

## Controladores

```
src > main > java > controller > servletalbum.java > servletalbum > editar(HttpServletRequest, HttpServletResponse)
1  package controller;
2
3  import java.io.IOException;
4  import java.util.List;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.http.HttpServlet;
8  import javax.servlet.http.HttpServletRequest;
9  import javax.servlet.http.HttpServletResponse;
10
11 import models.albumDao;
12 import models.albumVo;
13
```

Acá importo las librerías necesarias

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
    System.out.println("DoGet");
    String accion=req.getParameter("accion");

    switch(accion){
        case "formulario":
            formulario(req,resp);
            break;
        case "listar":
            listar(req,resp);
            break;
        case "eliminar":
            eliminar(req,resp);
            break;
        case "cambioestado":
            cambioestado(req,resp);
            break;
        case "index":
            req.getRequestDispatcher("index.jsp").forward(req, resp);
            break;
        case "edicion":
            edicion(req,resp);
            break;
    }
}
```

Hago un método doGet para devolver los datos, Luego un Switch y dentro de esto defino los casos que quiero que se puedan devolver según lo pedido

```

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
    System.out.println("DoPost");
    String accion=req.getParameter("accion");

    switch(accion){
        case "add":
            add(req,resp);
            break;
        case "editar":
            editar(req,resp);
            break;
    }
}

```

Hago el doPost para poder hacer las peticiones que se piden

```

private void listar(HttpServletRequest req, HttpServletResponse resp) {
    try {
        List<albumVo>album=gDao.listar();
        req.setAttribute("Album", album);
        req.getRequestDispatcher("views/listarAlbum.jsp").forward(req, resp);
        System.out.println("Datos listados de genero");
    } catch (Exception e) {
        System.out.println("Hay problemas al listar los datos "+e.getMessage().toString());
    }
}

private void formulario(HttpServletRequest req, HttpServletResponse resp) {
    try{
        req.getRequestDispatcher("views/agregarAlbum.jsp").forward(req, resp);
        System.out.println("El formulario abrio");
    }catch(Exception e){
        System.out.println("El formulario no abre"+e.getMessage().toString());
    }
}

```

Acá hago unos atributos privados donde hago un try que me lleva las instrucciones que son llamar a la vista listarAlbum o agregarAlbum, Luego el catch donde imprimo un mensaje de error y que me muestre la excepción o el error

```

private void cambioestado(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("id")!=null){
        gVo.setIdalbum(Integer.parseInt(req.getParameter("id")));
    }
    if(req.getParameter("estado")!=null){
        gVo.setEstadoalbum(Boolean.parseBoolean(req.getParameter("estado")));
    }
    try {
        gDao.estado(gVo.getEstadoalbum(), gVo.getIdalbum());
        System.out.println("El estado se cambio");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error no cambio el estado "+e.getMessage().toString());
    }
}

```

Hago un atributo privado donde defino unos if con los parámetros “id” y “estado” donde los if me dicen que los parámetros tienen que arrojar un resultado para que se puedan validar y se puedan llenar y hago un try para que la instrucción se pueda validar o el catch me muestre el error que ocurrió

```

private void eliminar(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("id")!=null){
        gVo.setIdalbum(Integer.parseInt(req.getParameter("id")));
    }
    try {
        gDao.eliminar(gVo.getIdalbum());
        System.out.println("Se elimino el genero");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error al eliminar genero "+e.getMessage().toString());
    }
}

```

Hago un atributo privado, Donde el if tiene la condición para poder hacer un resultado si es null o no, Y el try me da las instrucciones para que se pueda realizar lo pedido y si no ocurre lo pedido, El catch me muestra el error que pudo haber sucedido

```

private void add(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("nombre")!=null){
        gVo.setNombrealbum(req.getParameter("nombre"));
    }
    if(req.getParameter("chekeestado")!=null){
        gVo.setEstadoalbum(estadoalbum: true);
    }
    else{
        gVo.setEstadoalbum(estadoalbum: false);
    }
    if(req.getParameter("apublic")!=null){
        gVo.setapublicacionalbum(Integer.parseInt(req.getParameter("apublic")));
    }
    try {
        gDao.registrar(gVo);
        System.out.println("Se registro genero");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error en el registro de genero "+e.getMessage().toString());
    }
}

```

Hago un atributo privado donde están unos if y un else donde si el parámetro esta vacío toma el camino del else y si no está vacío toma el camino del if y el try esta para que me pueda cumplir las instrucciones y el catch esta para que me pueda mostrar algún error

```

private void editar(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("id")!=null){
        gVo.setIdalbum(Integer.parseInt(req.getParameter("id")));
    }
    if(req.getParameter("apublic")!=null){
        gVo.setapublicacionalbum(Integer.parseInt(req.getParameter("apublic")));
    }
    if(req.getParameter("nombre")!=null){
        gVo.setNombrealbum(req.getParameter("nombre"));
    }
    if(req.getParameter("chekeestado")!=null){
        gVo.setEstadoalbum(Boolean.parseBoolean(req.getParameter("chekeestado")));
    }
    try {
        gDao.editar(gVo.getEstadoalbum(), gVo.getIdalbum(), gVo.getNombrealbum(), gVo.getapublicacionalbum());
        System.out.println("Se actualizo el album");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error al actualizar el album "+e.getMessage().toString());
    }
}

```

Hago un atributo privado donde están los if, Quiere decir que si el valor es nulo nos manda por el camino del if y si no es nulo nos mandara por un camino diferente

El try hace que si el valor sea diferente a nulo nos haga cumplir la instrucción pedida y el catch nos mostrara si hay alguna excepción

```
private void edicion(HttpServletRequest req, HttpServletResponse resp) {
    try {
        if(req.getParameter("id")!=null){
            req.setAttribute("id", req.getParameter("id"));
        }
        if(req.getParameter("apublicacionalbum")!=null){
            req.setAttribute("apublicacionalbum", req.getParameter("apublicacionalbum"));
        }
        if(req.getParameter("nombre")!=null){
            req.setAttribute("nombre", req.getParameter("nombre"));
        }
        if(req.getParameter("chekeestado")!=null){
            req.setAttribute("chekeestado", req.getParameter("chekeestado"));
        }

        req.getRequestDispatcher("views/editaralbum.jsp").forward(req, resp);
        System.out.println("Datos listados de genero");
    } catch (Exception e) {
        System.out.println("Hay problemas al listar los datos "+e.getMessage().toString());
    }
}
```

Hago un atributo privado donde están los if, Quiere decir que si el valor es nulo nos manda por el camino del if y si no es nulo nos mandara por un camino diferente. El try hace que si el valor sea diferente a nulo nos haga cumplir la instrucción pedida y el catch nos mostrara si hay alguna excepción

```
package controller;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import models.generoDao;
import models.generoVo;
```

```
public class servletgenero extends HttpServlet{

    generoVo gVo=new generoVo();
    generoDao gDao=new generoDao();

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
    {
        System.out.println("DoGet");
        String accion=req.getParameter("accion");
        System.out.println(accion);

        switch(accion){
            case "formulario":
                formulario(req,resp);
                break;
            case "listar":
                listar(req,resp);
                break;
            case "eliminar":
                eliminar(req,resp);
                break;
            case "cambioestado":
                cambioestado(req,resp);
                break;
            case "index":
                req.getRequestDispatcher("index.jsp").forward(req, resp);
        }
    }
}
```

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException {
    System.out.println("DoPost");
    String accion=req.getParameter("accion");
    System.out.println(accion);
    switch(accion){
        case "add":
            add(req,resp);
            break;
    }
}

private void listar(HttpServletRequest req, HttpServletResponse resp) {
    try {
        List<generoVo> genero=gDao.listar();
        req.setAttribute("Generos", genero);
        req.getRequestDispatcher("views/pagina2.jsp").forward(req, resp);
        System.out.println("Datos listados de genero");
    } catch (Exception e) {
        System.out.println("Hay problemas al listar los datos "+e.getMessage().toString());
    }
}
```

```

private void formulario(HttpServletRequest req, HttpServletResponse resp) {
    try{
        req.getRequestDispatcher("views/pagina1.jsp").forward(req, resp);
        System.out.println("El formulario abrio");
    }catch(Exception e){
        System.out.println("El formulario no abre"+e.getMessage().toString());
    }
}

private void cambioestado(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("id")!=null){
        gVo.setIdgenero(Integer.parseInt(req.getParameter("id")));
    }
    if(req.getParameter("estado")!=null){
        gVo.setEstadogenero(Boolean.parseBoolean(req.getParameter("estado")));
    }
    try {
        gDao.estado(gVo.getEstadogenero(), gVo.getIdgenero());
        System.out.println("El estado se cambio");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error no cambio el estado "+e.getMessage().toString());
    }
}
}

```

```

private void eliminar(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("id")!=null){
        gVo.setIdgenero(Integer.parseInt(req.getParameter("id")));
    }
    try {
        gDao.eliminar(gVo.getIdgenero());
        System.out.println("Se elimino el genero");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error al eliminar genero "+e.getMessage().toString());
    }
}
}

```

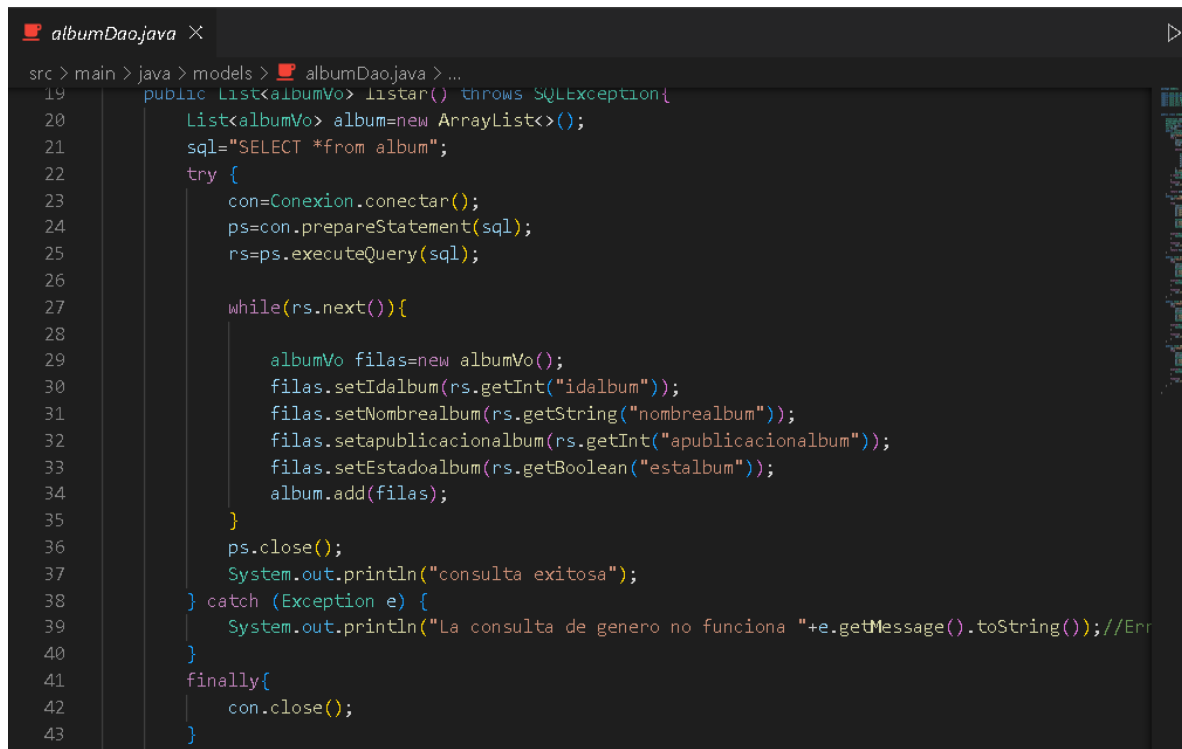


```

private void add(HttpServletRequest req, HttpServletResponse resp) {
    if(req.getParameter("nombre")!=null){
        gVo.setNombregenero(req.getParameter("nombre"));
    }
    if(req.getParameter("chekeestado")!=null){
        gVo.setEstadogenero(estadogenero: true);
    }
    else{
        gVo.setEstadogenero(estadogenero: false);
    }
    try {
        gDao.registrar(gVo);
        System.out.println("Se registro genero");
        listar(req, resp);
    } catch (Exception e) {
        System.out.println("Error en el registro de genero "+e.getMessage().toString());
    }
}
}

```

Modelos:



```

albumDao.java X
src > main > java > models > albumDao.java > ...
19 public List<albumVo> listar() throws SQLException{
20     List<albumVo> album=new ArrayList<>();
21     sql="SELECT *from album";
22     try {
23         con=Conexion.conectar();
24         ps=con.prepareStatement(sql);
25         rs=ps.executeQuery(sql);
26
27         while(rs.next()){
28
29             albumVo filas=new albumVo();
30             filas.setIdalbum(rs.getInt("idalbum"));
31             filas.setNombrealbum(rs.getString("nombrealbum"));
32             filas.setapublicacionalbum(rs.getInt("apublicacionalbum"));
33             filas.setEstadoalbum(rs.getBoolean("estalbum"));
34             album.add(filas);
35         }
36         ps.close();
37         System.out.println("consulta exitosa");
38     } catch (Exception e) {
39         System.out.println("La consulta de genero no funciona "+e.getMessage().toString());
40     }
41     finally{
42         con.close();
43     }
44 }

```

Acá lo que hago es llamar a List y seleccionar la tabla album que se encuentra en la base de datos, En el try hago la conexión a la base de datos, En el while hago el ciclo para que me pueda mostrar cada parámetro que ya fue definido, El catch me muestra que no se pudo hacer consulta y me muestra la excepción

```
public int registrar(albumVo album) throws SQLException{
    sql="INSERT INTO album(nombrealbum,apublicacionalbum,estalbum) values(?,?,?)";

    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        ps.setString(1, album.getNombrealbum());
        ps.setInt(2, album.getapublicacionalbum());
        ps.setBoolean(3, album.getEstadoalbum());

        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se registro el genero");

    }catch(Exception e){
        System.out.println("Error al registrarse "+e.getMessage().toString());
    }

    finally{
        con.close();
    }

    return filas;
}
```

Lo que hago es llamar a registrar y llamo los espacios como se están llamando en la base de datos, En el try hago la conexión a la base de datos además llamo los espacios que necesito para poder llenar, El catch me muestra si hay alguna excepción con el problema y el finally me ejecuta así haya alguna excepción

```

public void eliminar(int id) throws SQLException{
    sql="DELETE FROM album WHERE idalbum="+id;
    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se elimino el album");
    }catch(Exception e){
        System.out.println("Error al eliminar album "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
}

```

Acá llamo a eliminar y hago la sentencia sql donde llamo a la tabla álbum y selecciono eliminar, En el try conecto a la base de datos y hago el proceso para eliminar, El catch me muestra si hay alguna excepción con el error

```

public void estado(Boolean estadoalbum, int idalbum) throws SQLException{
    sql="UPDATE album SET estalbum="+estadoalbum+" WHERE idalbum="+idalbum;
    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se cambio el estado XD");
    }catch(Exception e){
        System.out.println("Error al cambiar de estado "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
}

```

Acá lo que hago es llamar a la tabla álbum con los parámetros a pedir para cambiar el estado, el try me hace la conexión a la base de datos y me hace la instrucción, El catch me lanza una excepción con el error

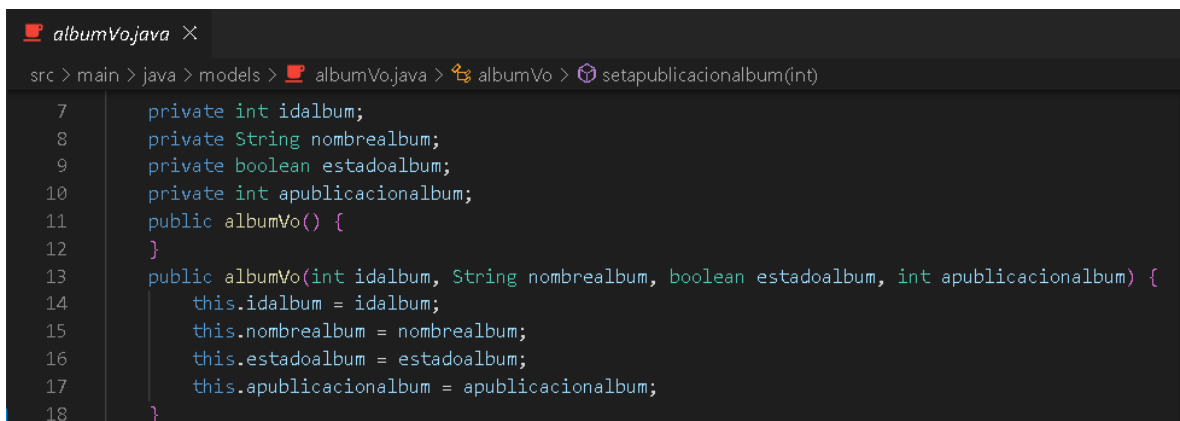
```

public void editar(Boolean estadoalbum, int idalbum, String nombrealbum, int apublicacionalbum)
    sql="UPDATE album SET nombrealbum='"+nombrealbum+"',apublicacionalbum='"+apublicacionalbum+"
    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se edito exitosamente");
    }catch(Exception e){
        System.out.println("Error al editarse "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
}

```

Lo que hago es llamar a editar con sus parámetros y a la tabla album que esta en la base de datos con los nombres ya específicos en la base de datos, El try me hace la conexión a la base de datos y me hace la instrucción para editar, El catch me lanza la excepción con el error



```

albumVo.java
src > main > java > models > albumVo.java > albumVo > setapublicacionalbum(int)
7     private int idalbum;
8     private String nombrealbum;
9     private boolean estadoalbum;
10    private int apublicacionalbum;
11    public albumVo() {
12    }
13    public albumVo(int idalbum, String nombrealbum, boolean estadoalbum, int apublicacionalbum) {
14        this.idalbum = idalbum;
15        this.nombrealbum = nombrealbum;
16        this.estadoalbum = estadoalbum;
17        this.apublicacionalbum = apublicacionalbum;
18    }

```

Acá llamo todos los atributos privados con su tipo de datos. Luego llamo a albumVo con sus parámetros. Con los this hago la referencia al objeto instanciado

```
public int getIdalbum() {  
    return idalbum;  
}  
public void setIdalbum(int idalbum) {  
    this.idalbum = idalbum;  
}  
public String getNombrealbum() {  
    return nombrealbum;  
}  
public void setNombrealbum(String nombrealbum) {  
    this.nombrealbum = nombrealbum;  
}  
public boolean getEstadoalbum() {  
    return estadoalbum;  
}  
public void setEstadoalbum(boolean estadoalbum) {  
    this.estadoalbum = estadoalbum;  
}  
public int getapublicacionalbum() {  
    return apublicacionalbum;  
}  
public void setapublicacionalbum(int apublicacionalbum) {  
    this.apublicacionalbum = apublicacionalbum;  
}
```

Acá llamo a cada valor con su parámetro get y luego llamo a cada clase instanciada para que sea un valor igual

```
generoDao.java X
src > main > java > models > generoDao.java > {} models
18 public List<generoVo> listar() throws SQLException{
19     List<generoVo> genero=new ArrayList<>();
20     sql="SELECT * FROM genero";
21     try {
22         con=Conexion.conectar();
23         ps=con.prepareStatement(sql);
24         System.out.println(ps);
25         rs=ps.executeQuery(sql);
26
27         while(rs.next()){
28
29             generoVo filas=new generoVo();
30             filas.setIdgenero(rs.getInt("idgenero"));
31             filas.setNombregenero(rs.getString("nombregenero"));
32             filas.setEstadogenero(rs.getBoolean("estgenero"));
33             genero.add(filas);
34         }
35         ps.close();
36         System.out.println("consulta exitosa");
37     } catch (Exception e) {
38         System.out.println("La consulta de genero no funciona "+e.getMessage().toString());//E
39     }
40     finally{
41         con.close();
42     }
43 }
```

Acá lo que hago es llamar a List y seleccionar la tabla genero que se encuentra en la base de datos, En le try hago la conexión a la base de datos, En el while hago el ciclo para que me pueda mostrar cada parámetro que ya fue definido, El catch me muestra que no se pudo hacer consulta y me muestra la excepción

```

public int registrar(generoVo genero) throws SQLException{
    sql="INSERT INTO genero(nombregenero,estgenero) values(?,?)";

    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        ps.setString(1, genero.getNombregenero());
        ps.setBoolean(2, genero.getEstadogenero());
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se registro el genero");

    }catch(Exception e){
        System.out.println("Error al registrarse "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
    return filas;
}

```

Lo que hago es llamar a registrar y llamo los espacios como se están llamando en la base de datos, En el try hago la conexión a la base de datos además llamo los espacios que necesito para poder llenar, El catch me muestra si hay alguna excepción con el problema y el finally me ejecuta así haya alguna excepción

```

public void eliminar(int id) throws SQLException{
    sql="DELETE FROM genero WHERE idgenero="+id;
    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se elimino el genero");

    }catch(Exception e){
        System.out.println("Error al eliminar genero "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
}

```

Acá llamo a eliminar y hago la sentencia sql donde llamo a la tabla genero y

selecciono eliminar, En el try conecto a la base de datos y hago el proceso para eliminar, El catch me muestra si hay alguna excepción con el error

```
public void estado(Boolean estadogenero, int idgenero) throws SQLException{
    sql="UPDATE genero SET estogenero="+estadogenero+" WHERE idgenero="+idgenero;
    try{
        con=Conexion.conectar();
        ps=con.prepareStatement(sql);
        System.out.println(ps);
        ps.executeUpdate();
        ps.close();
        System.out.println("Se cambio el estado XD");
    }catch(Exception e){
        System.out.println("Error al cambiar de estado "+e.getMessage().toString());
    }

    finally{
        con.close();
    }
}
```

Acá lo que hago es llamar a la tabla genero con los parámetros a pedir para cambiar el estado, el try me hace la conexión a la base de datos y me hace la instrucción, El catch me lanza una excepción con el error

```
generoVo.java X
src > main > java > models > generoVo.java > {} models
1  package models;
2
3  public class generoVo {
4      private int idgenero;
5      private String nombregenero;
6      private boolean estadogenero;
7      public generoVo() {
8      }
9      public generoVo(int idgenero, String nombregenero, boolean estadogenero) {
10         this.idgenero = idgenero;
11         this.nombregenero = nombregenero;
12         this.estadogenero = estadogenero;
13     }
```

Acá llamo todos los atributos privados con su tipo de datos. Luego llamo a albumVo con sus parámetros. Con los this hago la referencia al objeto instanciado



```
public int getIdgenero() {  
    return idgenero;  
}  
public void setIdgenero(int idgenero) {  
    this.idgenero = idgenero;  
}  
public String getNombregenero() {  
    return nombregenero;  
}  
public void setNombregenero(String nombregenero) {  
    this.nombregenero = nombregenero;  
}  
public boolean getEstadogenero() {  
    return estadogenero;  
}  
public void setEstadogenero(boolean estadogenero) {  
    this.estadogenero = estadogenero;  
}  
public void setEstadoalbum(boolean b) {  
}  
public void setNombrealbum(String string) {  
}
```

Acá llamo a cada valor con su parámetro get y luego llamo a cada clase instanciada para que sea un valor igual

Vistas

```

<%@include file="header.jsp" %>
<div class="container">
  <div class="col-12">
    <a type="submit" href="servletalbum?accion=index" class="btn btn-primary"> <-Atras</a>
  </div>
  <h1>Registro de album</h1>

  <form action="servletalbum" method="post">
    <div class="form-group">
      <label form="nombre">Nombre album</label>
      <input type="text" name="nombre" id="nombre" class="form-control">
    </div>
    <br>
    <br>
    <div class="form-group">
      <label form="año">Anio de publicacion album</label>
      <input type="number" name="apublic" id="apublic" class="form-control">
    </div>
    <br>
    <br>
    <div class="form-check">
      <input type="checkbox" name="chekeestado" id="chekeestado" checked class="form-check-input" />
      <label for="chekeestado">Activo</label>
    </div>
    <br>
    <div>
      <button type="submit" class="btn btn-primary" name="accion" value="add">Guardar</button>
    </div>
  </form>
</div>
<%@include file="footer.jsp" %>

```

Primero hago un div, Dentro de ese div hago un botom para retroceder al principio de de la pagina

Hago un h1 donde va el titulo

Hago otro div, Dentro del div hago un label que me define el nombre del album

Hago otro div, Dentro del div hago un label que me define el año de publicacion del album

Hago otro div, Dentro del div hago un label que me define el estado del album

Hago otro div, Dentro del div hago un botom

```

listarAlbum.jsp X
src > main > webapp > views > listarAlbum.jsp
1  <%@include file="header.jsp" %>
2  <div class="container">
3      <div class="col-12">
4          <a type="submit" href="servletalbum?accion=index" class="btn btn-primary"> <-Atras</a>
5      </div>
6
7      <br>
8      <br>
9
10     <table class="table">
11         <thead>
12             <tr>
13                 <th scope="col">IdAlbum</th>
14                 <th scope="col">NombreAlbum</th>
15                 <th scope="col">APublicacionAlbum</th>
16                 <th scope="col">EstAlbum</th>
17                 <th scope="col">Accion</th>
18             </tr>
19         </thead>
20
21         <c:forEach var="album" items="${Album}">
22             <tr>
23                 <td>${album.getIdalbum()}</td>
24                 <td>${album.getNombrealbum()}</td>
25                 <td>${album.getapublicacionalbum()}</td>
26
27                 <c:forEach var="album" items="${Album}">
28                     <tr>
29                         <td>${album.getIdalbum()}</td>
30                         <td>${album.getNombrealbum()}</td>
31                         <td>${album.getapublicacionalbum()}</td>
32
33                         <c:if test="${album.getEstadoalbum() == true}">
34                             <td><samp> Genero Activo</samp></td>
35                         </c:if>
36
37                         <c:if test="${album.getEstadoalbum() == false}">
38                             <td><samp>Genero Inactivo</samp></td>
39                         </c:if>
40
41                         <c:if test="${album.getEstadoalbum() == true}">
42                             <td>
43                                 <a href="servletalbum?accion=cambioestado&id=${album.getIdalbum()}&estado=false"><button class="button" type="button">
44                                     Genero Inactivo
45                                 </button>
46                             </td>
47                         </c:if>
48
49                         <c:if test="${album.getEstadoalbum() == false}">
50                             <td>
51                                 <a href="servletalbum?accion=cambioestado&id=${album.getIdalbum()}&estado=true"><button class="button" type="button">
52                                     Genero Activo
53                                 </button>
54                             </td>
55                         </c:if>
56
57                         <td>
58                             <a href="servletalbum?accion=eliminar&id=${album.getIdalbum()}"><button class="button" type="button"> Eliminar
59                         </td>
60                     </tr>
61                 </c:forEach>
62             </tr>
63         </c:forEach>
64     </table>
65 </div>
66 </div>

```

Primero hago un div, Dentro del div hago un botom qqueme lleve a la pagina principal

Hago una tabla y dentro los tr para las filas y los th para los encabezados

Luego unos forEach para traer los atributos

Luego creo los botones con sus opciones cada uno

```
pagina1.jsp X
src > main > webapp > views > pagina1.jsp
1  <%@include file="header.jsp" %>
2  <div class="container">
3      <div class="col-12">
4          <a type="submit" href="servletgenero?accion=index" class="btn btn-primary"> <-Atras</a>
5      </div>
6      <h1>Registro de genero</h1>
7
8      <form action="servletgenero" method="post">
9
10         <div class="form-group">
11             <label form="nombre">Nombre genero</label>
12             <input type="text" name="nombre" id="nombre" class="form-control">
13         </div>
14         <br>
15         <br>
16         <div class="form-check">
17             <input type="checkbox" name="chekeestado" id="chekeestado" checked class="form-check-input"
18             <label for="chekeestado">Activo</label>
19         </div>
20
21         <div>
22             <button type="submit" class="btn btn-primary" name="accion" value="add">Guardar</button>
23         </div>
24     </form>
25
26 </div>
```

Primero creo un div, Dentro del div un botom que me lleva a la pagina principal

Un h1 que es el titulo

Hago otro div, Dentro del div hago un label que me define el nombre del genero

Hago otro div, Dentro del div hago un label que me define el estado del genero

Hago otro div, Dentro del div hago un botom

```
index.jsp X
src > main > webapp > index.jsp
1  <%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <link rel="stylesheet" href="assets/css/estilos.css">
9      <title>Login</title>
10 </head>
11
12 <body>
13 <div>
14     
15 </div>
16 <form class="row g-3">
17     <div class="col-md-6">
18         <label for="inputEmail4" class="form-label">Correo</label>
19         <input type="email" class="form-control" id="inputEmail4" placeholder="Nombre@gmail.com">
20     </div>
21     <br>
22     <div class="col-md-6">
23         <label for="inputPassword4" class="form-label">Contraseña</label>
24         <input type="password" class="form-control" id="inputPassword4">
25     </div>
26 </form>
```

```
<div class="col-12">
    <button type="submit" class="btn btn-primary">Entrar</button>
</div>
<br>
<div class="col-12">
    <a type="submit" href="servletgenero?accion=formulario" class="btn btn-primary">Agregar usuario</a>
</div>

<br>
<div class="col-12">
    <a type="submit" href="servletgenero?accion=listar" class="btn btn-primary">Tabla</a>
</div>
<br>

<div class="col-12">
    <a type="submit" href="servletaalbum?accion=formulario" class="btn btn-primary">Agregar album</a>
</div>

<div class="col-12">
    <a type="submit" href="servletaalbum?accion=listar" class="btn btn-primary">Tabla</a>
</div>
<a>DESARROLLADO POR:SEBASTIAN MORALES MORALES</a><br>
```

Primero pongo una imagen

Hago otro div, Dentro del div hago un label que me define el correo

Hago otro div, Dentro del div hago un label que me define la contraseña

Luego pongo unos botones que me redireccionan según lo indicado

