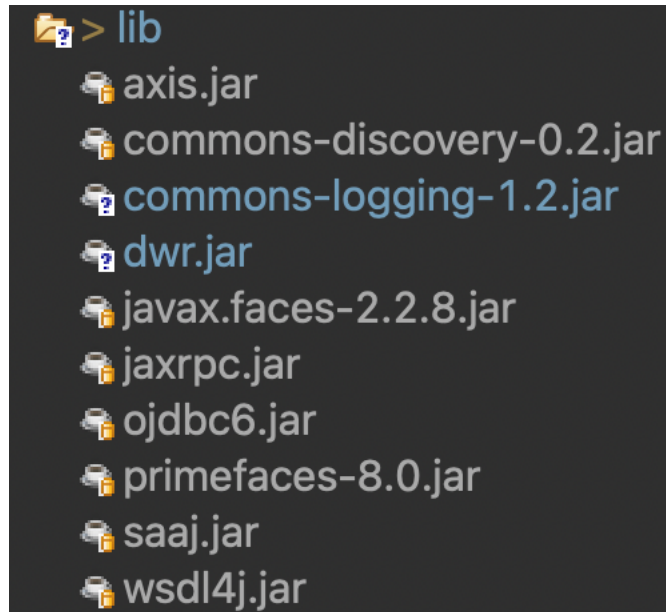


Nombre: Alejandro Herrera Mendoza

Descripción: Se agregó una búsqueda de doctores por nombre, obteniendo la filtración desde una sentencia SQL de Oracle.

Tecnologías: Oracle 11g, dwr, XHTML, DAO, AJAX, Java, JavaScript.

Se agregan las librerías necesarias para la ejecución del proyecto.



Archivo necesario para el mapeo de Doc para obtener la lista de doctores filtrados.
La conversión de la clase domain para ser accedido como un vean.

```
dwr.xml
<!DOCTYPE dwr PUBLIC
    "-//GetAhead Limited//DTD Direct Web Remoting 3.0//EN"
    "http://getahead.org/dwr/dwr30.dtd">

<dwr>
  <allow>

    <create creator="new" javascript="Doc" scope="script">
      <param name="class" value="dwr.Doc" />
    </create>

    <convert match="com.domain.Doctor" converter="bean" />
  </allow>
</dwr>
```

Configuración del archivo XML para el mapeo de dwr-invoker.

```
web.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns="http://java.sun.com/xml/ns/javaee"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
5   id="WebApp_ID" version="3.0">
6   <display-name>JSFDoctorDaoTx</display-name>
7   <welcome-file-list>
8     <welcome-file>faces/login.xhtml</welcome-file>
9   </welcome-file-list>
10
11   <servlet>
12     <servlet-name>JSF</servlet-name>
13     <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
14   </servlet>
15   <servlet-mapping>
16     <servlet-name>JSF</servlet-name>
17     <url-pattern>/faces/*</url-pattern>
18   </servlet-mapping>
19
20   <servlet>
21     <display-name>DWR Servlet</display-name>
22     <servlet-name>dwr-invoker</servlet-name>
23     <servlet-class>org.directwebremoting.servlet.DwrServlet</servlet-class>
24     <init-param>
25       <param-name>debug</param-name>
26       <param-value>true</param-value>
27     </init-param>
28   </servlet>
29
30   <servlet-mapping>
31     <servlet-name>dwr-invoker</servlet-name>
32     <url-pattern>/dwr/*</url-pattern>
33   </servlet-mapping>
34 </web-app>
```

Vista para mostrar los doctores dependiendo del carácter introducido.

```

search.xhtml 28
29
30 • <body>
31   <script type='text/javascript' src='/JSFDoctorOracle/dwr/engine.js'></script>
32 •   <script type='text/javascript'
33     src='/JSFDoctorOracle/dwr/interface/Doc.js'></script>
34   <script type='text/javascript' src='/JSFDoctorOracle/dwr/util.js'></script>
35
36 •   <script type="text/javascript">
37     var peopleCache = [];
38     var lastFilter = "";
39
40     function fillTable(doc) {
41
42         var filter = dwr.util.getValue("filter");
43
44         var pattern = new RegExp("(" + filter + ")", "i");
45         var filtered = [];
46
47         for (i = 0; i < doc.length; i++) {
48             if (pattern.test(doc[i].nombre)) {
49                 filtered.push(doc[i]);
50             }
51         }
52
53         dwr.util.removeAllRows("peoplebody");
54
55         dwr.util.addRows("peoplebody", filtered, [ function(doctor) {
56             return doctor.nombre;
57         }, function(doctor) {
58             return doctor.apellidos;
59         }, function(doctor) {
60             return doctor.direccion;
61         }, function(doctor) {
62             return doctor.telefono;
63         }, function(doctor) {
64             return doctor.rfc;
65         } ], {
66             escapeHtml : false
67         }, {
68             escapeHtml : false
69         });
70
71         peopleCache = doc;
72     }
73
74     function filterChanged() {
75         var filter = dwr.util.getValue("filter");
76
77         Doc.getDocFilter(filter, fillTable);
78     }
79   </script>
80   <br/>
81 •   <div class="container">
82     <a href="http://localhost:8080/JSFDoctorOracle/faces/inicio.xhtml"
83       type="button" class="btn btn-outline-dark">Regresar</a>
84   </div>
85 •   <div class="container">
86     <br /> <br />
87     <p>
88       Search: <input id="filter" onkeyup="filterChanged();" />
89     </p>
90 •   <table class="table table-hover">
91     <thead>
92     <tr>
93       <th>Nombre</th>
94       <th>Apellidos</th>
95       <th>Direccion</th>
96       <th>Telefono</th>
97       <th>Rfc</th>
98     </tr>
99     </thead>
100 •   <tbody id="peoplebody">
101   </tbody>
102 </table>

```

Método para ejecutar una sentencia SQL y obtener los doctores filtrados.

```
@Override
public List<Doctor> getDoctoresFilter(String caracter) {
    String sql = "SELECT id_doctor, nombre, apellidos, direccion, telefono, rfc, especialidad, sexo, correo, contra from doctor where nombre like ?";
    Connection conn = null;

    List<Doctor> listDocs = new ArrayList<Doctor>();
    try {
        conn = dataSource.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString(1, "%" + caracter + "%");
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            listDocs.add(new Doctor(rs.getInt("id_doctor"), rs.getString("nombre"), rs.getString("apellidos"),
                rs.getString("direccion"), rs.getString("telefono"), rs.getString("rfc"),
                rs.getString("especialidad"), rs.getString("sexo"), rs.getString("correo"),
                rs.getString("contra")));
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
        return null;
    }
    return listDocs;
}
```

Método donde se obtiene la lista de los doctores disponibles para el bean;

```
Doc.java
1 package dwr;
2
3 import java.util.ArrayList;
12
13 public class Doc {
14
15     public List<Doctor> getDocFilter(String filter) {
16         System.out.println("Filtro obtenido " + filter);
17         List<Doctor> list = new ArrayList<Doctor>();
18         HorarioService service2 = new HorarioDao();
19         list = service2.getDoctoresFilter(filter);
20         System.out.println("Info" + list);
21         return list;
22     }
23
24 }
25 |
```

Bienvenido Doctor(a):



Alejandro Herrera Mendoza

Información

Dirección: privada 16 de septiembre

Especialidad: java

Rfc: HEMAHG76

Sexo: Hombre

Fecha: Wed Jan 06 01:55:24 CST 2021

Busqueda

Crear horario

Semanas

1

Dia

Hora inicio

Hora termino

Lunes

0

0

Martes

0

0

Regresar

Search:

Nombre	Apellidos	Direccion	Telefono	Rfc
--------	-----------	-----------	----------	-----

Regresar

Search:

Nombre	Apellidos	Direccion	Telefono	Rfc
Marco	Mendoza	8 sur este	24916745474	MARDGF765

Regresar

Search:

Nombre	Apellidos	Direccion	Telefono	Rfc
Alejandro	Herrera Mendoza	Privada 16 de septiembre	24937474	HEMAHDHF76
Alejandro	Herrera Mendoza	privada 16 de septiembre	24915637	HEMAHG76
Alejandro	Herrera Mendoza	privada 16 de septiembre	24915637	HEMAHG76
Manuel	Mendoza Pérez	av 6 sur	249264747	HDMDG755
Raul	Mendez	Privada 16 norte	2484757	Doh