

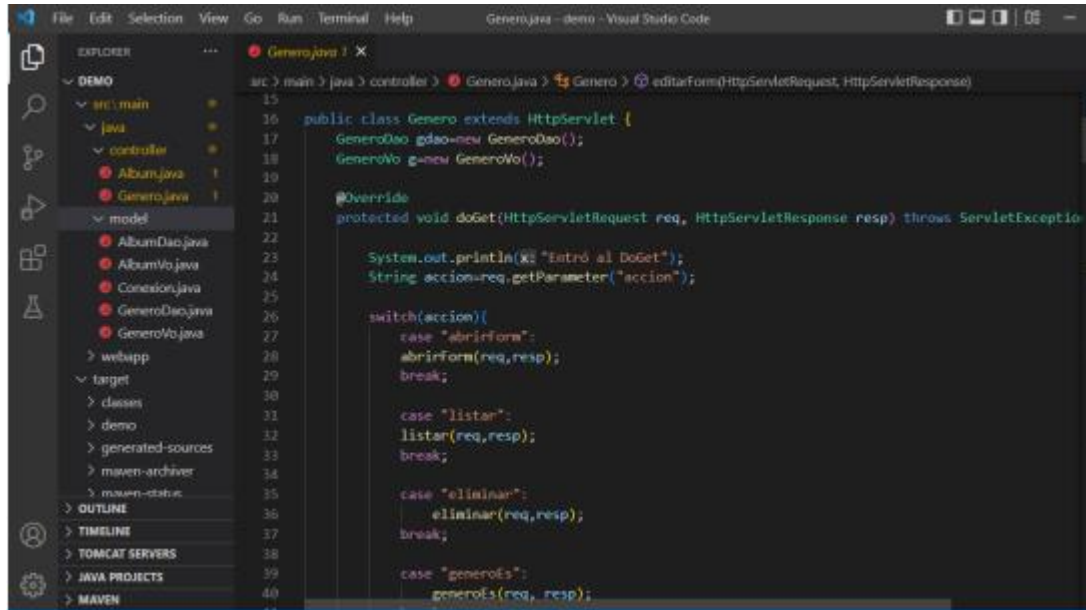
Proyecto Disquera-Demo

DISEÑO DE LA CRUD CON JAVA

STEFANIA CHUMBE RODRIGUEZ

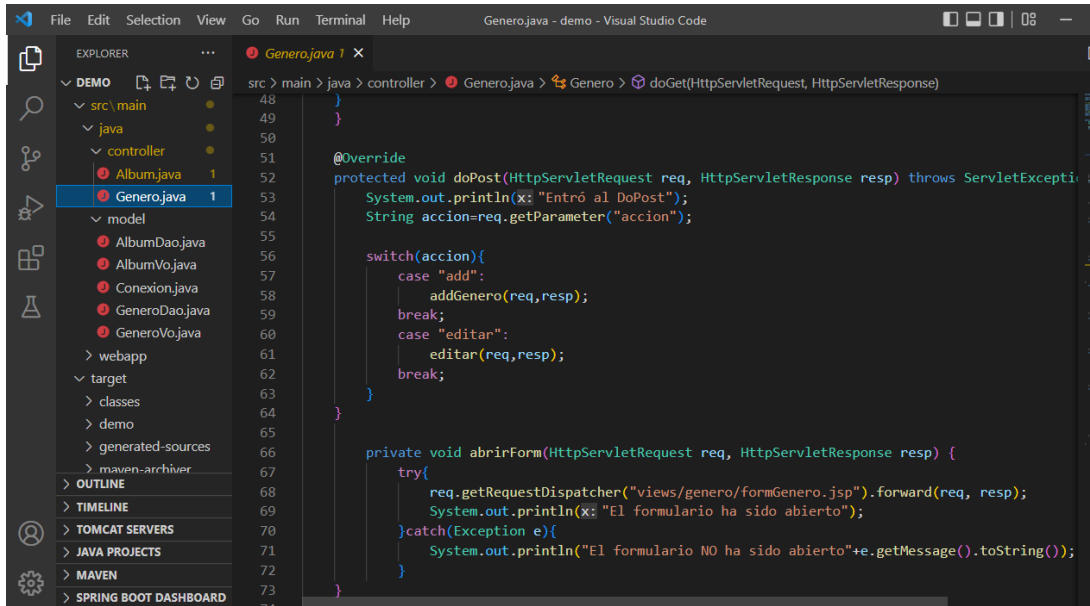
GENERO

- Controller o sevelet cuenta con los métodos do Get quien inicia los argumentos de requerimiento y respuesta y do Post quien también con requerimiento y respuesta para cuando el usuario realice una petición. También se muestra un switch con el que los recorridos, peticiones del usuario.



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The project structure includes 'src/main/java/controller' with files 'Album.java', 'Genero.java', and 'model'. The 'model' folder contains 'AlbumDao.java', 'AlbumVo.java', 'Conexion.java', 'GeneroDao.java', and 'GeneroVo.java'. The 'doGet' method in 'Genero.java' is highlighted, showing a switch statement for actions: 'abrirform', 'listar', 'eliminar', and 'generoEs'.

```
src > main > java > controller > Genero.java > Genero > editarForm(HttpServletRequest, HttpServletResponse)
15
16 public class Genero extends HttpServlet {
17     GeneroDao gdao=new GeneroDao();
18     GeneroVo g=new GeneroVo();
19
20     @Override
21     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException
22
23         System.out.println(x: "Entró al DoGet");
24         String accion=req.getParameter("accion");
25
26         switch(accion){
27             case "abrirform":
28                 abrirform(req,resp);
29                 break;
30
31             case "listar":
32                 listar(req,resp);
33                 break;
34
35             case "eliminar":
36                 eliminar(req,resp);
37                 break;
38
39             case "generoEs":
40                 generoEs(req, resp);
41         }
```



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The project structure is the same as the previous screenshot. The 'doPost' method in 'Genero.java' is highlighted, showing a switch statement for actions: 'add' and 'editar'. It also includes a private method 'abrirForm' that forwards the request to a JSP page.

```
src > main > java > controller > Genero.java > Genero > doGet(HttpServletRequest, HttpServletResponse)
48
49 }
50
51 @Override
52 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException
53     System.out.println(x: "Entró al DoPost");
54     String accion=req.getParameter("accion");
55
56     switch(accion){
57         case "add":
58             addGenero(req,resp);
59             break;
60         case "editar":
61             editar(req,resp);
62             break;
63     }
64
65     private void abrirForm(HttpServletRequest req, HttpServletResponse resp) {
66         try{
67             req.getRequestDispatcher("views/genero/formGenero.jsp").forward(req, resp);
68             System.out.println(x: "El formulario ha sido abierto");
69         }catch(Exception e){
70             System.out.println("El formulario NO ha sido abierto"+e.getMessage().toString());
71         }
72     }
73 }
```

- Con un método de retorno hacia el formulario donde en la consola me puede mostrar si se está ejecutando bien o no, esto haciendo en todos los métodos
- Método registro, donde se hace los requerimientos, validan datos previamente, luego de hacer procesos en el dao, nos dará una respuesta que se dirige a un listado de la base de datos.

The screenshot shows the Visual Studio Code editor with the file Explorer on the left. The Explorer shows a project structure with 'src/main/java/controller' containing 'Genero.java'. The main editor displays the 'addGenero' method in 'Genero.java'. The method signature is 'private void addGenero(HttpServletRequest req, HttpServletResponse resp)'. The code inside the method creates a 'GeneroDao' object, a 'GeneroVo' object, and checks for 'nombreGenero' and 'chkeestado' parameters. It sets the 'nombreGenero' and 'estadoGenero' (true/false) and then calls 'gdao.registrar(g)'. After registration, it sends a redirect to 'genero?accion=listar' and prints a success message. It also has a catch block for exceptions, printing an error message.

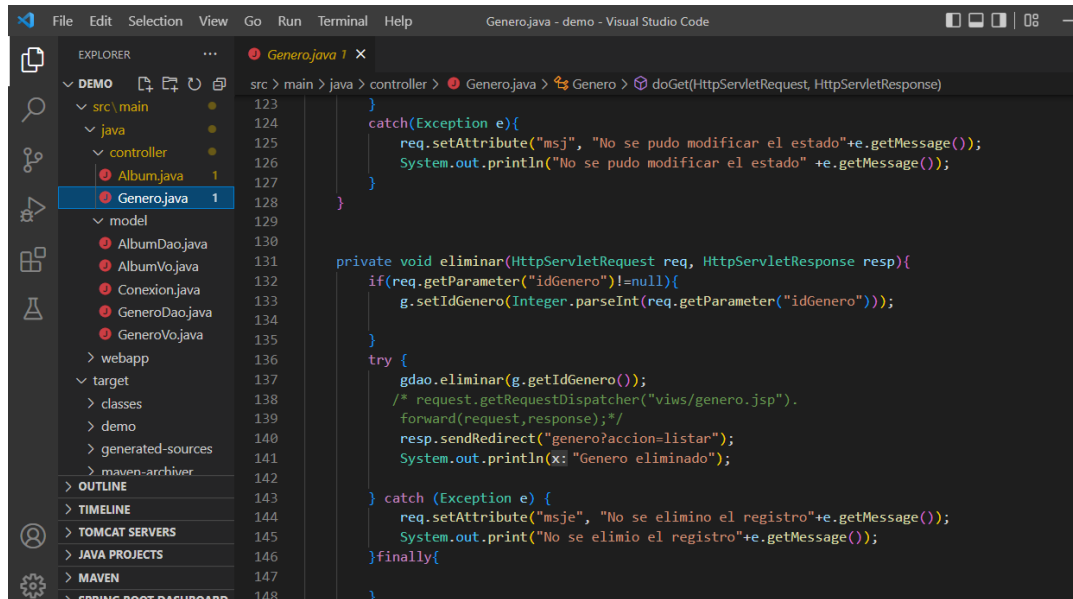
```
src > main > java > controller > Genero.java > Genero > doGet(HttpServletRequest, HttpServletResponse)
72     }
73 }
74
75 private void addGenero(HttpServletRequest req, HttpServletResponse resp) {
76     GeneroDao gdao=new GeneroDao();
77     GeneroVo g=new GeneroVo();
78     if(req.getParameter("nombreGenero")!=null){
79         g.setNombreGenero(req.getParameter("nombreGenero"));
80     }
81     if(req.getParameter("chkeestado")!=null){
82         g.setEstadoGenero(estadoGenero: true);
83     }
84     else{
85         g.setEstadoGenero(estadoGenero: false);
86     }
87     try {
88         gdao.registrar(g);
89         resp.sendRedirect("genero?accion=listar");
90         System.out.println(x: "Registro insertado correctamente");
91     } catch (Exception e) {
92         req.setAttribute("msje","No se pudo registrar el Genero"+e.getMessage());
93         System.out.println("No se pudo registrar el Genero "+e.getMessage().toString());
94     }
95 }
96
97 private void listar(HttpServletRequest req, HttpServletResponse resp) {
```

- En listar, el método permite listar toda la base de datos en una tabla, con las instancias de los modelos, con el Dao quien hará su proceso y si algo sale mal e la consola aparecerá.

The screenshot shows the Visual Studio Code editor with the file Explorer on the left. The Explorer shows a project structure with 'src/main/java/controller' containing 'Genero.java'. The main editor displays the 'listar' and 'generoEs' methods in 'Genero.java'. The 'listar' method signature is 'private void listar(HttpServletRequest req, HttpServletResponse resp)'. It creates a 'GeneroDao' object, a 'GeneroVo' object, and calls 'gdao.listar()' to get a 'List<GeneroVo>'. It sets the 'generos' attribute, forwards the request to 'views/genero/listGenero.jsp', and prints a success message. It also has a catch block for exceptions, printing an error message. The 'generoEs' method signature is 'private void generoEs(HttpServletRequest req, HttpServletResponse resp)'. It checks for 'idGenero' parameter and calls 'g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")))'. It also has a catch block for exceptions, printing an error message.

```
90     System.out.println(x: "Registro insertado correctamente");
91 } catch (Exception e) {
92     req.setAttribute("msje","No se pudo registrar el Genero"+e.getMessage());
93     System.out.println("No se pudo registrar el Genero "+e.getMessage().toString());
94 }
95 }
96
97 private void listar(HttpServletRequest req, HttpServletResponse resp) {
98
99     GeneroDao gdao=new GeneroDao();
100     GeneroVo g=new GeneroVo();
101
102     try {
103         List<GeneroVo> generos=gdao.listar();
104         req.setAttribute("generos", generos);
105         req.getRequestDispatcher("views/genero/listGenero.jsp").forward(req, resp);
106         System.out.println(x: "Generos listados correctamente");
107     } catch (Exception e) {
108         System.out.println("Hay problemas al listar los generos "+e.getMessage().toString());
109     }
110 }
111
112 private void generoEs(HttpServletRequest req, HttpServletResponse resp){
113     if(req.getParameter("idGenero")!=null) {
114         g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")));
115     }
116 }
```

- Eliminar, con el parámetro id y el modelo dao se desarrollara un proceso, quien eliminara esta acción.

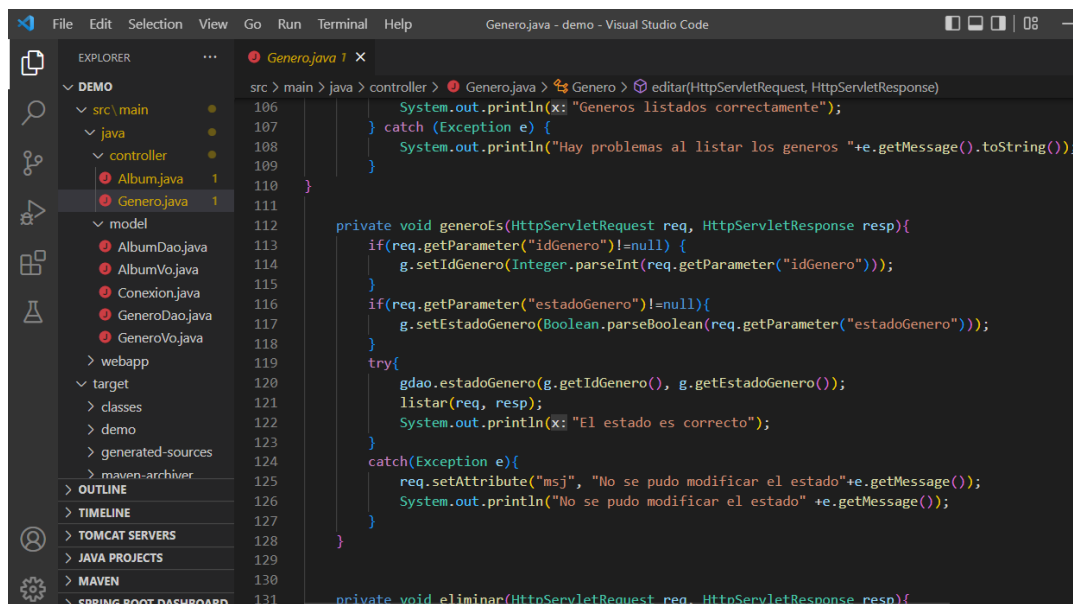


```
File Edit Selection View Go Run Terminal Help Genero.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
    controller
      Album.java
      Genero.java
    model
      AlbumDao.java
      AlbumVo.java
      Conexion.java
      GeneroDao.java
      GeneroVo.java
    webapp
    target
    classes
    demo
    generated-sources
    maven-archiver
  OUTLINE
  TIMELINE
  TOMCAT SERVERS
  JAVA PROJECTS
  MAVEN
  SPRING BOOT DASHBOARD

Genero.java 1
src > main > java > controller > Genero.java > Genero > doGet(HttpServletRequest, HttpServletResponse)
123 }
124 catch(Exception e){
125     req.setAttribute("msj", "No se pudo modificar el estado"+e.getMessage());
126     System.out.println("No se pudo modificar el estado" +e.getMessage());
127 }
128 }
129
130 private void eliminar(HttpServletRequest req, HttpServletResponse resp){
131     if(req.getParameter("idGenero")!=null){
132         g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")));
133     }
134     try {
135         gdao.eliminar(g.getIdGenero());
136         /* request.getRequestDispatcher("viws/genero.jsp").
137         forward(request,response);*/
138         resp.sendRedirect("genero?accion=listar");
139         System.out.println(x: "Genero eliminado");
140     } catch (Exception e) {
141         req.setAttribute("msje", "No se elimino el registro"+e.getMessage());
142         System.out.print("No se elimio el registro"+e.getMessage());
143     }finally{
144
145     }
146 }
147
148 }
```

- Cambiar estado, una vez ya instanciados los modelos, identificar el id, el estado y que en especial sea tipo boolean podremos indicar si quiero que sea activo o inactivo , con el dao, de lo contrario saldrá error ya avisado.



```
File Edit Selection View Go Run Terminal Help Genero.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
    controller
      Album.java
      Genero.java
    model
      AlbumDao.java
      AlbumVo.java
      Conexion.java
      GeneroDao.java
      GeneroVo.java
    webapp
    target
    classes
    demo
    generated-sources
    maven-archiver
  OUTLINE
  TIMELINE
  TOMCAT SERVERS
  JAVA PROJECTS
  MAVEN
  SPRING BOOT DASHBOARD

Genero.java 1
src > main > java > controller > Genero.java > Genero > editar(HttpServletRequest, HttpServletResponse)
106 System.out.println(x: "Generos listados correctamente");
107 } catch (Exception e) {
108     System.out.println("Hay problemas al listar los generos "+e.getMessage().toString());
109 }
110 }
111
112 private void generoEs(HttpServletRequest req, HttpServletResponse resp){
113     if(req.getParameter("idGenero")!=null) {
114         g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")));
115     }
116     if(req.getParameter("estadoGenero")!=null){
117         g.setEstadoGenero(Boolean.parseBoolean(req.getParameter("estadoGenero")));
118     }
119     try{
120         gdao.estadoGenero(g.getIdGenero(), g.getEstadoGenero());
121         listar(req, resp);
122         System.out.println(x: "El estado es correcto");
123     }
124     catch(Exception e){
125         req.setAttribute("msj", "No se pudo modificar el estado"+e.getMessage());
126         System.out.println("No se pudo modificar el estado" +e.getMessage());
127     }
128 }
129
130 private void eliminar(HttpServletRequest req, HttpServletResponse resp){
131     if(req.getParameter("idGenero")!=null){
132         g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")));
133     }
134     try {
135         gdao.eliminar(g.getIdGenero());
136         /* request.getRequestDispatcher("viws/genero.jsp").
137         forward(request,response);*/
138         resp.sendRedirect("genero?accion=listar");
139         System.out.println(x: "Genero eliminado");
140     } catch (Exception e) {
141         req.setAttribute("msje", "No se elimino el registro"+e.getMessage());
142         System.out.print("No se elimio el registro"+e.getMessage());
143     }finally{
144
145     }
146 }
147
148 }
```

- Actualizar, instancias, id, enviar en el listar los de actualizar a Dao, indicara si hay error o no

```
File Edit Selection View Go Run Terminal Help Genero.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
    controller
      Album.java
      Genero.java
  model
    AlbumDao.java
    AlbumVo.java
    Conexion.java
    GeneroDao.java
    GeneroVo.java
  webapp
  target
  classes
  demo
  generated-sources
  maven-archiver
OUTLINE
TIMELINE
TOMCAT SERVERS
JAVA PROJECTS
MAVEN
SPRING BOOT DASHBOARD

src > main > java > controller > Genero.java > Genero > eliminar(HttpServletRequest, HttpServletResponse)

172 private void editar(HttpServletRequest req, HttpServletResponse resp) {
173
174     if(req.getParameter("idGenero")!=null) {
175         g.setIdGenero(Integer.parseInt(req.getParameter("idGenero")));
176
177     if(req.getParameter("nombreGenero")!=null) {
178         g.setNombreGenero(req.getParameter("nombreGenero"));
179     }
180
181     if(req.getParameter("chkeestado")!=null) {
182         g.setEstadoGenero(Boolean.parseBoolean(req.getParameter("chkeestado")));
183     }
184
185     try{
186         gdao.editar (g.getIdGenero(), g.getNombreGenero(), g.getEstadoGenero());
187         listar(req, resp);
188         System.out.println(x; "Actualización correcta");
189     }
190
191     catch(Exception e){
192         req.setAttribute("msje", "No se pudo actualizar el genero" + e.getMessage());
193         System.out.println("No se pudo actualizar el genero" + e.getMessage());
194     }
195
196     finally{
197
```

- Editar, instancias, verificación de formulario, valores en el dao, mensaje de error si es que lo hay

```
File Edit Selection View Go Run Terminal Help Genero.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
    controller
      Album.java
      Genero.java
  model
    AlbumDao.java
    AlbumVo.java
    Conexion.java
    GeneroDao.java
    GeneroVo.java
  webapp
  target
  classes
  demo
  generated-sources
  maven-archiver
OUTLINE
TIMELINE

src > main > java > controller > Genero.java > Genero

54 String accion=req.getParameter("accion");
55
56 switch(accion){
57     case "add":
58         addGenero(req,resp);
59         break;
60     case "editar":
61         editar(req,resp);
62         break;
63 }
64
65 private void abrirForm(HttpServletRequest req, HttpServletResponse resp) {
66     try{
67         req.getRequestDispatcher("views/genero/formGenero.jsp").forward(req, resp);
68         System.out.println(x; "El formulario ha sido abierto");
69     }catch(Exception e){
70         System.out.println("El formulario NO ha sido abierto"+e.getMessage().toString());
71     }
72 }
73
74
```

- Modelos donde se encuentra el VO corresponde a los objetos de datos que se muestran en la interfaz y el DAO clase abierta con atributos privados

```

GeneroDao.java
src > main > java > model > GeneroDao.java
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class GeneroDao {
    //atributos
    Connection con; //objeto para la conexion
    PreparedStatement ps; //objeto para las sentencias preparadas
    ResultSet rs; //objeto para almacenar los resultados de las consultas
    String sql=null; // variable para almacenar las sentencias de SQL
    int r;

    public List<GeneroVo> listar() throws Exception{
        List<GeneroVo> generos=new ArrayList<>();

        sql="SELECT * FROM genero";
        try {
            con=Conexion.conectar();//abrir conexion
            ps=con.prepareStatement(sql);//prepara sentencias select
            System.out.println(ps);
            rs=ps.executeQuery(); // ejecutar la sentencias y guardamos los resultados
            while(rs.next()){
                GeneroVo r=new GeneroVo();
                r.setIdGenero(rs.getInt("columnLabel:idGenero"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return generos;
    }
}

```

```

GeneroVo.java
src > main > java > model > GeneroVo.java
package model;

public class GeneroVo {
    private int idGenero;
    private String nombreGenero;
    private boolean estadoGenero;

    public GeneroVo(){

    }

    public GeneroVo(int idGenero,String nombreGenero,Boolean estadoGenero){
        this.idGenero=idGenero;
        this.nombreGenero=nombreGenero;
        this.estadoGenero=estadoGenero;
    }

    public int getIdGenero() {
        return idGenero;
    }

    public void setIdGenero(int idGenero) {
        this.idGenero = idGenero;
    }

    public String getNombreGenero() {
        return nombreGenero;
    }
}

```

- Base de datos

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0013 segundos.)

SELECT * FROM `genero`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

	idGenero	nombreGenero	estadoGenero
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	18	looo	0
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	20	lina	1
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	22	regaae	1

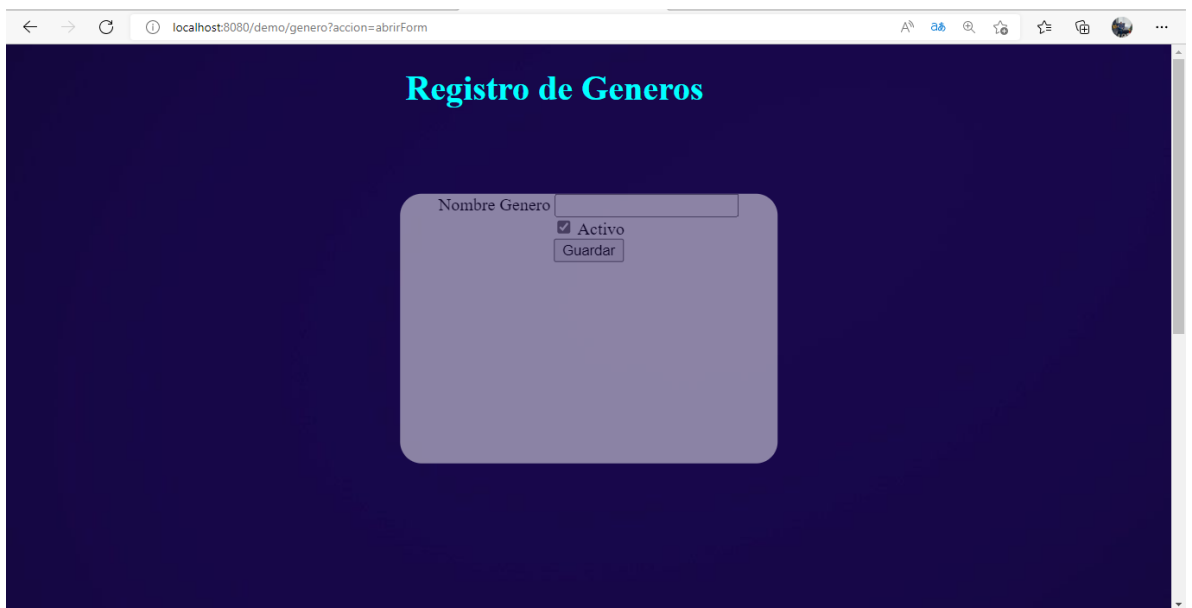
Seleccionar todo | Para los elementos que están marcados: Editar Copiar Borrar Exportar

VISTAS DE GENERO

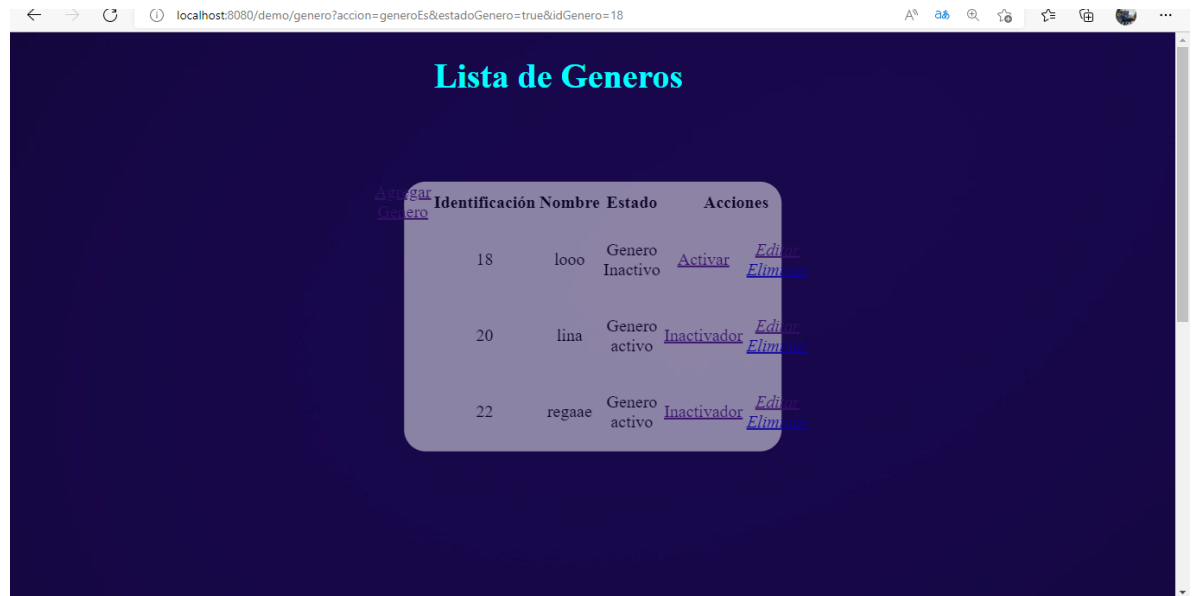
- INDEX



- REGISTRO DE GENERO

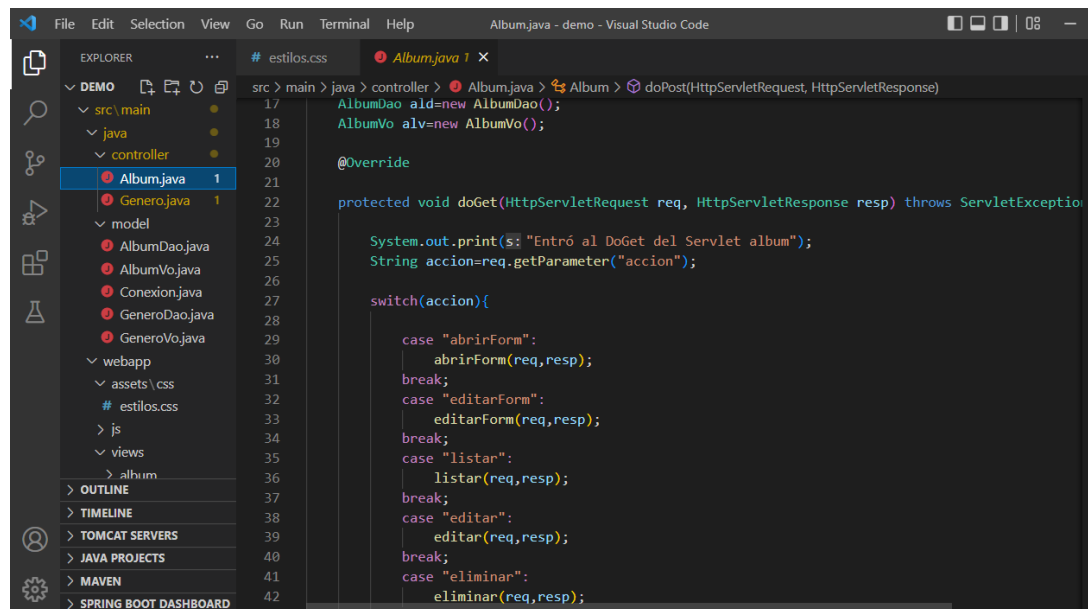


- LISTA DE GENERO



ALBUM

- Controller o servlet cuenta con los métodos do Get quien inicia los argumentos de requerimiento y respuesta y do Post quien también con requerimiento y respuesta para cuando el usuario realice una petición. También se muestra un switch con el que los recorridos, peticiones del usuario.




```

59         case "addAlbum":
60             addAlbum(req, resp);
61             break;
62     }
63 }
64
65 private void listar(HttpServletRequest req, HttpServletResponse resp) {
66
67     AlbumDao ald=new AlbumDao();
68     AlbumVo alv=new AlbumVo();
69
70     try{
71         List<AlbumVo> album = ald.listar();
72         req.setAttribute("albumes", album);
73         req.getRequestDispatcher("views/album/listAlbum.jsp").forward(req, resp);
74         System.out.println(x: "Albumes encontrados");
75     }
76
77     catch(Exception e){
78         req.setAttribute("msje", "No se listar los Albumes" + e.getMessage());
79         System.out.println("No se listar los Albumes" + e.getMessage());
80     }
81 }
82
83 private void editar(HttpServletRequest req, HttpServletResponse resp) {
84

```

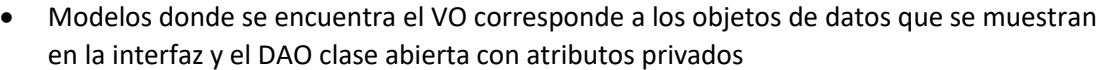
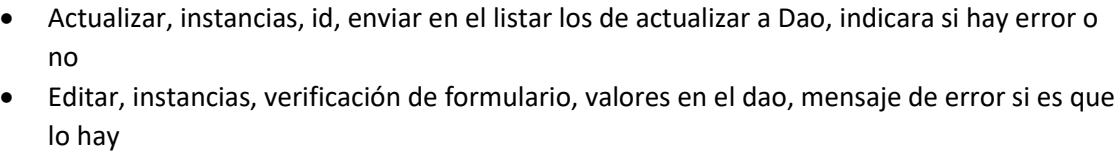
- Eliminar, con el parámetro id y el modelo dao se desarrollara un proceso, quien eliminara esta acción.

```

110     finally{
111     }
112 }
113
114 private void listar(HttpServletRequest req, HttpServletResponse resp) {
115
116     if(req.getParameter("idAlbum")!=null) {
117         alv.setIdAlbum(Integer.parseInt(req.getParameter("idAlbum")));
118     }
119
120     try{
121         ald.eliminar(alv.getIdAlbum());
122         //request.getRequestDispatcher("views/role-edit.jsp").forward(request, response);
123         resp.sendRedirect("album?accion=listar");
124         System.out.println(x: "album Eliminado");
125     }
126
127     catch(Exception e){
128         req.setAttribute("msje", "No se pudo eliminar el registro de album" + e.getMessage());
129         System.out.println("No se pudo eliminar el registro de album" + e.getMessage());
130     }
131 }
132
133 finally{
134 }
135 }

```

- Cambiar estado, una vez ya instanciados los modelos, identificar el id, el estado y que en especial sea tipo boolean podremos indicar si quiero que sea activo o inactivo , con el dao, de lo contrario saldrá error ya avisado.



```
File Edit Selection View Go Run Terminal Help AlbumDao.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
  controller
    Album.java
    Genero.java
  model
    AlbumDao.java
    AlbumVo.java
    Conexion.java
    GeneroDao.java
    GeneroVo.java
  webapp
    assets\css
    # estilos.css
    js
    views
    album

OUTLINE
TIMELINE
TOMCAT SERVERS
JAVA PROJECTS
MAVEN
SPRING BOOT DASHBOARD

# estilos.css AlbumDao.java X
src > main > java > model > AlbumDao.java > AlbumDao > listar()
11 //atributos
12 Connection con; //objeto para la conexion
13 PreparedStatement ps; //objeto para las sentencias preparadas
14 ResultSet rs; //objeto para almacenarlos resultados de las consultas
15 String sql=null; // variable para almacenar las sentecias de SQL
16 int r;
17
18 public List<AlbumVo> listar() throws Exception{
19
20     List<AlbumVo> albums=new ArrayList<>();
21     sql="SELECT * FROM album";
22
23
24     try {
25         con=Conexion.conectar();
26         ps=con.prepareStatement(sql);
27         System.out.println(ps);
28         rs=ps.executeQuery(sql);
29
30         while(rs.next()) {
31             AlbumVo r=new AlbumVo();
32
33             r.setIdAlbum(rs.getInt(columnLabel: "idAlbum"));
34             r.setNombreAlbum(rs.getString(columnLabel: "nombreAlbum"));
35             r.setAnioPublicacion(rs.getInt(columnLabel: "anioPublicacion"));
36             r.setEstadoAlbum(rs.getBoolean(columnLabel: "estadoAlbum"));
```

```
File Edit Selection View Go Run Terminal Help AlbumVo.java - demo - Visual Studio Code

EXPLORER
DEMO
src\main
  java
  controller
    Album.java
    Genero.java
  model
    AlbumDao.java
    AlbumVo.java
    Conexion.java
    GeneroDao.java
    GeneroVo.java
  webapp
    assets\css
    # estilos.css
    js
    views
    album

OUTLINE
TIMELINE
TOMCAT SERVERS
JAVA PROJECTS
MAVEN
SPRING BOOT DASHBOARD

# estilos.css AlbumVo.java X
src > main > java > model > AlbumVo.java > AlbumVo > nombreAlbum
1 package model;
2
3 public class AlbumVo {
4
5
6     private int idAlbum;
7     private String nombreAlbum;
8     private int anioPublicacion;
9     private Boolean estadoAlbum;
10
11     public AlbumVo() {
12     }
13
14     public AlbumVo(int idAlbum, String nombreAlbum, int anioPublicacion,
15         Boolean estadoAlbum) {
16         this.idAlbum = idAlbum;
17         this.nombreAlbum = nombreAlbum;
18         this.anioPublicacion = anioPublicacion;
19         this.estadoAlbum = estadoAlbum;
20     }
21
22     public int getIdAlbum() {
23         return idAlbum;
24     }
25
26     public void setIdAlbum(int idAlbum) {
```

- Base de datos

localhost/phpmyadmin/index.php?route=/sql&server=1&db=disquera&table=album&pos=0

phpMyAdmin

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Más

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0011 segundos.)

`SELECT * FROM `album``

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

+ Opciones

		idAlbum	nombreAlbum	añoPublicacion	estadoAlbum		
<input type="checkbox"/>	Editar	Copiar	Borrar	13	love	2020	1
<input type="checkbox"/>	Editar	Copiar	Borrar	14	123	2019	1
<input type="checkbox"/>	Editar	Copiar	Borrar	15	hola	1999	1
<input type="checkbox"/>	Editar	Copiar	Borrar	16	lina	1999	1

VISTA ALBUM

- REGISTRO ALBUM

localhost:8080/demo/album?action=abrirForm

Traducido

Registro de Album

Album Nombre

Año Publicacion

☒ Activo

- LISTA ALBUM

Lista de Album

Identificación	Álbum Nombre	Año Publicacion	Estado	Acciones
13	amar	2020	Álbum Activo	Inactivador Editar Eliminar
14	123	2019	Álbum Activo	Inactivador Editar Eliminar
15	hola	1999	Álbum Activo	Inactivador Editar Eliminar
16	lina	1999	Álbum Activo	Inactivador Editar Eliminar