

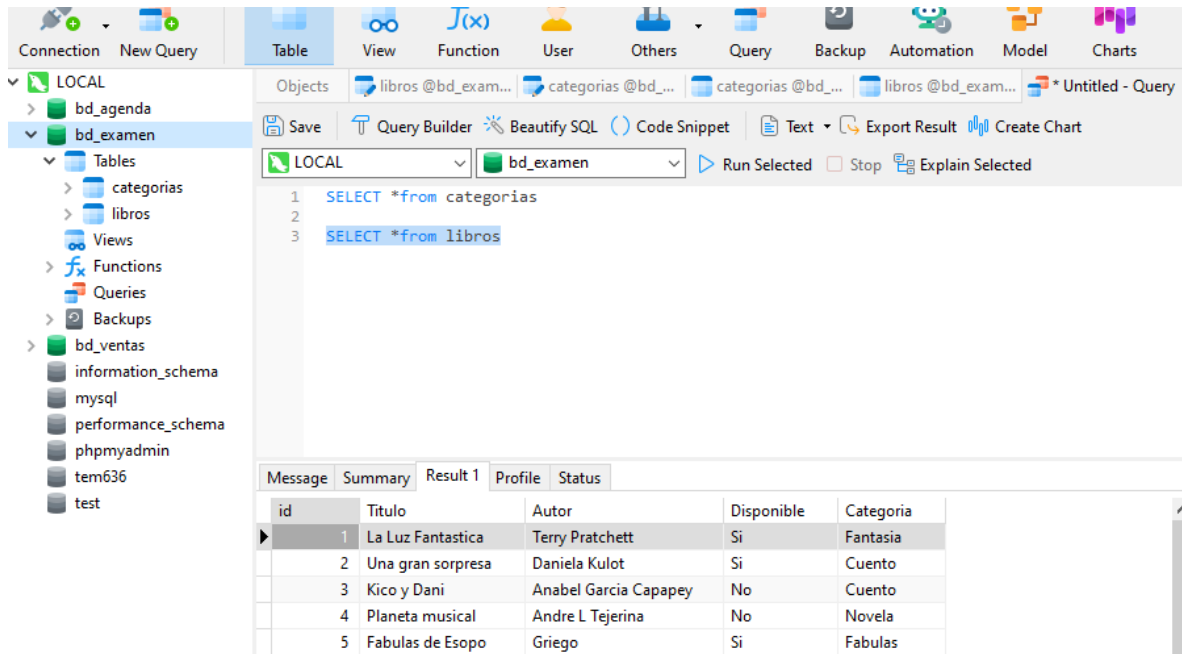
## SISTEMAS EMERGENTES II – SEGUNDO PARCIAL PARCIAL

**NOMBRE: MARIA EUGENIA CHAMBI MOLINA**

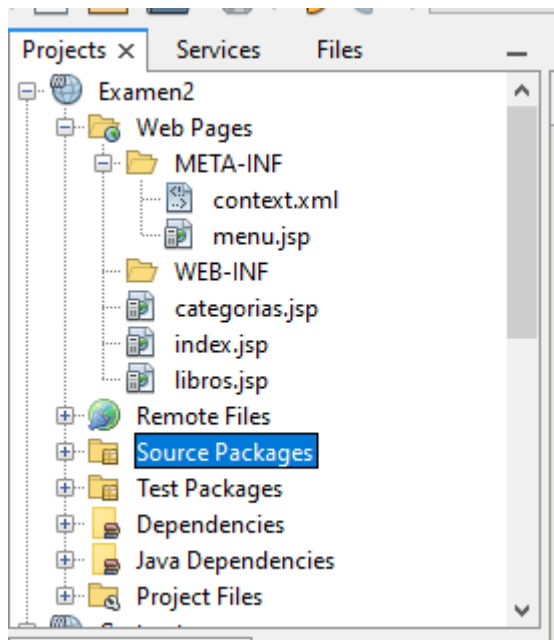
**C.I.: 4979295**

**CODIGO:**

**CREANDO MI BASE DE DATOS NAVICAT:**

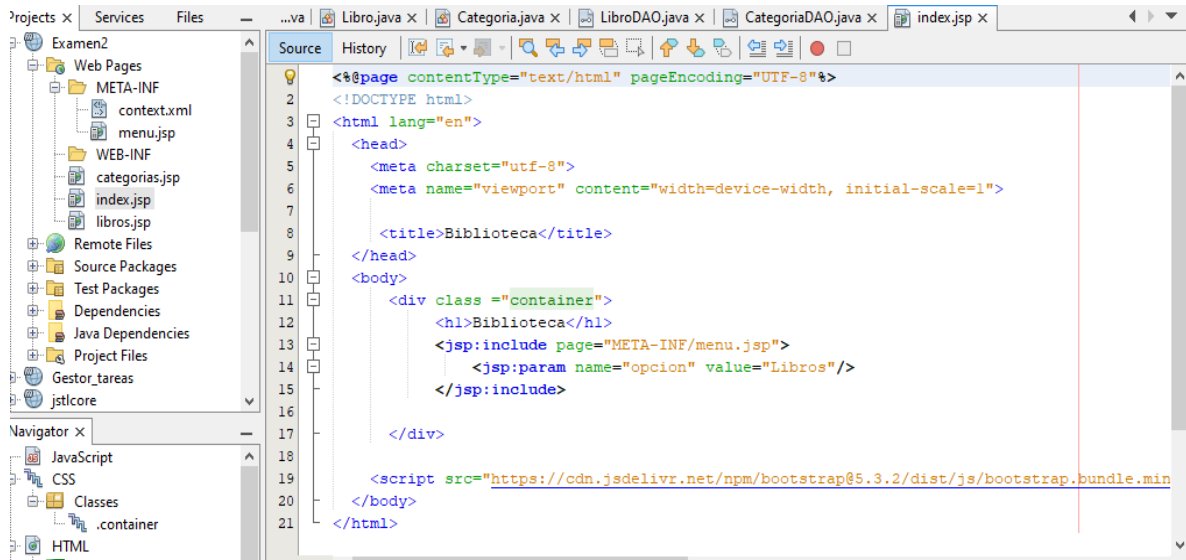


**CREANDO MI PROYECTO EN NETBEANS:**



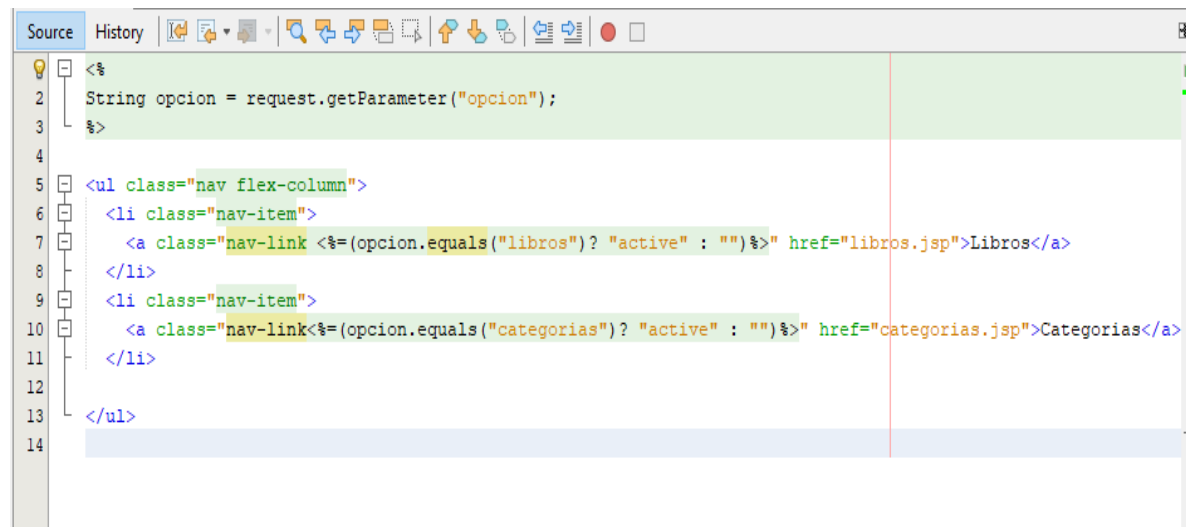
## CREANDO INDEX, MENU :

### INDEX.JSP



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7
8 <title>Biblioteca</title>
9 </head>
10 <body>
11 <div class="container">
12 <h1>Biblioteca</h1>
13 <jsp:include page="META-INF/menu.jsp">
14 <jsp:param name="opcion" value="Libros"/>
15 </jsp:include>
16
17 </div>
18
19 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min"
20 </body>
21 </html>
```

### MENU.JSP



```
1 <%
2 String opcion = request.getParameter("opcion");
3 %>
4
5 <ul class="nav flex-column">
6 <li class="nav-item">
7 <a class="nav-link" <%= (opcion.equals("libros")) ? "active" : "" %> href="libros.jsp">Libros</a>
8 </li>
9 <li class="nav-item">
10 <a class="nav-link" <%= (opcion.equals("categorias")) ? "active" : "" %> href="categorias.jsp">Categorias</a>
11 </li>
12
13 </ul>
14
```

# Biblioteca

- [Libros](#)
- [Categorias](#)

## AHORA AÑADIMOS la TABLAS A LIBROS

### CODIGO LIBRO.jsp:

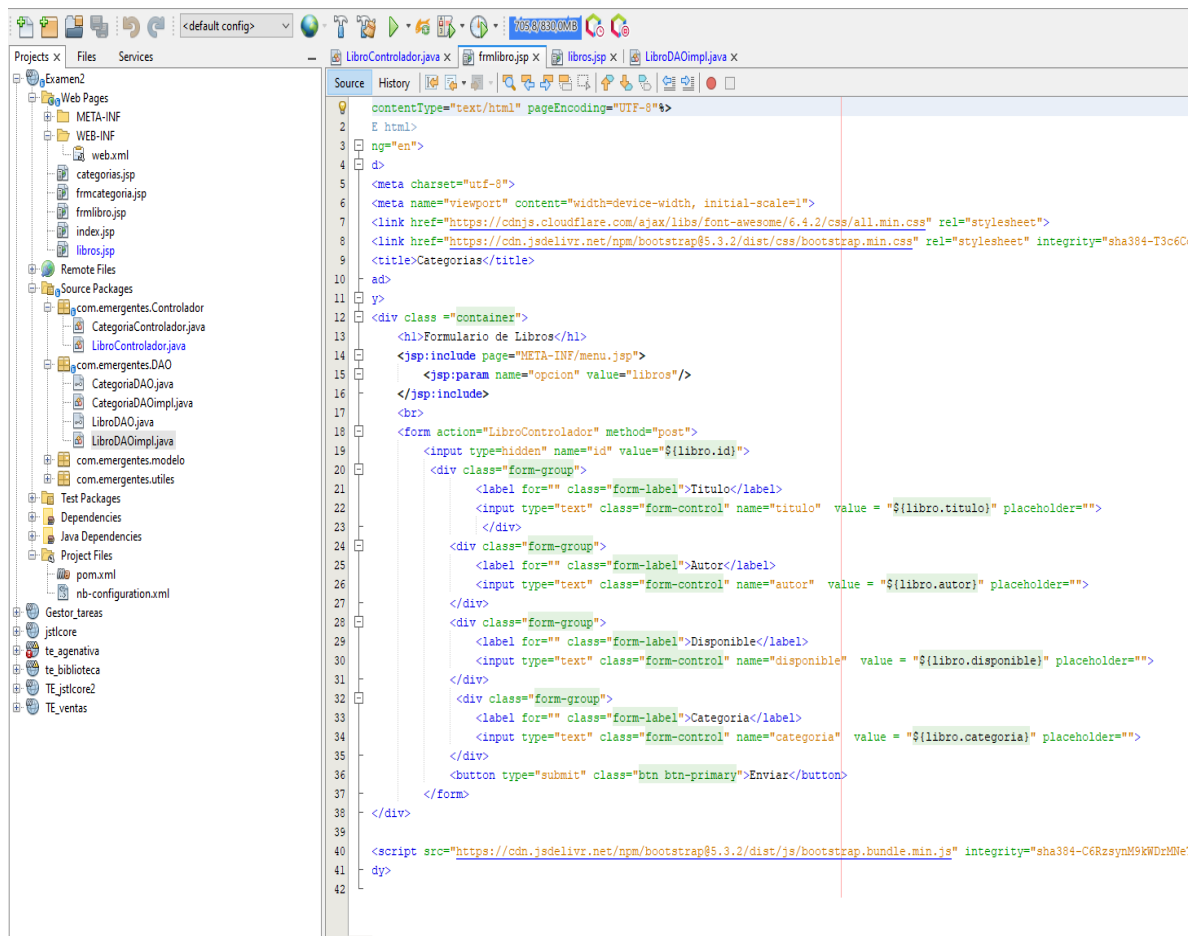
```
<? page="libros" title="Libros" %>
<!-- charset="utf-8" -->
<!-- name="viewport" content="width=device-width, initial-scale=1" -->
<!-- href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" rel="stylesheet" -->
<!-- href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3o6Co116uLrA9TneNEoa7RxnatzjcDSCmG1MXkSRlGAsXU" -->
</libros>
</title>
</head>
<body>
<div class="container">
<div>Libros</div>
<jsp:include page="META-INF/menu.jsp">
<jsp:param name="opcion" value="libros"/>
</jsp:include>
<a href="LibroControlador?action=add" class="btn btn-primary btn-sm"><i class="fa-solid fa-plus"></i> Nuevo </a>
<table class="table table-striped">
<tr>
<th>Id</th>
<th>Titulo</th>
<th>Autor</th>
<th>Disponible</th>
<th>Categoria</th>
</tr>
<c:forEach var="item" items="{libros}">
<tr>
<td>{item.id}</td>
<td>{item.titulo}</td>
<td>{item.autor}</td>
<td>{item.disponible}</td>
<td>{item.categoria}</td>
<td><a href="LibroControlador?action=edit&id={item.id}">Editar</a></td>
<td><a href="LibroControlador?action=delete&id={item.id}" onclick="esta seguro ?">Eliminar</a></td>
</tr>
</c:forEach>
</table>
</div>
</body>
</html>
<!-- src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9KWDrtMNeT87bh95OGNy2PhcTNXj1NW7RuBCsyN/o0jlpcV8Qy" -->
```

### CODIGO LIBROCONTROLADOR:

```
@WebServlet(name = "LibroControlador", urlPatterns = {"/LibroControlador"})
public class LibroControlador extends HttpServlet {

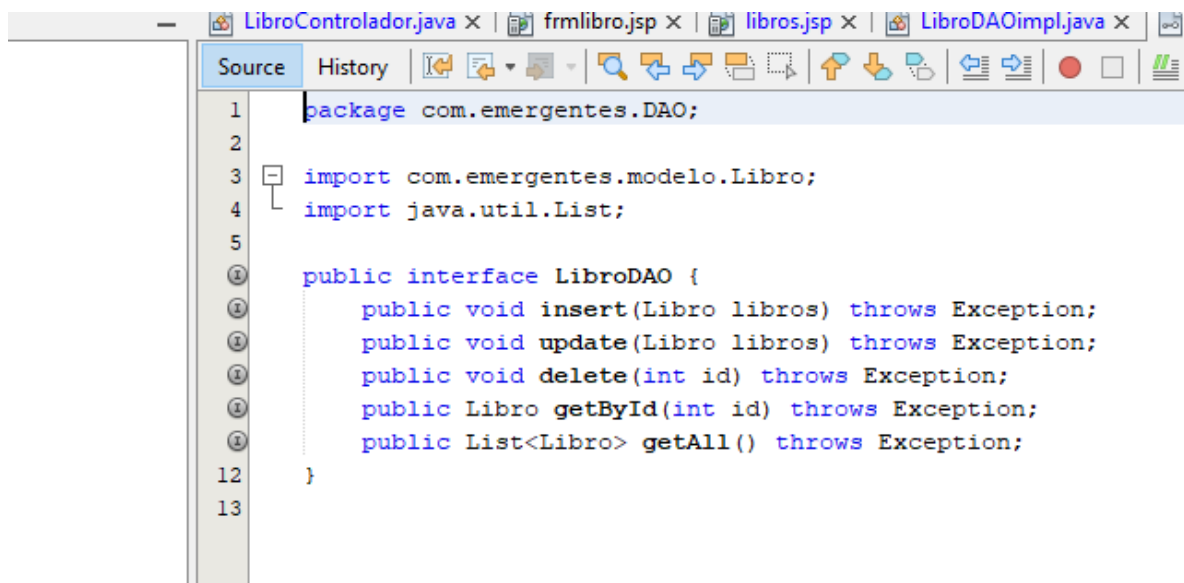
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            LibroDAO dao = new LibroDAOimpl();
            Libro lib = new Libro();
            int id;
            String action = (request.getParameter("action") != null) ? request.getParameter("action") : "view";
            switch (action) {
                case "add":
                    request.setAttribute("libro", lib);
                    request.getRequestDispatcher("frmLibro.jsp").forward(request, response);
                    break;
                case "edit":
                    id = Integer.parseInt(request.getParameter("id"));
                    lib = dao.getById(id);
                    request.setAttribute("libro", lib);
                    request.getRequestDispatcher("frmLibro.jsp").forward(request, response);
                    break;
                case "delete":
                    id = Integer.parseInt(request.getParameter("id"));
                    dao.getById(id);
                    response.sendRedirect("LibroControlador");
                    break;
                case "view":
                    List<Libro> lista = dao.getAll();
                    request.setAttribute("libros", lista);
                    request.getRequestDispatcher("libros.jsp").forward(request, response);
                    break;
            }
        } catch (Exception ex) {
            System.out.println("Error" + ex.getMessage());
        }
    }
}
```

## CODIGO FRMLIBRO:



```
1 contentType="text/html" pageEncoding="UTF-8">
2 <html>
3 <ng="en">
4 <div>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c60"
9 <title>Categorias</title>
10 <div>
11 <div class="container">
12 <h1>Formulario de Libros</h1>
13 <jsp:include page="META-INF/menu.jsp">
14 <jsp:param name="opcion" value="libros"/>
15 </jsp:include>
16 <br>
17 <form action="LibroControlador" method="post">
18 <input type="hidden" name="id" value="{libro.id}">
19 <div class="form-group">
20 <label for="" class="form-label">Titulo</label>
21 <input type="text" class="form-control" name="titulo" value="{libro.titulo}" placeholder="">
22 </div>
23 <div class="form-group">
24 <label for="" class="form-label">Autor</label>
25 <input type="text" class="form-control" name="autor" value="{libro.autor}" placeholder="">
26 </div>
27 <div class="form-group">
28 <label for="" class="form-label">Disponible</label>
29 <input type="text" class="form-control" name="disponible" value="{libro.disponible}" placeholder="">
30 </div>
31 <div class="form-group">
32 <label for="" class="form-label">Categoria</label>
33 <input type="text" class="form-control" name="categoria" value="{libro.categoria}" placeholder="">
34 </div>
35 <button type="submit" class="btn btn-primary">Enviar</button>
36 </form>
37 </div>
38 </div>
39 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9KWDtMNe"
40 </script>
41 </body>
42 </html>
```

## CODIGO LIBRODAO



```
1 package com.emergentes.DAO;
2
3 import com.emergentes.modelo.Libro;
4 import java.util.List;
5
6 public interface LibroDAO {
7     public void insert(Libro libros) throws Exception;
8     public void update(Libro libros) throws Exception;
9     public void delete(int id) throws Exception;
10    public Libro getById(int id) throws Exception;
11    public List<Libro> getAll() throws Exception;
12 }
13
```

## CODIGO LIBRODAOIMPL:

```
LibroControlador.java x | trmLibro.jsp x | libros.jsp x | LibroDAOImpl.java x | LibroDAO.java x
Source History
1 package com.emergentes.DAO;
2
3
4 import com.emergentes.modelo.Libro;
5 import com.emergentes.utiles.ConexionDB;
6 import java.sql.PreparedStatement;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class LibroDAOImpl extends ConexionDB implements LibroDAO {
13
14     @Override
15     public void insert(Libro libro) throws Exception {
16         // String sql = "insert into libros (titulo,autor,disponible,categoria) Values (?,?,?,?)";
17         this.conectar();
18         PreparedStatement ps = this.conn.prepareStatement("insert into libros (titulo,autor,disponible,categoria) Values (?,?,?,?)");
19         ps.setString(1, libro.getTitulo());
20         ps.setString(2, libro.getAutor());
21         ps.setString(3, libro.getDisponible());
22         ps.setString(4, libro.getCategoria());
23         ps.executeUpdate();
24         this.desconectar();
25     }
26
27     @Override
28     public void update(Libro libro) throws Exception {
29         String sql = " update libros set titulo=?, autor=?,disponible=?, categoria=? where id=?";
30         this.conectar();
31         PreparedStatement ps = this.conn.prepareStatement(sql);
32         ps.setString(1,libro.getTitulo());
33         ps.setString(2,libro.getAutor());
34         ps.setString(3,libro.getDisponible());
35         ps.setString(4,libro.getCategoria());
36         ps.setInt(5,libro.getId());
37         ps.executeUpdate();
38         this.desconectar();
39     }
40
41     @Override
42     public void delete(int id) throws Exception {
43         try{
44             this.conectar();
45             String sql = "delete from libros where id=?";
46             PreparedStatement ps = this.conn.prepareStatement(sql);
47             ps.setInt(1, id);
48             ps.executeUpdate();
49         }catch(Exception e){
50             throw e;
51         } finally {
52             this.desconectar();
53         }
54     }
55
56
57     @Override
58     public List<Libro> getAll() throws Exception {
59         List<Libro> lista = null;
60         String sql = " select * from libros";
61         this.conectar();
62         PreparedStatement ps = this.conn.prepareStatement(sql);
63         ResultSet rs = ps.executeQuery();
64
65         lista = new ArrayList<Libro>();
66         while(rs.next()){
67             Libro avi = new Libro();
68             avi.setId(rs.getInt("id"));
69             avi.setTitulo(rs.getString("titulo"));
70             avi.setAutor(rs.getString("autor"));
71             avi.setDisponible(rs.getString("disponible"));
72             avi.setCategoria(rs.getString("categoria"));
73             lista.add(avi);
74         }
75         this.desconectar();
76         return lista;
77     }
}
```

```
79      @Override
80      public Libro getById(int id) throws Exception {
81          Libro avi = new Libro();
82          try {
83              this.conectar();
84              String sql = "select * from libros where id=?";
85              PreparedStatement ps = this.conn.prepareStatement(sql);
86              ps.setInt(1, id);
87              ResultSet rs = ps.executeQuery();
88              while(rs.next()){
89                  avi.setId(rs.getInt("id"));
90                  avi.setTitulo(rs.getString("titulo"));
91                  avi.setAutor(rs.getString("autor"));
92                  avi.setDisponible(rs.getString("disponible"));
93                  avi.setCategoria(rs.getString("categoria"));
94              }
95          } catch (SQLException e) {
96              throw e;
97          } finally {
98              this.desconectar();
99          }
100      }
101
102      return avi;
103  }
104
105  }
```

CORRIDA:

Libros

Libros

Categorías

+ Nuevo

Id	Título	Autor	Disponible	Categoría		
1	fabulas	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	cien años de soledad	Dario	Si	Novela	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	quien se comio mi queso	robert	No	Fantasia	<a href="#">Editar</a>	<a href="#">Eliminar</a>
4	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
5	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
6	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>

## AÑADIENDO UN NUEVO ARCHIVO

### Formulario de Libros

[Libros](#)

[Categorías](#)

Título

Autor

Disponible

Categoría

[Enviar](#)

## Libros

[Libros](#)

[Categorías](#)

[+ Nuevo](#)

Id	Título	Autor	Disponible	Categoría		
1	pinocho	rosa	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	cien años de soledad	Dario	Si	Novela	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	quien se comio mi queso	robert	No	Fantasia	<a href="#">Editar</a>	<a href="#">Eliminar</a>
5	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
6	el lobo y la oveja	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
7	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
9	HARRY POTER	IORENA	Si	FANTASIA	<a href="#">Editar</a>	<a href="#">Eliminar</a>

## EDITANDO LA FABULA A LAS GALLINAS GORDAS Y FLACAS

### Formulario de Libros

[Libros](#)

[Categorías](#)

Título

Autor

Disponible

Categoría

[Enviar](#)

# Libros

Libros

Categorías

+ Nuevo

Id	Título	Autor	Disponible	Categoría		
1	pinocho	rosa	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	cien años de soledad	Dario	Si	Novela	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	quien se comio mi queso	robert	No	Fantasia	<a href="#">Editar</a>	<a href="#">Eliminar</a>
5	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
6	el lobo y la oveja	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
7	GALLINAS GORDAS Y FLACAS	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
9	HARRY POTER	IORENA	Si	FANTASIA	<a href="#">Editar</a>	<a href="#">Eliminar</a>

ELIMINAMOS EL LOBO Y LA OVEJA:

# Libros

Libros

Categorías

+ Nuevo

Id	Título	Autor	Disponible	Categoría		
1	pinocho	rosa	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	cien años de soledad	Dario	Si	Novela	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	quien se comio mi queso	robert	No	Fantasia	<a href="#">Editar</a>	<a href="#">Eliminar</a>
5	fabula	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
7	GALLINAS GORDAS Y FLACAS	Griego	Si	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
9	HARRY POTER	IORENA	Si	FANTASIA	<a href="#">Editar</a>	<a href="#">Eliminar</a>

LA TABLA CATEGORIA:

# Categorías

Libros

Categorías

+ Nuevo

Id	Categoría		
1	Novela	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	Cuento	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	Fantasia	<a href="#">Editar</a>	<a href="#">Eliminar</a>
4	chocolate	<a href="#">Editar</a>	<a href="#">Eliminar</a>