Cargo:	Docente Especialista		
Nombre:	Ing. Elvis Pachacama		
Asignatura:	Lenguajes de programación		
Carrera:	Desarrollo de Software	**Nivel:**	Tercer nivel
Estudiante:	Ismael Anrrango, Cristian Villacis		

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO
IMPLEMENTACION DEL CODIGO

CODIGO CATEGORIAS

```
<%--
  Created by IntelliJ IDEA.
  User: Estudiante
  Date: 13/12/2024
  Time: 8:07
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java"
import="java.util.*, models.*"%>
<%
    Categoria categoria = (Categoria) request.getAttribute("categoria");
%>

<html>
<head>
    <title>Formulario Categoria</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h1 class="text-center mb-4">Formulario de Categoria</h1>

    <div class="row justify-content-center">
        <div class="col-md-8">
            <form action="<%= request.getContextPath()
%>/categorias/form" method="post" class="bg-white p-4 border rounded
shadow-sm">
                <!-- Campo de ID de la categoría (oculto) -->
                <input type="hidden" name="idCategoria" value="<%=
categoria.getIdCategoria() %>">

                <!-- Campo Nombre de la Categoría -->
                <div class="mb-3">
                    <label for="nombre" class="form-label">Ingrese el
nombre de la categoria:</label>
                    <input type="text" id="nombre" name="nombre"
class="form-control" value="<%= categoria.getNombre() != null ?
categoria.getNombre() : "" %>">
                </div>

                <!-- Campo Estado (solo se muestra "Activo" ya que es
estático) -->
                <div class="mb-3">
                    <label for="estado" class="form-
```

```
label">Estado:</label>
                    <input type="hidden" name="estado" id="estado"
value="1">
                    <p class="form-control-plaintext">Activo</p>
                </div>

                <!-- Botón de envío -->
                <div class="mb-3">
                    <button type="submit" class="btn btn-primary w-
100">ENVIAR</button>
                </div>
            </form>
        </div>
    </div>
</div>
</body>
</html>
```

CODIGO SERVICES

```
package service;

import models.Categoria;

import java.util.List;
import java.util.Optional;

public interface CategoriaService {



    void guardar(Categoria categoria);
    List<Categoria> listarCategoria();
    Optional<Categoria> porIdCategoria(Long idCategoria);
}
```

IMPLEMENTACION

```
package service;

import models.Categoria;
import repositories.CategoriaRepositoryJdbcImplement;
import repositories.Repository;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;
import java.util.Optional;

public class CategoriaServiceJdbcImplement implements CategoriaService {

    private Repository<Categoria> repositoryCategoriaJdbc;

    public CategoriaServiceJdbcImplement(Connection connection) {
        this.repositoryCategoriaJdbc = new
CategoriaRepositoryJdbcImplement(connection);
    }

    @Override
    public void guardar(Categoria categoria) {
        try {
            repositoryCategoriaJdbc.guardar(categoria);
        } catch (SQLException throwables) {
            throw new
ServiceJdbcException(throwables.getMessage(),throwables.getCause());
        }
    }
```

```
    @Override
    public List<Categoria> listarCategoria() {
        try {
            return repositoryCategoriaJdbc.listar();
        } catch (SQLException throwables) {
            throw new
ServiceJdbcException(throwables.getMessage(),throwables.getCause());
        }
    }



    @Override
    public Optional<Categoria> porIdCategoria(Long idCategoria) {
        try {
            return
Optional.ofNullable(repositoryCategoriaJdbc.porId(idCategoria));
        } catch (SQLException throwables) {
            throw new ServiceJdbcException(throwables.getMessage(),
throwables.getCause());
        }
    }



}
```

IMPLEMENTACION REPOSITORIO

```
package repositories;

import models.Categoria;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class CategoriaRepositoryJdbcImplement implements
Repository<Categoria> {
    private Connection conn;

    public CategoriaRepositoryJdbcImplement(Connection conn) {
        this.conn = conn;
    }

    @Override
    public List<Categoria> listar() throws SQLException {
        List<Categoria> categorias = new ArrayList<>();
        try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery("SELECT * FROM categoria")) {
            while (rs.next()) {
                Categoria c = getCategoria(rs);
                categorias.add(c);
            }
        }
        return categorias;
    }

    @Override
    public Categoria porId(Long idCategoria) throws SQLException {
        Categoria categoria = null;
        try (PreparedStatement stmt = conn.prepareStatement("SELECT *
FROM categoria WHERE idcategoria = ?")) {
            stmt.setLong(1, idCategoria);
            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
```

```
                    categoria = getCategoria(rs);
                }
            }
        }
        return categoria;
    }

    @Override
    public void guardar(Categoria categoria) throws SQLException {
        String sql;
        if (categoria.getIdCategoria() != null &&
categoria.getIdCategoria() > 0) {
            // Si tiene ID, es una actualización
            sql = "UPDATE categoria SET nombre = ?, estado = ? WHERE
idcategoria = ?";
        } else {
            // Si no tiene ID, es una inserción
            sql = "INSERT INTO categoria (nombre, estado) VALUES (?, ?)";
        }

        try (PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(1, categoria.getNombre());
            stmt.setInt(2, categoria.getEstado());

            if (categoria.getIdCategoria() != null &&
categoria.getIdCategoria() > 0) {
                stmt.setLong(3, categoria.getIdCategoria()); // Para la
actualización
            }
            stmt.executeUpdate();
        }
    }

    @Override
    public void eliminar(Long id) throws SQLException {
        // Implementa la lógica para eliminar si lo necesitas
    }

    @Override
    public void invalidar(Categoria categoria) throws SQLException {
        // Primer paso: obtener el estado actual de la categoría
        String selectSql = "SELECT estado FROM categoria WHERE
idcategoria = ?";
        try (PreparedStatement selectStmt =
conn.prepareStatement(selectSql)) {
            selectStmt.setLong(1, categoria.getIdCategoria());
            ResultSet rs = selectStmt.executeQuery();

            // Si existe la categoría
            if (rs.next()) {
                int estadoActual = rs.getInt("estado");
                int nuevoEstado = (estadoActual == 0) ? 1 : 0; // Cambiar
de 0 a 1 o de 1 a 0

                // Segundo paso: actualizar el estado con el valor
opuesto
                String updateSql = "UPDATE categoria SET estado = ? WHERE
idcategoria = ?";
                try (PreparedStatement updateStmt =
conn.prepareStatement(updateSql)) {
                    updateStmt.setInt(1, nuevoEstado);
                    updateStmt.setLong(2, categoria.getIdCategoria());
                    updateStmt.executeUpdate();
                }
            }
        }
    }
```

```
    @Override
    public void actualizar(Categoria categoria) throws SQLException {

    }


    private static Categoria getCategoria(ResultSet rs) throws
SQLException {
        Categoria c = new Categoria();
        c.setIdCategoria(rs.getLong("idcategoria"));
        c.setNombre(rs.getString("nombre"));
        c.setEstado(rs.getInt("estado"));
        return c;
    }

}
```

CATEGORIA FROM CONTROLADOR

```
package controllers;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import models.Categoria;
import models.Productos;
import service.CategoriaService;
import service.CategoriaServiceJdbcImplement;
import service.ProductoService;
import service.ProductoServiceJdbcImplement;

import java.io.IOException;
import java.sql.Connection;
import java.util.Optional;

@WebServlet("/categorias/form")
public class CategoriaFromControlador extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        // Recuperamos la conexión
        Connection conn = (Connection) req.getAttribute("conn");
        CategoriaService services = new
CategoriaServiceJdbcImplement(conn);

        // Recuperamos el ID de la categoría para editar (si existe)
        long id = 0L;
        try {
            // Recogemos el parámetro 'idCategoria' de la URL
            id = Long.parseLong(req.getParameter("idCategoria"));
        } catch (NumberFormatException e) {
            // Si no existe o no es válido, lo dejamos en 0L
        }

        // Si existe un id válido, obtenemos la categoría para editar
        Categoria categoria = new Categoria();
        if (id > 0) {
            Optional<Categoria> o = services.porIdCategoria(id);
            if (o.isPresent()) {
                categoria = o.get();
            }
        }
```

```
        // Si no se encuentra la categoría, la dejamos en blanco para
crear
        req.setAttribute("categoria", categoria);

getServletContext().getRequestDispatcher("/formularioCategoria.jsp").forw
ard(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        // Obtenemos la conexión y el servicio
        Connection conn = (Connection) req.getAttribute("conn");
        CategoriaService service = new
CategoriaServiceJdbcImplement(conn);

        // Recuperamos los parámetros del formulario
        String nombre = req.getParameter("nombre");
        int estado = 1;  // Por defecto "Activo"

        // Recuperamos el ID de la categoría (si es edición)
        Long idCategoria = 0L;
        try {
            idCategoria = Long.valueOf(req.getParameter("idCategoria"));
        } catch (NumberFormatException e) {
            idCategoria = 0L;
        }


        // Creamos o actualizamos la categoría
        Categoria categoria = new Categoria();
        categoria.setIdCategoria(idCategoria);
        categoria.setNombre(nombre);
        categoria.setEstado(estado);

        // Guardamos la categoría (insertar o actualizar según el caso)
        service.guardar(categoria);

        // Redireccionamos para evitar reenvíos del formulario
        resp.sendRedirect(req.getContextPath() + "/categorias");
    }
}
```

VISTAS

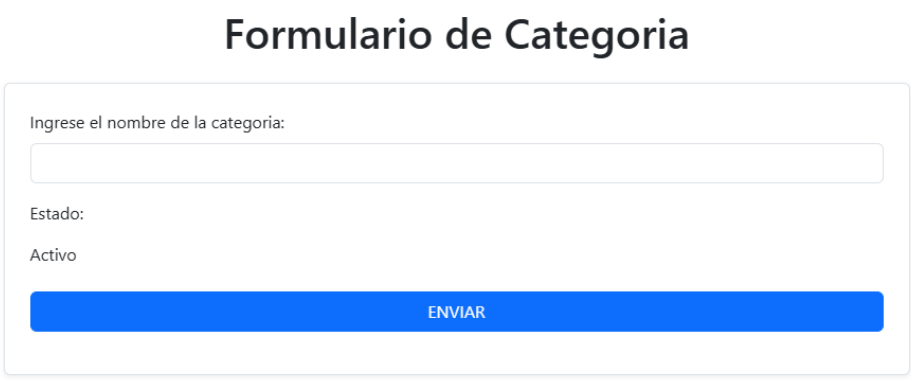
```
<%--
  Created by IntelliJ IDEA.
  User: Estudiante
  Date: 12/12/2024
  Time: 15:01
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java"
import="java.util.*,models.*" %>
<%
  List<Categoria> categoria = (List<Categoria>)
request.getAttribute("categoria");
  Optional<String> username = (Optional<String>)
request.getAttribute("username");
%>
<html>
<head>
  <title>Lista de Categorías</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
```

```
<div class="container mt-4">
  <h1 class="text-center mb-4">Listado de Categorías</h1>

  <% if (username.isPresent()) { %>
  <div class="alert alert-info">
    Hola <%= username.get() %>, bienvenido a la aplicación!
  </div>
  <div class="mb-4">
    <a href="${pageContext.request.contextPath}/categorias/form"
class="btn btn-primary">Ingresar Categoría</a>
  </div>
  <% } %>

  <table class="table table-bordered table-striped">
    <thead class="thead-dark">
    <tr>
      <th>ID CATEGORÍA</th>
      <th>NOMBRE</th>
      <% if (username.isPresent()) { %>
      <th>ESTADO</th>
      <th>OPCIONES</th>
      <% } %>
    </tr>
    </thead>
    <tbody>
    <% for (Categoria c : categoria) { %>
    <tr>
      <td><%= c.getIdCategoria() %></td>
      <td><%= c.getNombre() %></td>
      <% if (username.isPresent()) { %>
      <td><%= c.getEstado() == 1 ? "Activo" : "Inactivo" %></td>
      <td>
        <a href="<%= request.getContextPath()
%>/categorias/form?idCategoria=<%= c.getIdCategoria() %>" class="btn btn-
warning btn-sm">Editar</a>

        <a href="<%= request.getContextPath()
%>/categorias/inavilitar?idCategoria=<%= c.getIdCategoria() %>"
class="btn btn-danger btn-sm">Inavilitar/Avilitar</a>
      </td>
      <% } %>
    </tr>
    <% } %>
    </tbody>
  </table>
</div>
</body>
</html>
```

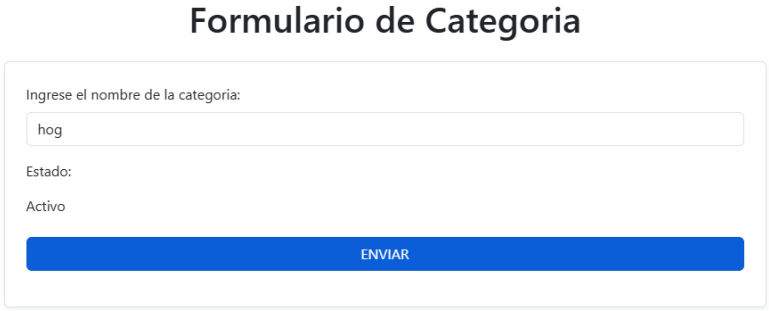
CAPTURAS DE LAS VISTAS

Listado de Categorías

INGRESAR



EDITAR



STOCK PRODUCTOS

LINK REPOSITORIO: https://github.com/Ismakolis/CarroUltimaVersion.git