

PROYECTO PROGRAMACIÓN II



Manual de Usuario Tecnico

Fecha: 17/11/2019

HOJA DE CONTROL

CONTROL DE DISTRIBUCIÓN

| Nombre y Apellidos |
|--|
| Ángel Roque Morataya 6590-18-5318 sección A |
| Oliver Jolon Marín 6590-18-4347 sección A |
| |
| |

1 DESCRIPCIÓN DEL SISTEMA

1.1 *Objeto*

Lograr que el usuario técnico pueda entender paso a paso el cómo debe de utilizar el programa ya realizado, que pueda entender cada sentencia ya puesta dentro del mismo y logre llegar a una satisfacción con los resultados de cada uno de los temas.

1.2 *Alcance*

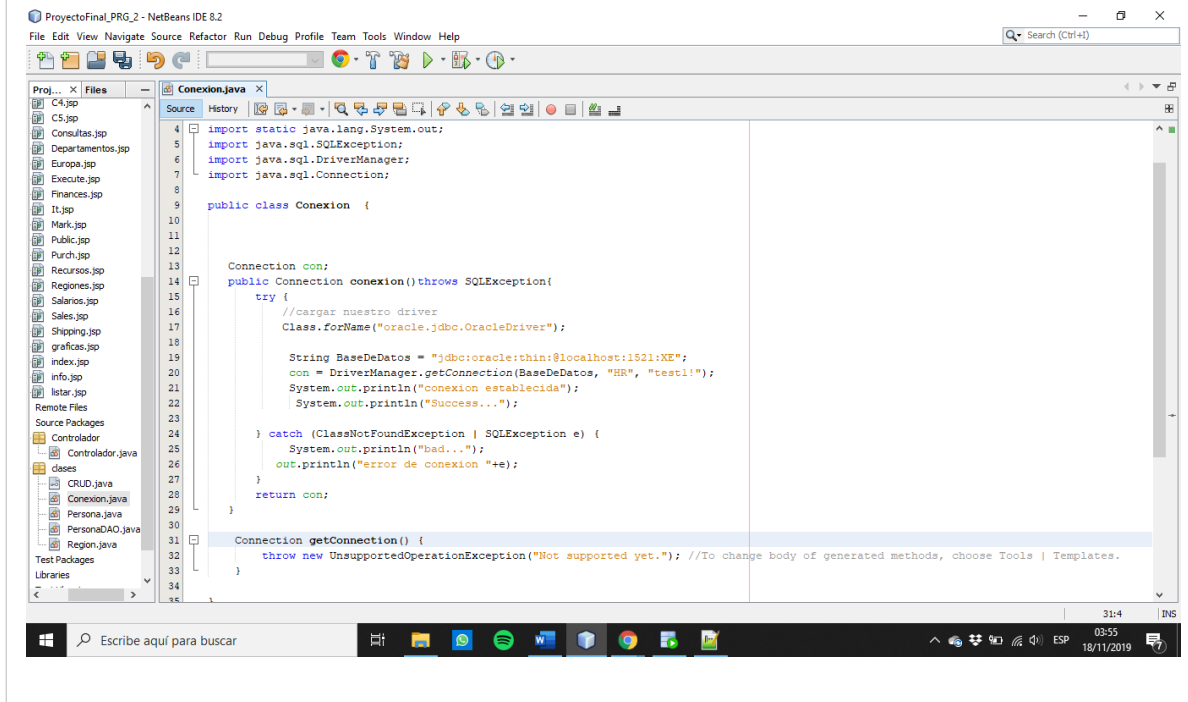
Llegar a dar a conocer por parte de la documentación más a fondo sobre el programa antes de utilizarlo y ponerlo en práctica, que pueda ser entendible y así mismo útil el documento como parte esencial.

1.3 *Funcionalidad*

En este caso el programa realizado en el lenguaje Java utilizando Netbeans 8.2 y la base de datos de Oracle SQL Developer, un desarrollador web que es BOOTSTRAP, tiene como fin varias funciones, tiene diferentes consultas sobre las tablas de HR de la base de datos, realizado de una forma muy practica y sencilla especialmente para el usuario técnico, que pueda utilizarlo de una forma fácil y rápida para obtener resultados confiables que ya han sido tomados en cuenta por parte nuestra, que los datos son verídicos y exactos en cada una de las acciones que se dan dentro de cada pantalla.

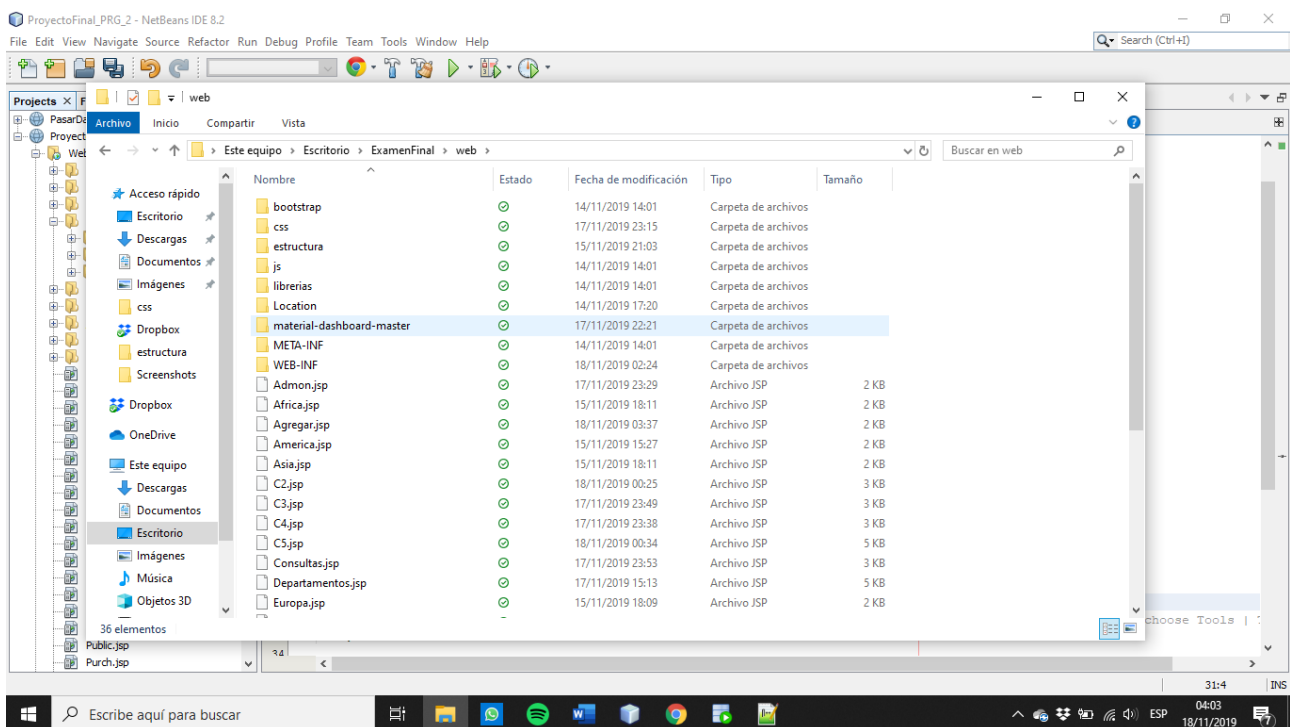
2. CONEXIÓN A LA BASE DE DATOS

El programa primero consiste en hacer la conexión a la base de datos hacemos un paquete nuevo que le nombramos clases en el folder Source Packages.

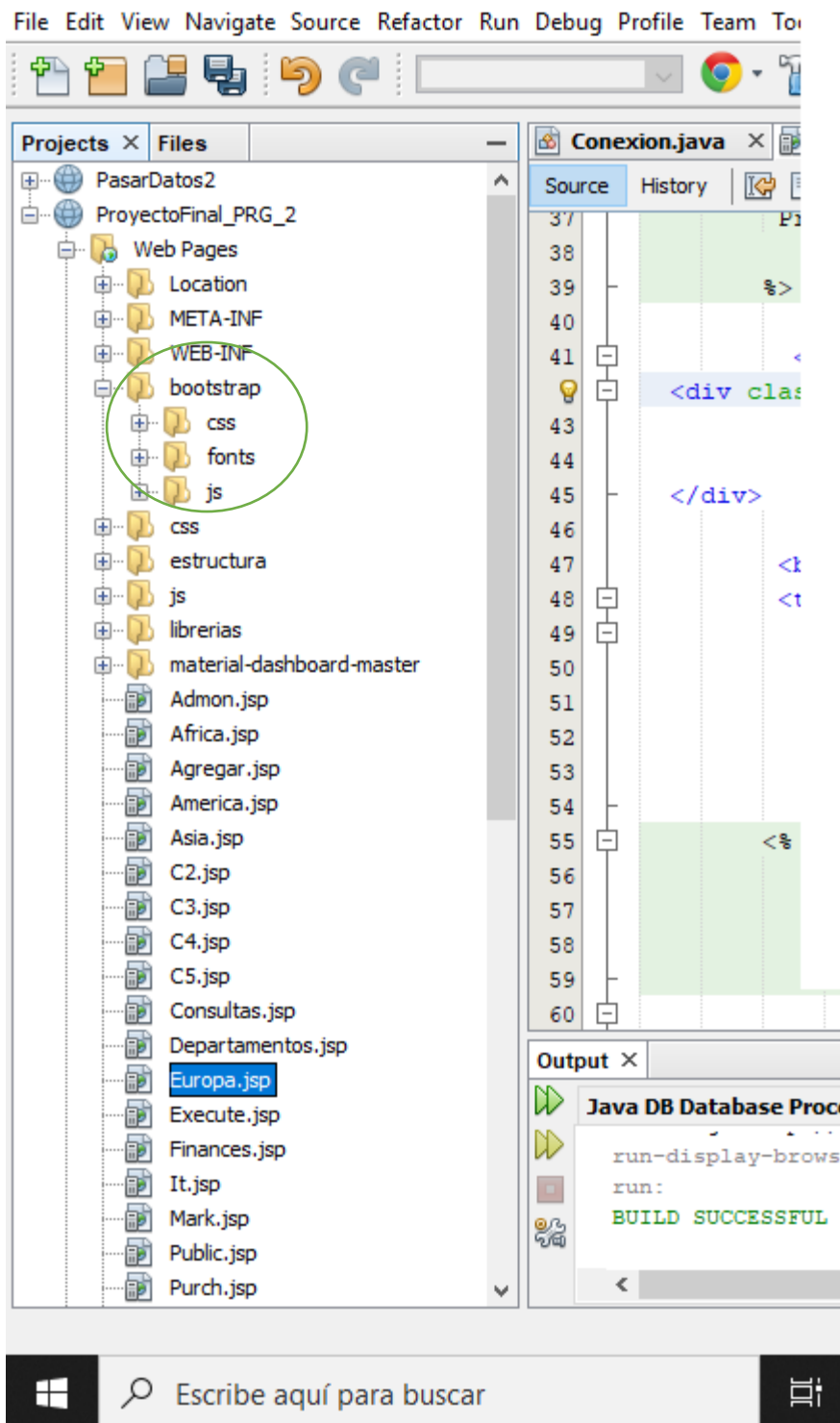


1.4 INSERTAMOS LA CARPETA BOOTSTRAP

Se inserta la carpeta del Bootstrap directamente al proyecto en la carpeta web para poder hacer uso de sus diseños.

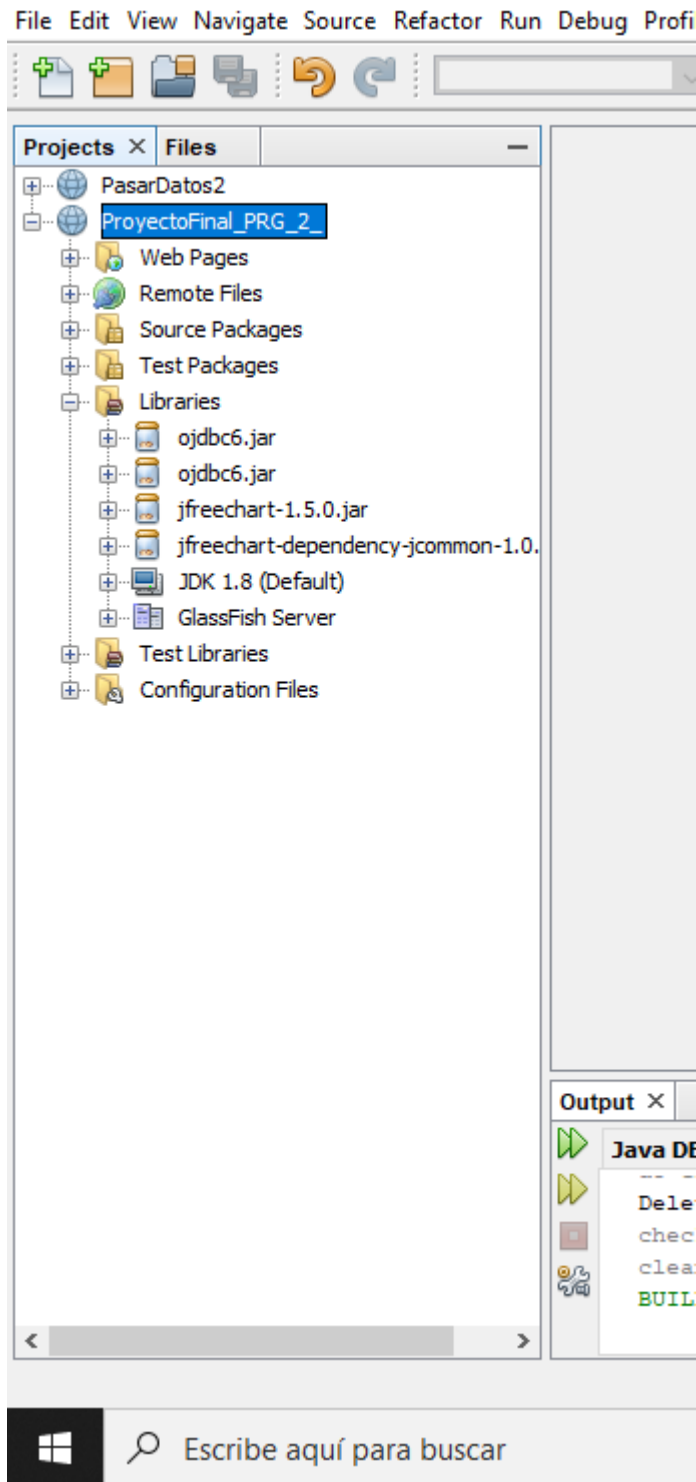


Al insertar la tendra que aparecer en netbeans carpeta del bootstrap.



1.5 EN LA CARPETA LIBRARIES

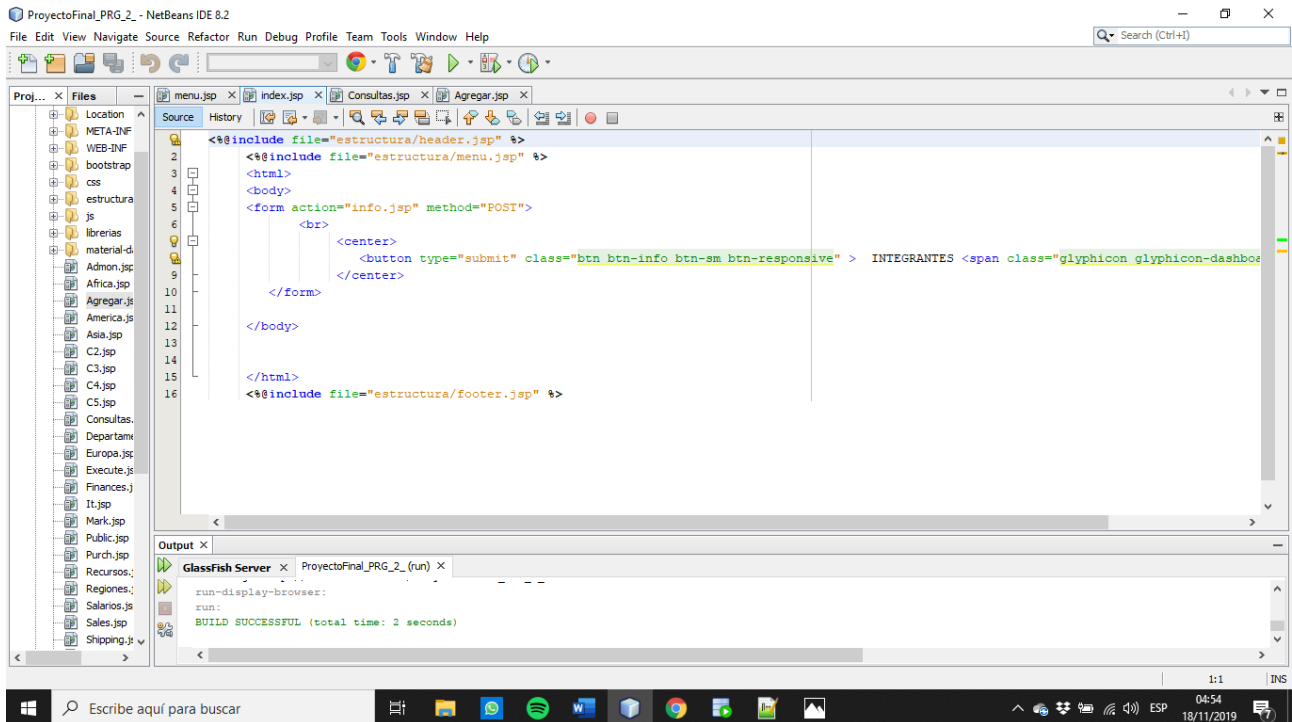
Aparece el servidor local que usamos en este caso es GlassFish Server, también utilizamos un controlador JDBC (Java Database Connectivity) para usar con SQL Server Database que es jdbc 6 y un jfreechart para poder graficar.



Y por defecto utilizamos el JDK 1.8.

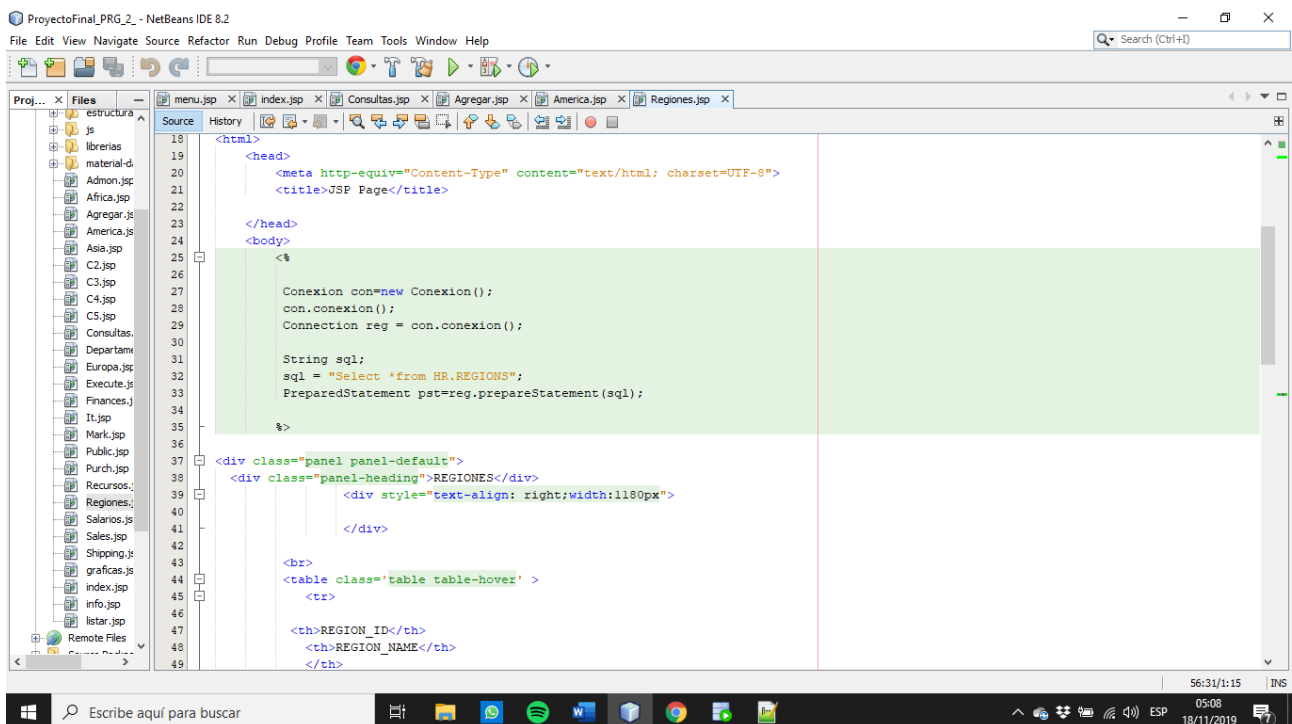
2.3 INDEX

En el index que se crea por default solo esta hecho un botón con la información de los integrantes

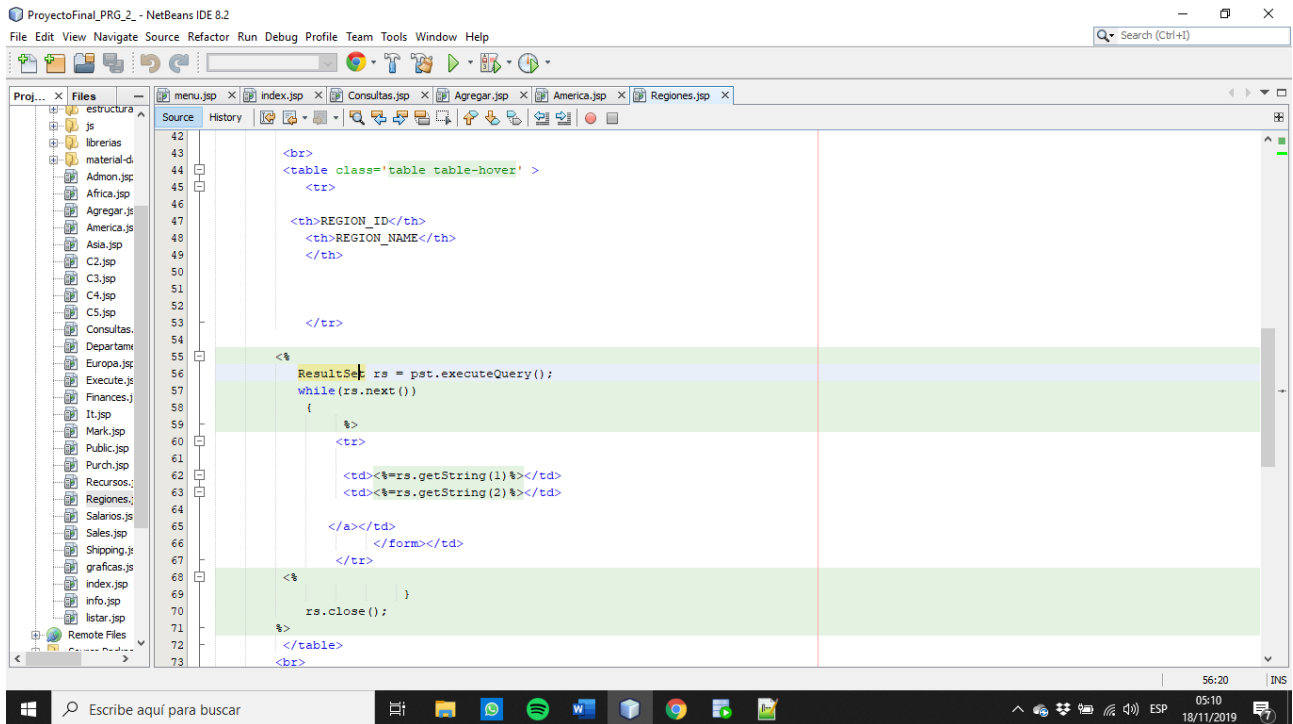


2.4 LA PRIMERA PANTALLA

Acá se hace un JSP para hacer la primera pantalla acá se hace una consulta a la base de datos para obtener la tabla de Regiones, se declara la conexión y con un "Select *from HR.REGIONS" se obtiene la tabla del Oracle de SQL.



Mandamos a imprimir la información con `ResultSet rs = pst.executeQuery()` y en una tabla de HTML un `while(rs.next())` y `<%=rs.getString(1)%>`

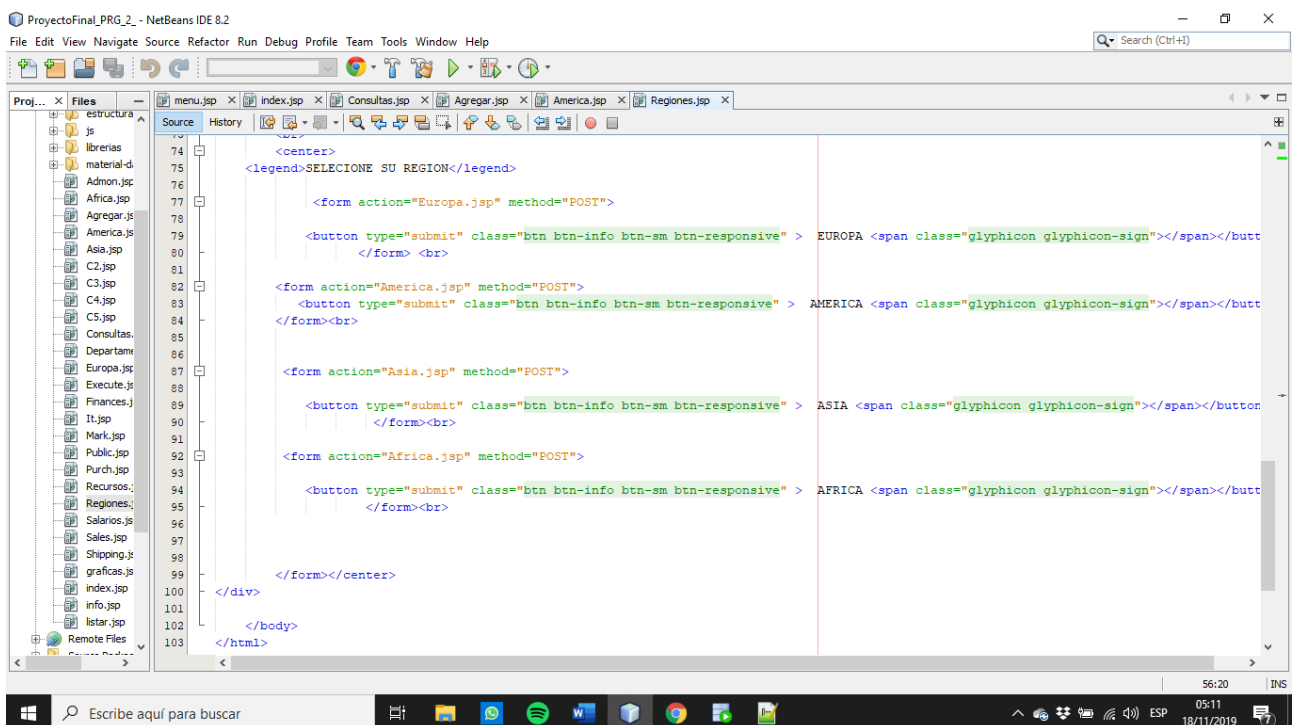


```

42
43
44 <table class="table table-hover" >
45 <tr>
46
47 <th>REGION_ID</th>
48 <th>REGION_NAME</th>
49 </th>
50
51
52 </tr>
53
54
55 <%
56     ResultSet rs = pst.executeQuery();
57     while(rs.next())
58     {
59         <%
60         <tr>
61
62         <td><%=rs.getString(1)%></td>
63         <td><%=rs.getString(2)%></td>
64
65         </td></td>
66         </form></td>
67         </tr>
68
69         <%
70         rs.close();
71         <%
72     </table>
73     <br>

```

Luego se creo en botones los campos de Regiones para que muestren los países que hay registrados por continente.



```

74
75 <center>
76 <legend>SELECCIONE SU REGION</legend>
77
78 <form action="Europa.jsp" method="POST">
79 <button type="submit" class="btn btn-info btn-sm btn-responsive" > EUROPA <span class="glyphicon glyphicon-sign"></span></button>
80 </form> <br>
81
82 <form action="America.jsp" method="POST">
83 <button type="submit" class="btn btn-info btn-sm btn-responsive" > AMERICA <span class="glyphicon glyphicon-sign"></span></button>
84 </form><br>
85
86 <form action="Asia.jsp" method="POST">
87 <button type="submit" class="btn btn-info btn-sm btn-responsive" > ASIA <span class="glyphicon glyphicon-sign"></span></button>
88 </form><br>
89
90 <form action="Africa.jsp" method="POST">
91 <button type="submit" class="btn btn-info btn-sm btn-responsive" > AFRICA <span class="glyphicon glyphicon-sign"></span></button>
92 </form><br>
93
94 </form></center>
95
96 </div>
97
98 </body>
99
100 </html>
101
102
103

```


Se creo un JSP por cada continente para que muestre los países de cada continente que tenga registrado.

```

30  <%
31
32      Conexion con=new Conexion();
33      con.conexion();
34      Connection reg = con.conexion();
35
36      String sql;
37      sql = "SELECT * FROM HR.COUNTRIES WHERE REGION_ID='2'";
38      PreparedStatement pst=reg.prepareStatement(sql);
39
40  >%
41
42  <div class="panel panel-default">
43  <div class="panel-heading">PAISES DE AMERICA
44
45
46
47  </div>
48
49  <br>
50  <table class="table table-bordered">
51  <tr>
52  <th>COUNTRY_ID</th>
53  <th>NOMBRE</th>
54  <th>REGION_ID</th>
55  </tr>
56
57  <%
58      ResultSet rs = pst.executeQuery();
59      while(rs.next())
60      {

```

Se hace referencia a Regiones.jsp para crear la pantalla en Bootstrap en la carpeta estructura en menú.jsp.

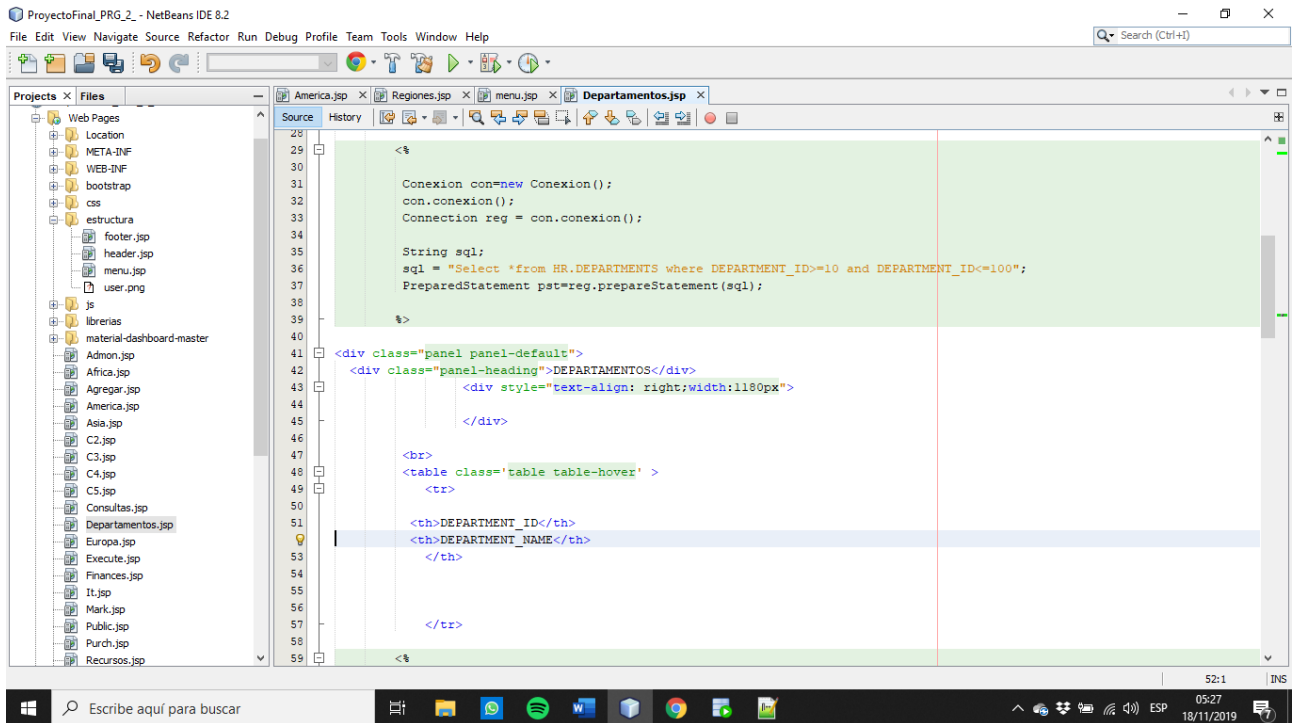
```

2
3
4  <div class="collapse navbar-collapse" id="navegacion-fm">
5  <ul class="nav navbar-nav">
6
7  <li class="active"><a href="index.jsp" class="glyphicon glyphicon-home"> Inicio</a></li>
8
9  <li class="active"><a href="Regiones.jsp" class="glyphicon glyphicon-globe"> Regiones</a></li>
10 <li class="active"><a href="Departamentos.jsp" class="glyphicon glyphicon-briefcase"> Departamentos</a></li>
11 <li class="active"><a href="Consultas.jsp" class="glyphicon glyphicon-book"> Consultas</a></li>
12 <li class="active"><a href="listar.jsp" class="glyphicon glyphicon-list"> Mostrar Registros</a></li>
13 <li class="active"><a href="Agregar.jsp" class="glyphicon glyphicon-user"> Agregar</a></li>
14 <li class="active"><a href="graficas.jsp" class="glyphicon glyphicon-signal"> Graficas</a></li>
15
16 </ul>
17
18 </div>
19
20
21
22
23
24
25
26
27

```

2.5 PANTALLA DOS

De la misma manera que se consulta la tabla de Regiones se consulta la tabla de Departamentos y se crea un botón por cada departamento y cada botón es un JSPs por campo de esa misma tabla, así muestra los empleados que están por departamento y se manda a llamar al menu.jsp al igual que el de regiones.



```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

<%
    Conexion con=new Conexion();
    con.conexion();
    Connection reg = con.conexion();

    String sql;
    sql = "Select *from HR.DEPARTMENTS where DEPARTMENT_ID=>10 and DEPARTMENT_ID<=100";
    PreparedStatement pst=reg.prepareStatement(sql);

    %>

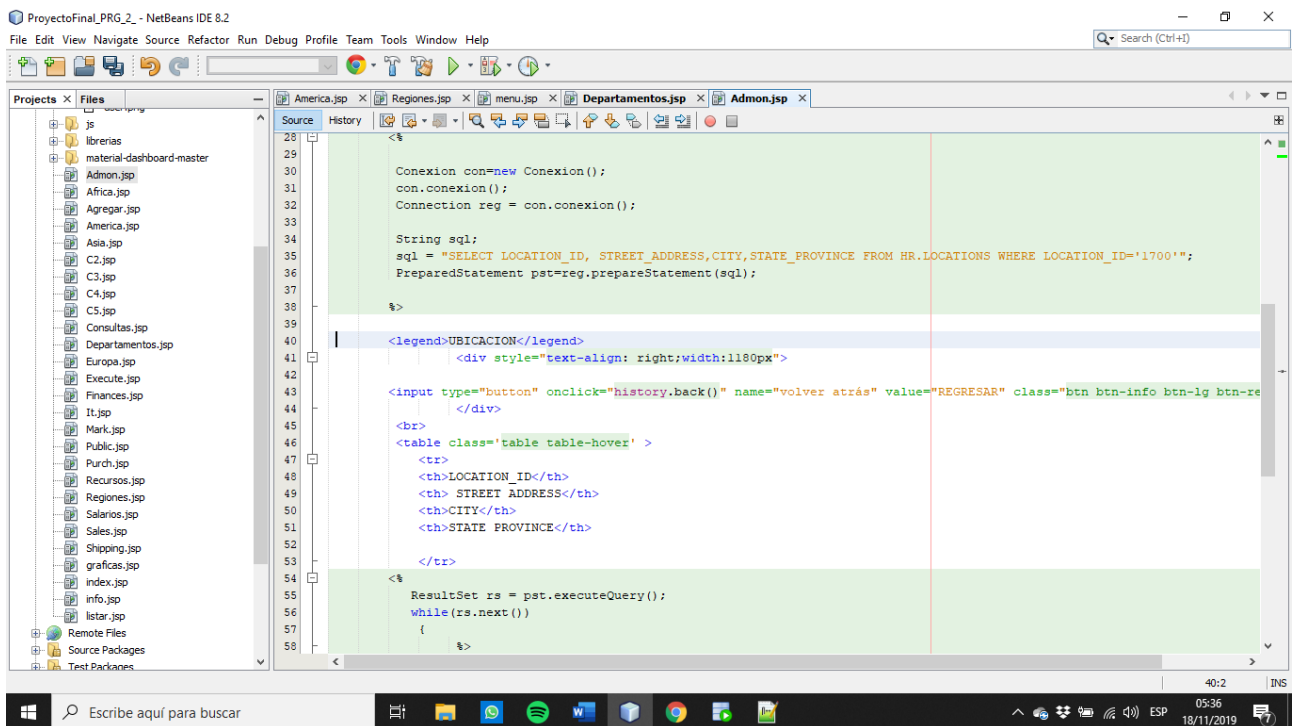
<div class="panel panel-default">
<div class="panel-heading">DEPARTAMENTOS</div>
<div style="text-align: right; width: 1180px">
    </div>

    <br>
    <table class="table table-hover">
    <tr>
    <th>DEPARTMENT_ID</th>
    <th>DEPARTMENT_NAME</th>
    </tr>

    </tr>

    <%

```



```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

<%
    Conexion con=new Conexion();
    con.conexion();
    Connection reg = con.conexion();

    String sql;
    sql = "SELECT LOCATION_ID, STREET_ADDRESS, CITY, STATE_PROVINCE FROM HR.LOCATIONS WHERE LOCATION_ID='1700'";
    PreparedStatement pst=reg.prepareStatement(sql);

    %>

<legend>UBICACION</legend>
<div style="text-align: right; width: 1180px">
    </div>

    <input type="button" onclick="history.back()" name="volver atrás" value="REGRESAR" class="btn btn-info btn-lg btn-re">
    <br>
    <table class="table table-hover">
    <tr>
    <th>LOCATION_ID</th>
    <th>STREET ADDRESS</th>
    <th>CITY</th>
    <th>STATE PROVINCE</th>
    </tr>

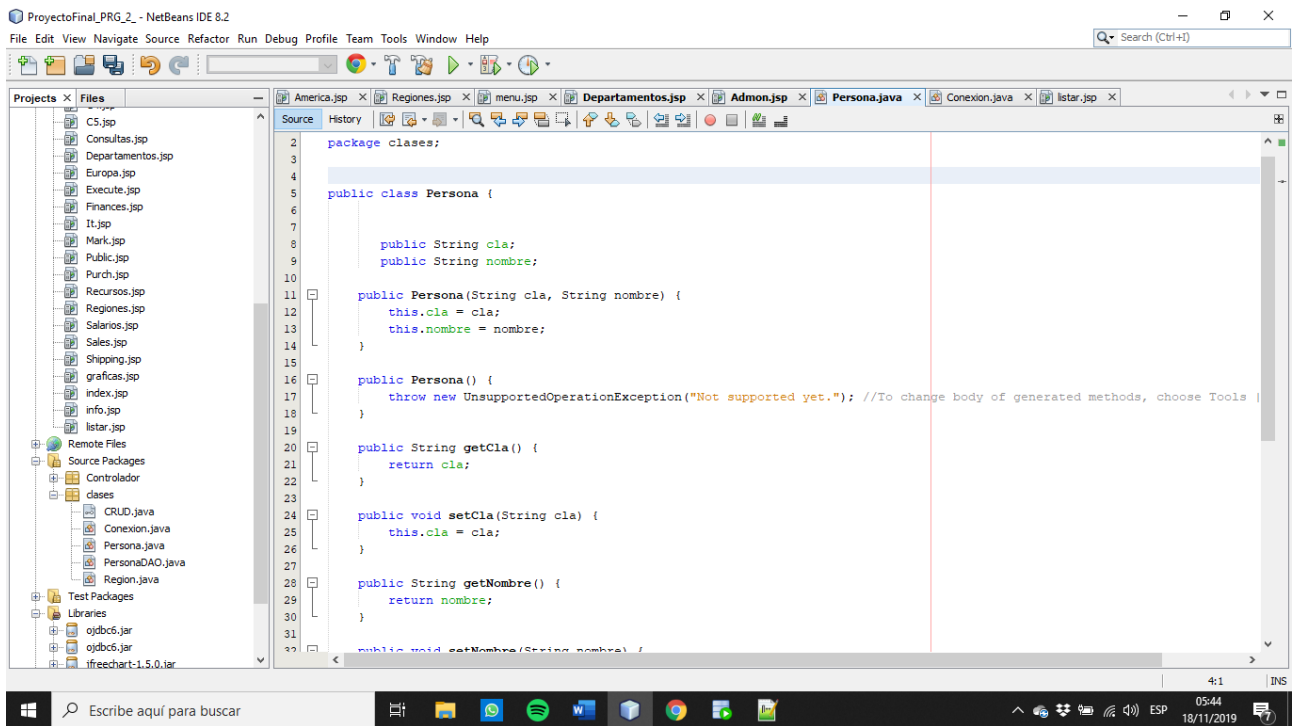
    </tr>

    <%
    ResultSet rs = pst.executeQuery();
    while(rs.next())
    {
        %>

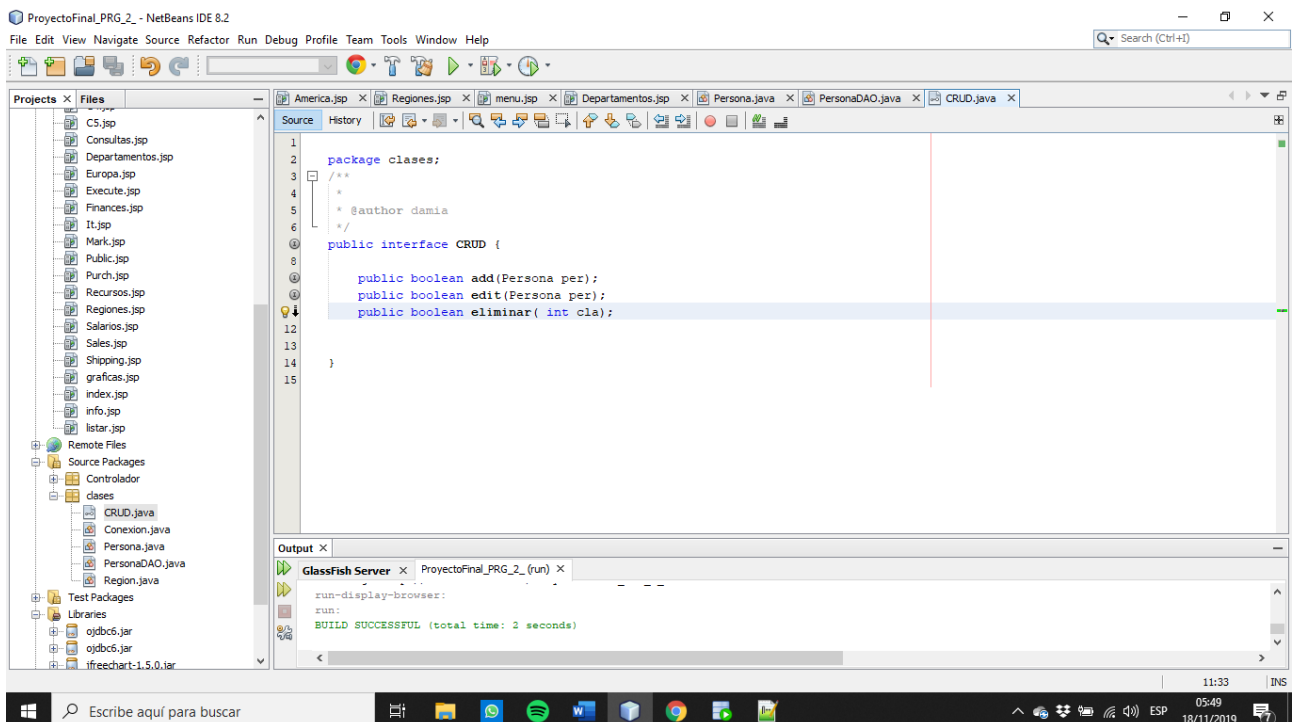
```

2.6 PANTALLA TRES

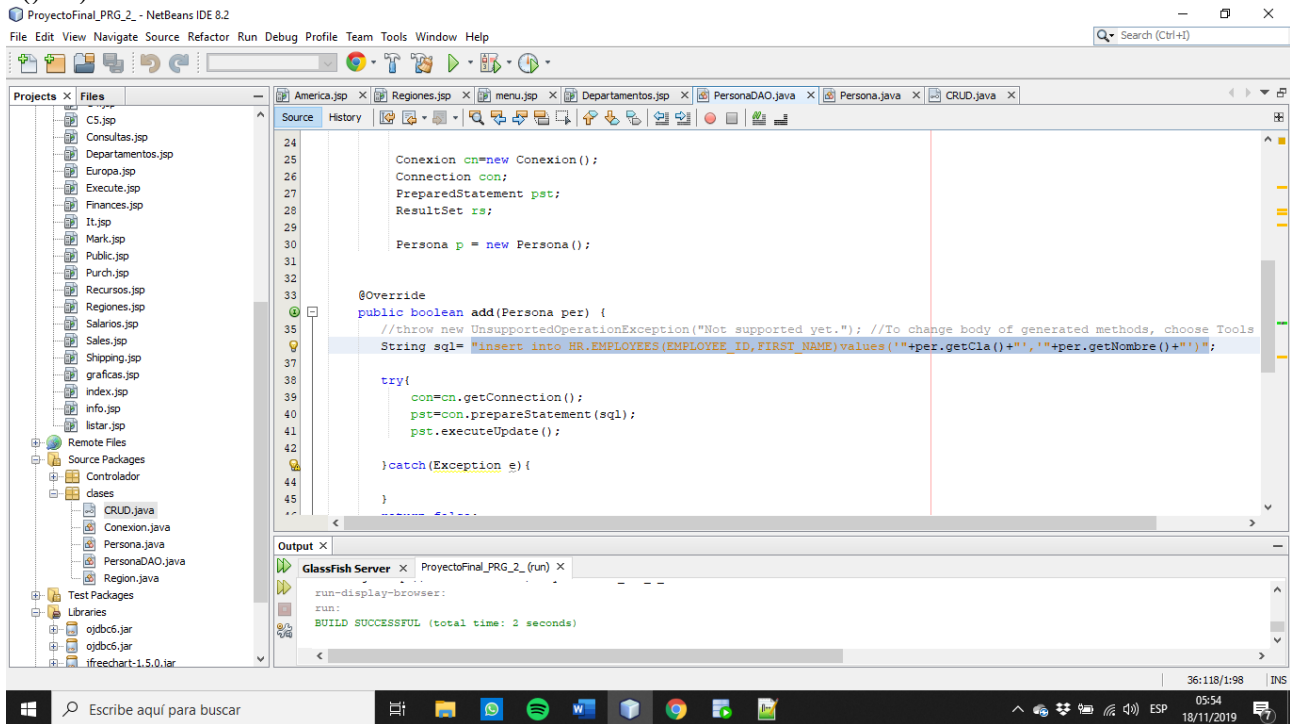
El procedimiento de esto se hace la carpeta Source Packages y se crea un paquete clases y un Controlador en esta pantalla están las altas, bajas y consultas estas realizan con el MVC se crea una clase Persona con los objetos ID y Nombre con un constructor y un con los métodos get y set.



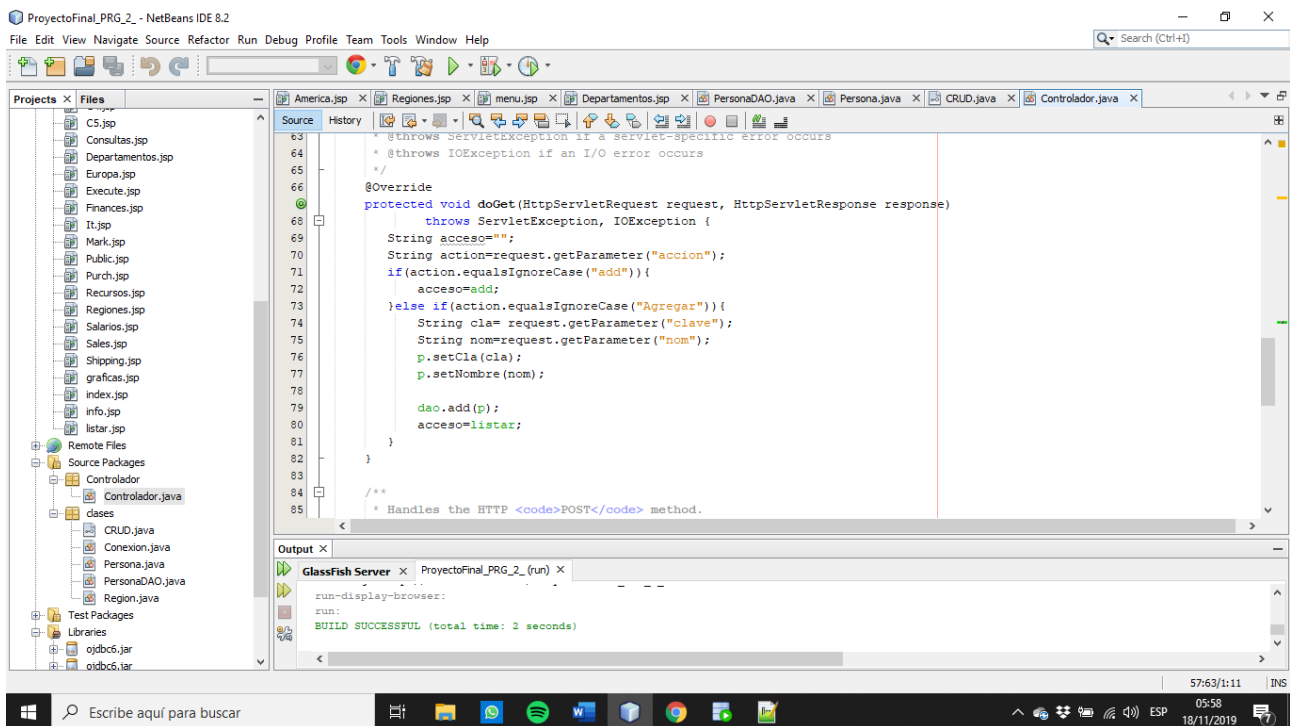
Luego se crea un intfaz.java con el nombre de CRUD para declarar metodos que se van a realizar.



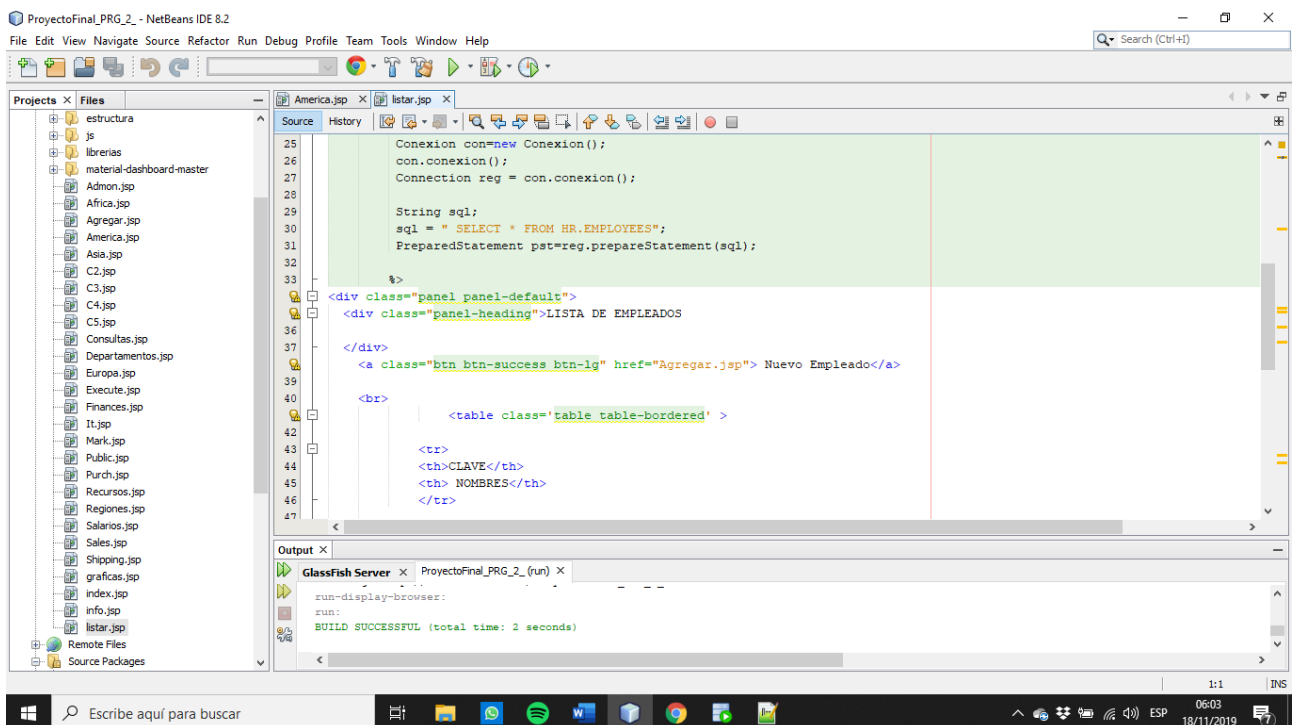
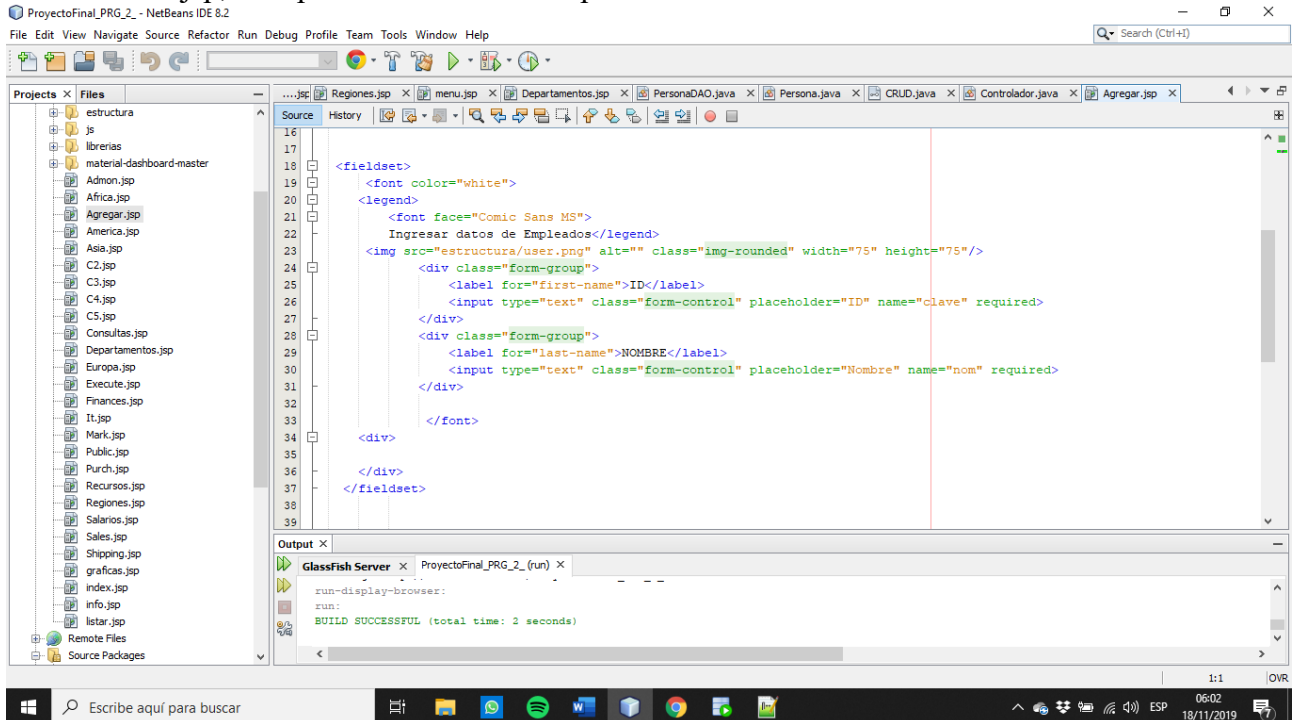
Se hace otra clase con el nombre PersonaDao se llaman los métodos que se del CRUD.java ahí se la conexión a la base de datos para hacer el método de agregar un registro desde el JSPs con "insert into HR.EMPLOYEES(EMPLOYEE_ID,FIRST_NAME)values('"+per.getCla()+"', '"+per.getNombre()+"'")".



Luego de esto nos vamos al paquete Controlador y crea un servlet con el nombre de controlador y se trabaja en método doGet para dar acceso de poder ingresar un nuevo dato.

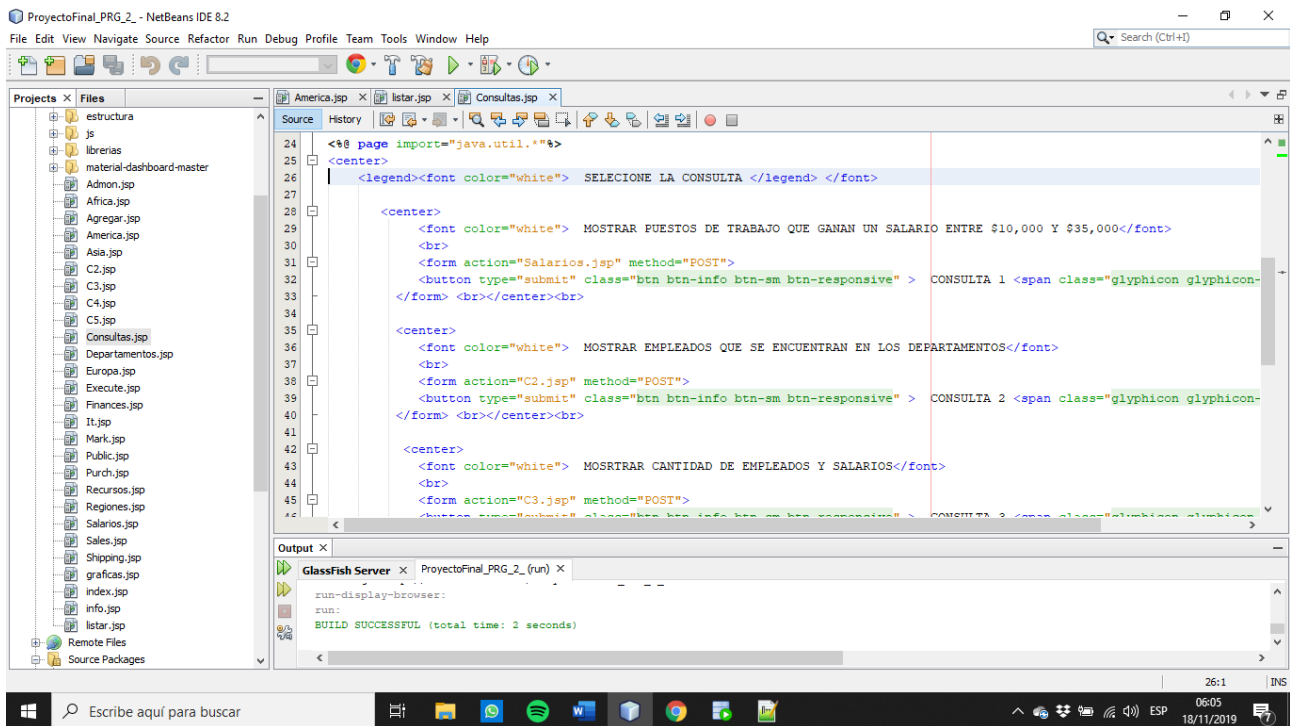


Luego nos vamos a la vista donde el Agregar.jsp se crea un formulario para crear dos campos donde se agregara un nuevo registro, ahí un botón que cuando se la función se dirige a un jsp con el nombre listar.jsp, ahí aparecen todos los campos de la tabla EMPLOYEES.



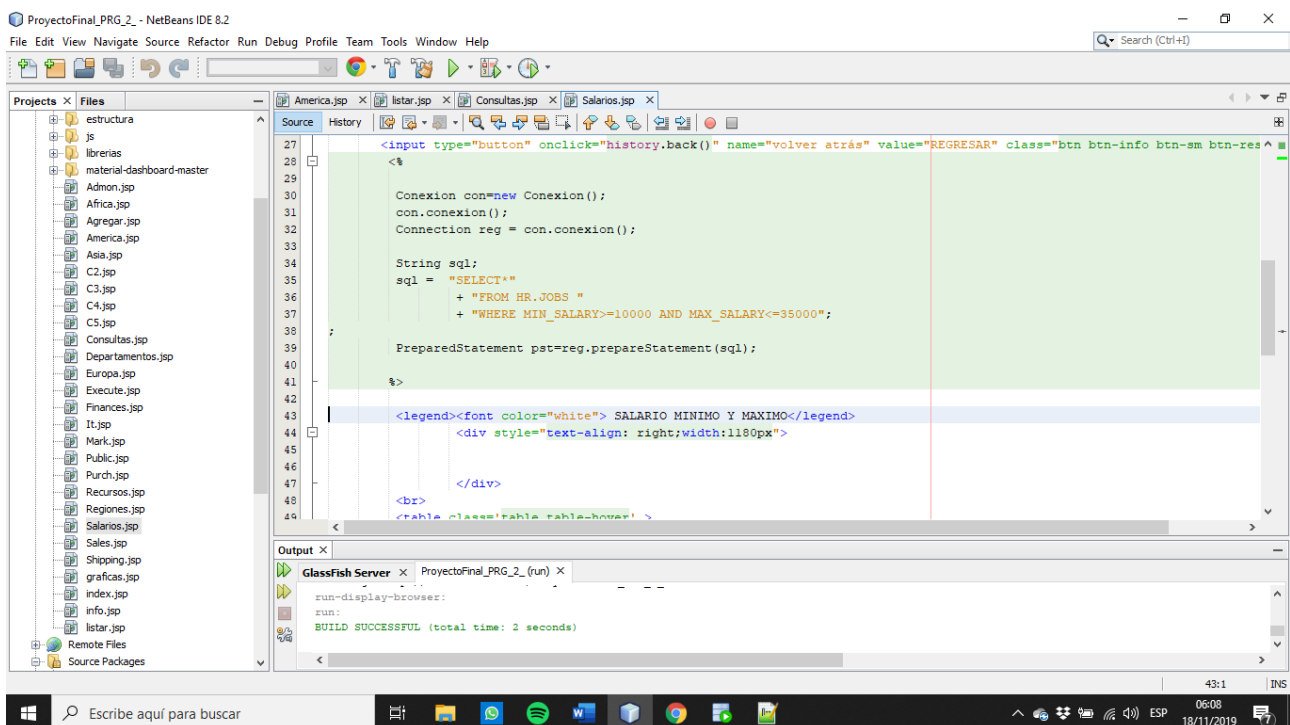
2.7 PANTALLA CUATRO

Acá se crea un jsp por cada consulta se creo un Consultas.jsp para hacer todos lo botones de cada consulta con HTML y BOOTSTRAP



CONSULTA 1

Se manda a llamar la tabla con "SELECT* FROM HR.JOBS WHERE MIN_SALARY>=10000 AND MAX_SALARY<=35000";



CONSULTA 2

Se consulta la tabla con "select

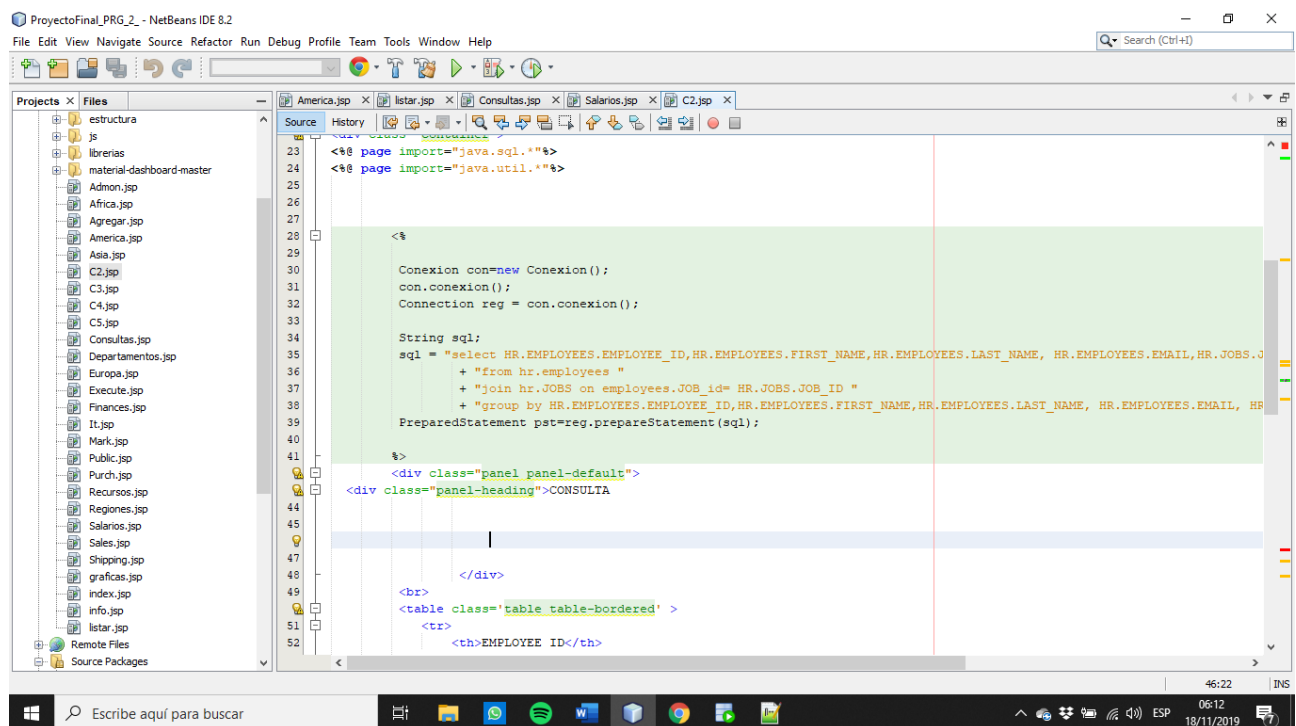
HR.EMPLOYEES.EMPLOYEE_ID,HR.EMPLOYEES.FIRST_NAME,HR.EMPLOYEES.LAST_NAME, HR.EMPLOYEES.EMAIL,HR.JOBS.JOB_ID, HR.JOBS.JOB_TITLE, HR.EMPLOYEES.SALARY "

+ "from hr.employees "

+ "join hr.JOBS on employees.JOB_id= HR.JOBS.JOB_ID "

+ "group by

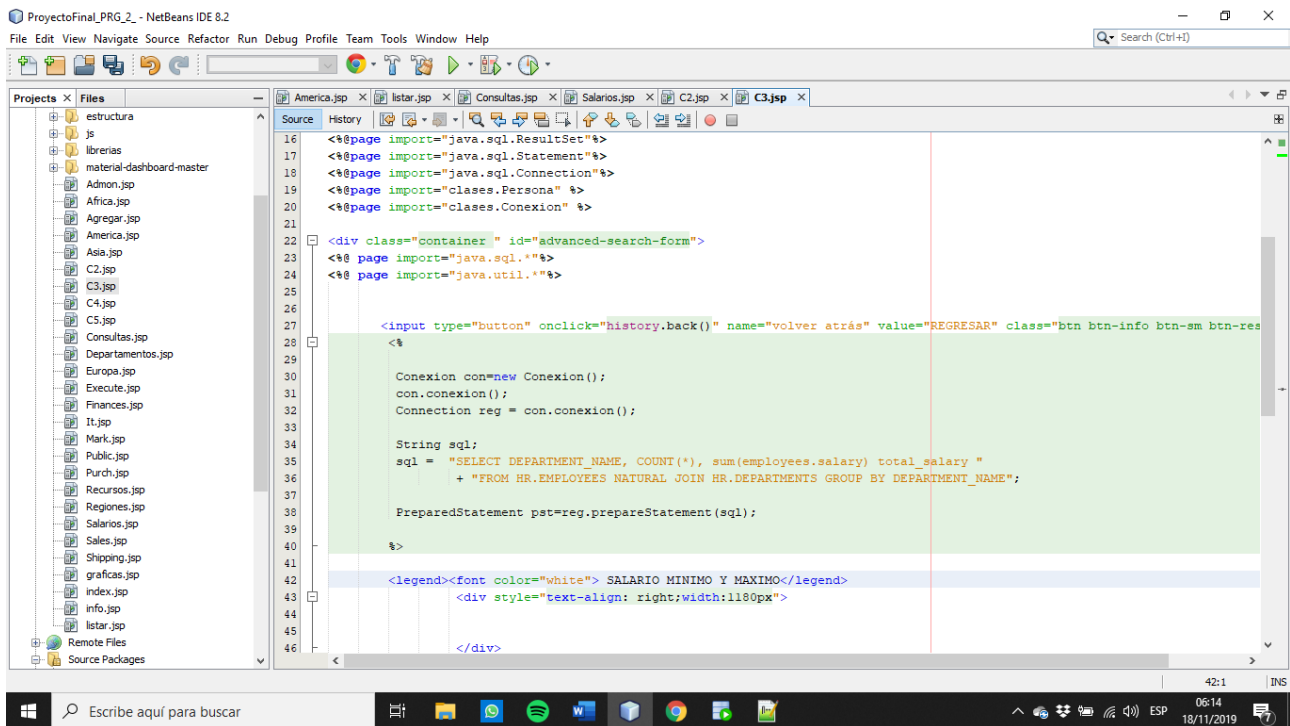
HR.EMPLOYEES.EMPLOYEE_ID,HR.EMPLOYEES.FIRST_NAME,HR.EMPLOYEES.LAST NAME, HR.EMPLOYEES.EMAIL, HR.JOBS.JOB_ID, HR.JOBS.JOB_TITLE, HR.EMPLOYEES.SALARY ";



CONSULTA 3

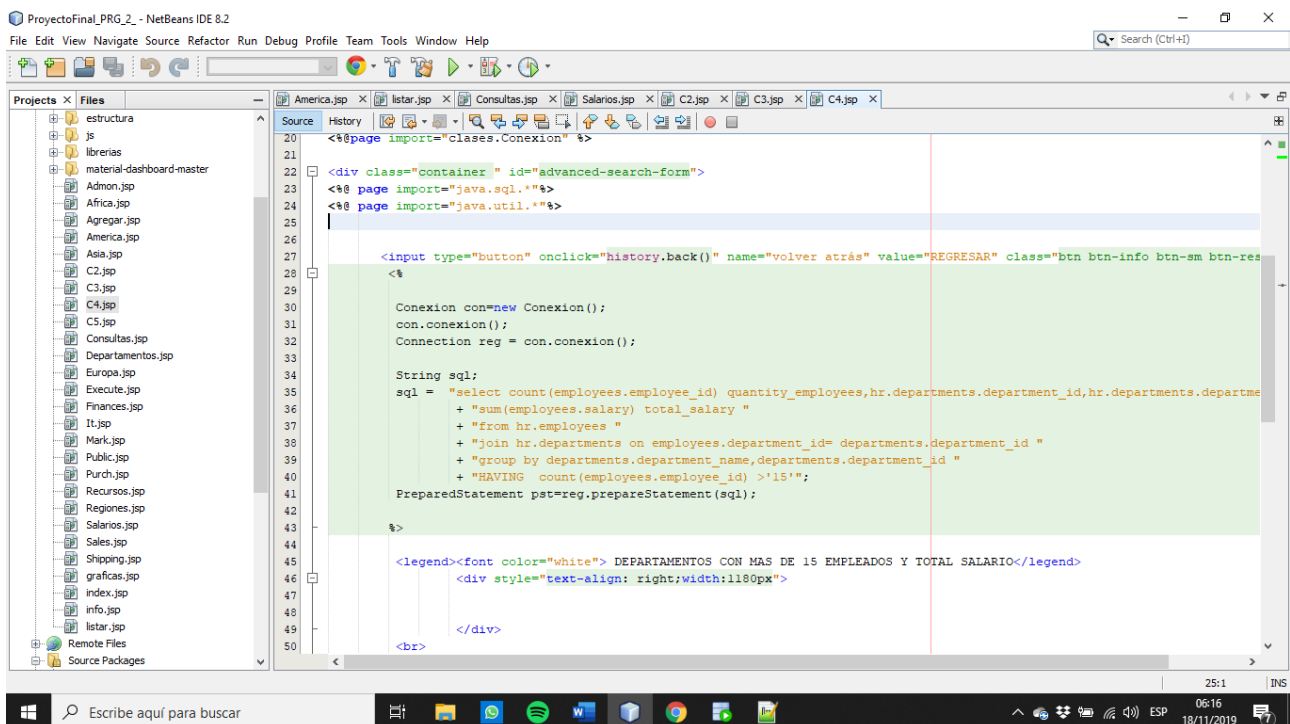
Se consulta la table con "SELECT DEPARTMENT_NAME, COUNT(*), sum(employees.salary) total_salary "

+ "FROM HR.EMPLOYEES NATURAL JOIN HR.DEPARTMENTS GROUP BY DEPARTMENT_NAME";



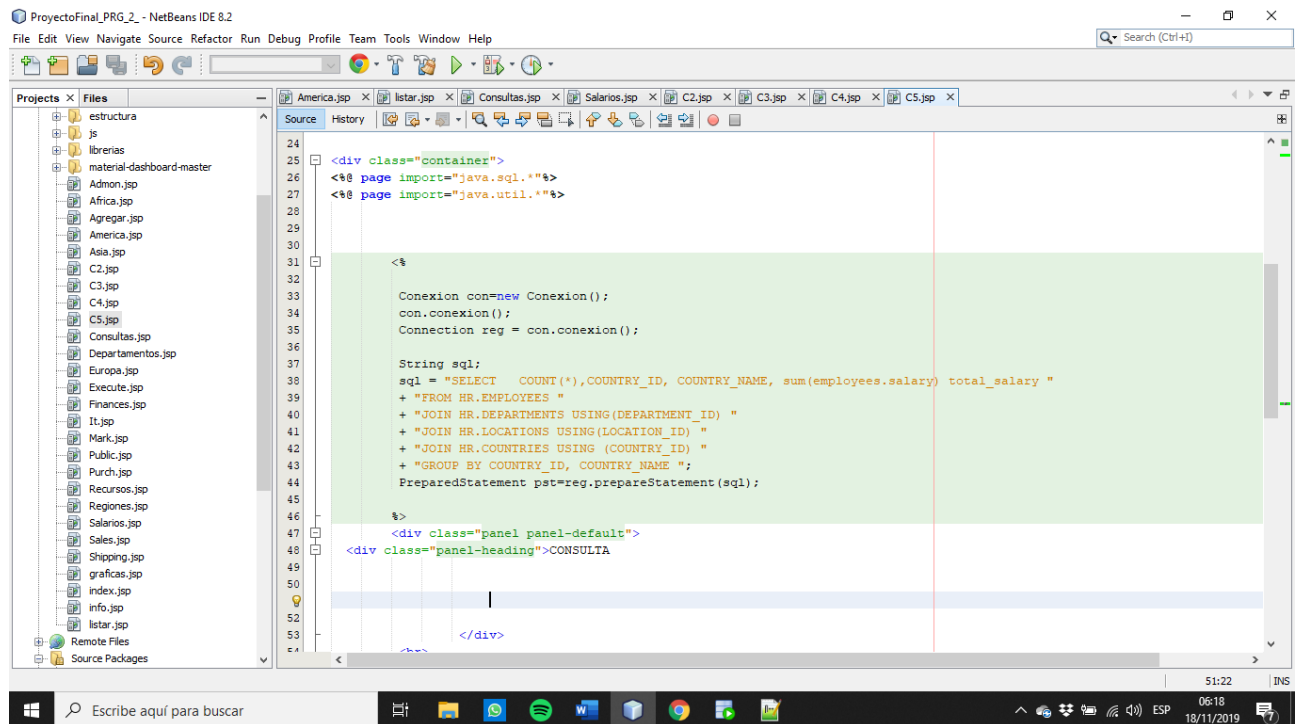
CONSULTA 4

Se consulta la tabla con "select count(employees.employee_id)
 quantity_employees,hr.departments.department_id,hr.departments.department_name,"
 + "sum(employees.salary) total_salary "
 + "from hr.employees "
 + "join hr.departments on employees.department_id= departments.department_id "
 + "group by departments.department_name,departments.department_id "
 + "HAVING count(employees.employee_id) >'15'";



CONSULTA 5

Se consulta la tabla con "SELECT COUNT(*),COUNTRY_ID, COUNTRY_NAME, sum(employees.salary) total_salary FROM HR.EMPLOYEES JOIN HR.DEPARTMENTS USING(DEPARTMENT_ID) JOIN HR.LOCATIONS USING(LOCATION_ID) JOIN HR.COUNTRIES USING (COUNTRY_ID) GROUP BY COUNTRY_ID, COUNTRY_NAME ";



GRAFICAS

Se utilizo la libreria JFreeChart para hacer la graficas, se importaron sus paquetes y se hace la Conexion con la base de datos para pedir la tabla que se va a graficar y luego se va a Instanciar un objeto que corresponde al tipo de grafico continuamos, Cargaremos la data las series y por ultimo hacemos Generar el grafico.

