

Facultad de Ciencias Físico Matemáticas

Laboratorio Programación Orientada Objetos

Semestre Agosto 2025 - Diciembre 2025

Mtro. Jorge Alberto Islas Pineda

PIA

Nombre: Melina Sanchez Ávila

Matricula: 2120543

Grupo: 032

Horario: sábado 7:00 am – 9:00 am

Fecha: 22 Noviembre 2025

Contenido

Introducción	3
Descripción del Proyecto.....	4
Modelo – Vista – Controlador	5
Modelo de Datos.....	6
Script MySQL (Sencillo)	7
Código JSP (Web Pages)	8
Index.jsp.....	8
mostrarUsuarios.jsp	13
guardarUsuarios.jsp	18
modificarUsuarios.jsp.....	26
eliminarUsuarios.jsp.....	36
Código Java (Source Packages - Agenda)	45
ConexionDAO.java.....	45
EntidadUsuario.java	47
UsuarioDAO.java	50
Servletusuario.java.....	55
Capturas de Pantallas de la Pagina Web.....	65
Menú con conexión	65
Menú sin conexión	66
Guardar de Contactos	68
Modificar de Contactos.....	70
Eliminar de Contactos.....	72
Validaciones visuales	74
Validaciones visuales extra de modificar y eliminar	75
Repositorio GitHub	76
Conclusión	77

Introducción

El proyecto Agenda CRUD con Java Servlets tiene como finalidad desarrollar una aplicación web funcional que permita administrar una lista de contactos utilizando el patrón de arquitectura Modelo–Vista–Controlador (MVC). Este sistema fue construido empleando Java, JSP, Servlets, MySQL y desplegado en el servidor Apache Tomcat, integrando así tecnologías esenciales del desarrollo web basado en Java.

La aplicación permite realizar las cuatro operaciones fundamentales de persistencia de datos (Crear, Leer, Actualizar y Eliminar) sobre una entidad denominada Usuario. Cada usuario contiene información personal como nombre, apellidos, sexo, teléfono, dirección y tipo de contacto. Los datos se almacenan en una base de datos MySQL y son gestionados mediante un componente de acceso a datos (DAO) que se comunica con la clase de conexión (ConexionDAO).

El patrón MVC se aplicó de manera estructurada:

El Modelo está compuesto por la clase EntidadUsuario que representa los datos del contacto.

El Controlador se implementó mediante el servlet `servletusuario.java`, responsable de recibir las solicitudes del cliente, validar datos y comunicarse con el DAO.

La Vista está compuesta por diferentes páginas JSP (`mostrarUsuarios.jsp`, `guardarUsuarios.jsp`, `modificarUsuarios.jsp`, etc.), encargadas de presentar la información al usuario final.

Este proyecto permite comprender el flujo completo de una aplicación web en Java, desde la captura de datos en un formulario JSP, pasando por el procesamiento mediante un Servlet, la interacción con la base de datos, hasta la visualización final de los resultados. Además, refuerza el uso correcto de servlets, manejo de dependencias, estructuración de paquetes, JSPs y conexión a MySQL, habilidades fundamentales en el desarrollo de aplicaciones web profesionales.

Descripción del Proyecto

El proyecto consiste en el desarrollo de una aplicación web de Agenda de Usuarios, cuyo objetivo es permitir la gestión completa de contactos mediante operaciones CRUD (Crear, Leer, Actualizar y Eliminar). La aplicación fue construida utilizando Java, Servlets, JSP, MySQL y el servidor Apache Tomcat, siguiendo la arquitectura Modelo–Vista–Controlador (MVC) para mantener una estructura ordenada, escalable y fácil de mantener.

La aplicación cuenta con una interfaz web desarrollada con páginas JSP, desde las cuales el usuario puede registrar nuevos contactos, visualizar la lista de contactos existentes, modificar información o eliminar registros. Cada contacto contiene los siguientes atributos: nombre, apellido paterno, apellido materno, sexo, teléfono, dirección y tipo de contacto (por ejemplo, “CASA” o “TRABAJO”).

Para la gestión de los datos, el sistema utiliza una base de datos MySQL, a la cual se accede mediante una clase de conexión (ConexionDAO) y un objeto DAO (UsuarioDAO). Este DAO encapsula las consultas SQL necesarias para realizar las operaciones CRUD, asegurando una comunicación ordenada y segura entre la aplicación y la base de datos.

El controlador principal, implementado en `servletusuario.java`, es el encargado de recibir las solicitudes del usuario desde las vistas (JSP), procesar la información recibida, validar los datos y coordinar las operaciones necesarias en el DAO. Dependiendo de la acción solicitada—como guardar, modificar o eliminar—el servlet redirige a la vista correspondiente y actualiza la información presentada al usuario.

La vista está compuesta por diferentes páginas JSP como `guardarUsuarios.jsp`, `modificarUsuarios.jsp`, `mostrarUsuarios.jsp` y `eliminarUsuarios.jsp`, que permiten navegar de manera sencilla por las distintas funcionalidades del sistema. Estas vistas presentan los formularios, las tablas con los registros y los mensajes de operación realizada.

En conjunto, el proyecto proporciona una aplicación funcional que demuestra el flujo completo de una aplicación web dinámica en Java, desde la captura de datos, su procesamiento en el servidor, la conexión con la base de datos y la presentación final en la interfaz del usuario.

Modelo – Vista – Controlador

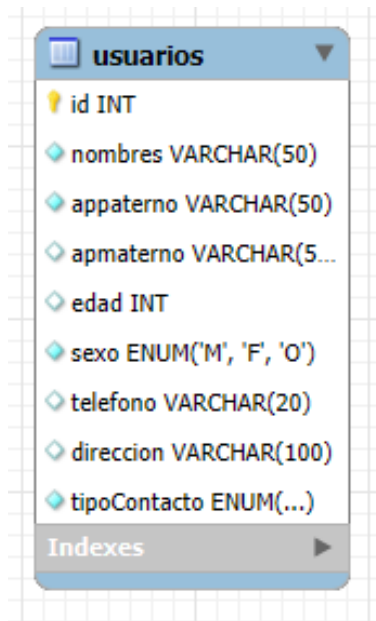
En este proyecto la arquitectura se basa en el patrón **MVC (Modelo–Vista–Controlador)**, aplicado utilizando únicamente **JSP para la parte visual** y **Servlets para toda la lógica del servidor**, sin usar páginas HTML independientes.

El **Modelo** está compuesto por la clase `EntidadUsuario.java`, que representa la estructura de los contactos dentro de la agenda. Esta clase contiene los atributos principales del usuario (id, nombre, apellidos, sexo, teléfono, dirección y tipo de contacto), junto con sus métodos getters y setters. Su función es encapsular los datos y servir como objeto que se intercambia entre el controlador y el DAO. El modelo no incluye lógica de negocio ni conexión a la base de datos; únicamente define los datos que la aplicación manipula.

El **Controlador** está implementado en el servlet `servletusuario.java`. Este servlet es el encargado de recibir todas las solicitudes enviadas desde las vistas JSP, como guardar, modificar, eliminar o mostrar usuarios. Dependiendo de la acción requerida, el controlador valida los datos que el usuario ingresa, crea y llena un objeto de tipo `EntidadUsuario`, y posteriormente se comunica con la capa de acceso a datos (`UsuarioDAO`) para ejecutar las operaciones CRUD en la base de datos MySQL. Además, el servlet se encarga de redirigir a las vistas correctas después de cada operación y de enviar la información necesaria para que se muestre al usuario. Todo el procesamiento, validación y manejo de datos se realiza exclusivamente en el servlet.

Finalmente, la **Vista** está formada por las distintas páginas JSP utilizadas en el proyecto: `index.jsp`, `guardarUsuarios.jsp`, `modificarUsuarios.jsp`, `eliminarUsuarios.jsp` y `mostrarUsuarios.jsp`. Estas páginas funcionan como la interfaz gráfica de la aplicación y se encargan únicamente de mostrar formularios, tablas o mensajes al usuario. Como no se utilizó HTML separado, todas las pantallas fueron construidas directamente en JSP, lo que permite integrar elementos dinámicos mediante Java embebido o expresiones JSP. Las vistas no contienen lógica de negocio ni validaciones, ya que su única función es mostrar información y enviar los datos al servlet, manteniendo así una separación clara de responsabilidades dentro del patrón MVC.

Modelo de Datos



El modelo de datos del proyecto está compuesto por una única tabla llamada **usuarios**, la cual almacena la información correspondiente a cada contacto registrado en la agenda. Esta tabla se encuentra dentro de la base de datos **dbusur** y está diseñada para representar de forma simple y eficiente los datos personales de cada usuario del sistema.

La tabla **usuarios** contiene los siguientes campos:

- **id (INT, PRIMARY KEY, AUTO_INCREMENT):**
Es el identificador único de cada registro. Se genera automáticamente y permite diferenciar a cada usuario dentro del sistema. Sirve como clave primaria y se utiliza en las operaciones de edición, eliminación y consulta.
- **nombres (VARCHAR(50)):**
Almacena el nombre o nombres del usuario. Es un campo de texto que permite registrar hasta 50 caracteres y corresponde a la parte principal del nombre de la persona.
- **appaterno (VARCHAR(50)):**
Guarda el apellido paterno del usuario. También acepta hasta 50 caracteres y se utiliza como parte fundamental para identificar al contacto.
- **apmaterno (VARCHAR(50)):**
Contiene el apellido materno del usuario. Su estructura es similar al campo anterior y complementa la información personal del contacto.
- **edad (INT):**
Representa la edad del usuario como un valor numérico entero. Este dato puede utilizarse para clasificar, validar o mostrar información relevante dentro de la aplicación.

La estructura de la tabla está pensada para satisfacer las operaciones CRUD requeridas por el sistema, permitiendo insertar, actualizar, eliminar y mostrar información de manera eficiente. La clave primaria **id** facilita la identificación de cada registro y garantiza la integridad de las operaciones.

Script MySQL (Sencillo)

```
create database dbusr;
```

```
use dbusr;
```

```
create table usuarios(
```

```
    id int auto_increment not null primary key,
```

```
    nombres varchar(50),
```

```
    appaterno varchar(50),
```

```
    apmaterno varchar(50),
```

```
    edad int
```

```
);
```

Código JSP (Web Pages)

Index.jsp

```
<%--
```

```
    Document : index
```

```
    Created on : 17 nov 2025, 9:56:11 p.m.
```

```
    Author : melin
```

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@page import="Agenda.ConexionDAO"%>
```

```
<%
```

```
    boolean dbOnline = ConexionDAO.probarConexion();
```

```
%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>CRUD Agenda</title>
```

```
    <style>
```

```
        body {
```

```
            font-family: Arial, Helvetica, sans-serif;
```

```
            background-color: #f3f4f6;
```

```
            margin: 0;
```

```
            padding: 0;
```

```
            color: #333;
```

```
        }
```

```
        h1 {
```

```
            text-align: center;
```



```
margin-top: 40px;  
}
```

```
h2 {  
  text-align: center;  
  margin-bottom: 30px;  
  color: #555;  
}
```

```
.container {  
  width: 100%;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu {  
  background: #ffffff;  
  padding: 30px 40px;  
  border-radius: 10px;  
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);  
  text-align: center;  
}
```

```
.menu a {  
  text-decoration: none;  
}
```

```
.btn {  
  display: block;  
  width: 220px;  
  margin: 12px auto;  
  padding: 12px 20px;  
  background-color: #2563eb;
```

```
color: white;
border: none;
border-radius: 6px;
cursor: pointer;
font-size: 15px;
transition: 0.25s;
}
```

```
.btn:hover {
  background-color: #1e4fbe;
}
```

```
.status {
  position: fixed;
  top: 10px;
  left: 10px;
  padding: 8px 12px;
  background: #ffffff;
  border-radius: 6px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.2);
  font-size: 14px;
  display: flex;
  align-items: center;
  gap: 8px;
}
```

```
.status span {
  width: 12px;
  height: 12px;
  border-radius: 50%;
  display: inline-block;
}
```

```
.online {
```

```
        background-color: #22c55e; /* verde */
    }

    .offline {
        background-color: #ef4444; /* rojo */
    }

</style>
</head>

<body>
    <div class="status">
        <span class="<%= dbOnline ? "online" : "offline" %>"></span>
        <%= dbOnline ? "Conectado a la BD" : "Sin conexión" %>
    </div>

    <h1>CRUD Agenda</h1>
    <h2>Menú Principal</h2>

    <div class="container">
        <div class="menu">

            <a href="servletusuario?action=mostrar">
                <button class="btn">Mostrar Contactos</button>
            </a>

            <a href="servletusuario?action=guardar">
                <button class="btn">Guardar Contactos</button>
            </a>

            <a href="servletusuario?action=modificar">
                <button class="btn">Modificar Contactos</button>
            </a>


```

```
<a href="servletusuario?action=eliminar">  
  <button class="btn">Eliminar Contactos</button>  
</a>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

mostrarUsuarios.jsp

<%--

Document : mostrarUsuarios

Created on : 17 nov 2025, 10:07:59 p.m.

Author : melin

--%>

<%@page import="java.util.List"%>

<%@page import="Agenda.EntidadUsuario"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Tabla de Contactos</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

background-color: #f3f4f6;

margin: 0;

padding: 0;

color: #333;

}

h1 {

text-align: center;

margin-top: 40px;

color: #222;

}

.contenedor {

```
width: 90%;
max-width: 900px;
margin: 30px auto;
background: #ffffff;
padding: 25px 35px;
border-radius: 10px;
box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}
```

```
.btn-regresar {
display: inline-block;
margin-bottom: 20px;
padding: 10px 18px;
background-color: #2563eb;
color: white;
text-decoration: none;
border-radius: 6px;
font-size: 14px;
transition: 0.25s;
}
```

```
.btn-regresar:hover {
background-color: #1e4f8e;
}
```

```
table {
width: 100%;
border-collapse: collapse;
margin-top: 15px;
}
```

```
th {
background-color: #2563eb;
```

```

        color: white;
        padding: 12px;
        font-size: 14px;
    }

    td {
        padding: 10px;
        background: #fafafa;
        border-bottom: 1px solid #ddd;
        text-align: center;
    }

    tr:hover td {
        background-color: #f0f4ff;
    }

    .mensaje-vacio {
        text-align: center;
        padding: 20px;
        color: #555;
    }
</style>
</head>

<body>

    <h1>Tabla de Contactos</h1>

    <div class="contenedor">

        <a class="btn-regresar" href="index.jsp">Regresar al Menú</a>

        <%

```

```

        List<EntidadUsuario> usuarios = (List<EntidadUsuario>)
request.getAttribute("listausuarios");

        if (usuarios == null || usuarios.isEmpty()) {
%>

<div class="mensaje-vacio">No hay usuarios registrados.</div>

<% } else { %>

<table>
    <tr>
        <th>ID</th>
        <th>Nombres</th>
        <th>Apellido Paterno</th>
        <th>Apellido Materno</th>
        <th>Edad</th>
        <th>Sexo</th>
        <th>Telefono</th>
        <th>Direccion</th>
        <th>Tipo de Contacto</th>
    </tr>

    <% for (EntidadUsuario usuario : usuarios) { %>
    <tr>
        <td><%= usuario.getId() %></td>
        <td><%= usuario.getNombres() %></td>
        <td><%= usuario.getAppaterno() %></td>
        <td><%= usuario.getApmaterno() %></td>
        <td><%= usuario.getEdad() %></td>
        <td><%= usuario.getSexo() %></td>
        <td><%= usuario.getTelefono() %></td>
        <td><%= usuario.getDireccion() %></td>
        <td><%= usuario.getTipoContacto() %></td>
    </tr>

```


<% } %>

</table>

<% } %>

</div>

</body>

</html>

guardarUsuarios.jsp

<%--

Document : guardarUsuarios

Created on : 17 nov 2025, 10:11:27 p.m.

Author : melin

--%>

<%@page import="java.util.List"%>

<%@page import="Agenda.EntidadUsuario"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Guardar Contactos</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

background-color: #f3f4f6;

margin: 0;

padding: 0;

color: #333;

}

h1 {

text-align: center;

margin-top: 40px;

color: #222;

}

.contenedor {

```
width: 90%;  
max-width: 900px;  
margin: 30px auto;  
background: #ffffff;  
padding: 25px 35px;  
border-radius: 10px;  
box-shadow: 0 4px 10px rgba(0,0,0,0.1);  
}
```

```
.btn-regresar {  
  display: inline-block;  
  margin-bottom: 20px;  
  padding: 10px 18px;  
  background-color: #2563eb;  
  color: white;  
  text-decoration: none;  
  border-radius: 6px;  
  font-size: 14px;  
  transition: 0.25s;  
}
```

```
.btn-regresar:hover {  
  background-color: #1e4f8e;  
}
```

```
form {  
  margin-bottom: 30px;  
}
```

```
label {  
  display: block;  
  margin: 10px 0 6px;  
  font-weight: bold;
```

```
}
```

```
input[type="text"] {  
  width: 60%;  
  padding: 10px;  
  border: 1px solid #ccc;  
  border-radius: 6px;  
  font-size: 14px;  
  transition: 0.2s;  
}
```

```
input[type="text"]:focus {  
  border-color: #2563eb;  
  outline: none;  
  background: #f0f4ff;  
}
```

```
.btn-guardar {  
  margin-top: 20px;  
  padding: 12px 25px;  
  background-color: #2563eb;  
  border: none;  
  color: white;  
  border-radius: 6px;  
  cursor: pointer;  
  font-size: 15px;  
  transition: 0.2s;  
}
```

```
.btn-guardar:hover {  
  background-color: #1e4fbe;  
}
```

```
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 15px;
}

th {
    background-color: #2563eb;
    color: white;
    padding: 12px;
    font-size: 14px;
}

td {
    padding: 10px;
    background: #fafafa;
    border-bottom: 1px solid #ddd;
    text-align: center;
}

tr:hover td {
    background-color: #f0f4ff;
}

.mensaje-vacio {
    text-align: center;
    padding: 15px;
    color: #555;
}

select {
    width: 60%;
    padding: 10px;
    border: 1px solid #ccc;
```

```

        border-radius: 6px;
        font-size: 14px;
        background: white;
        transition: 0.2s;
    }

    select:focus {
        border-color: #2563eb;
        background: #f0f4ff;
        outline: none;
    }
}

</style>
</head>

<body>

<h1>Guardar Contactos</h1>

<div class="contenedor">

    <a class="btn-regresar" href="index.jsp">Regresar al Menú</a>

    <%
        String error = (String) request.getAttribute("error");
        if (error != null) {
    %>
        <div style="background:#ffdddd; color:#a33; padding:12px;
            border:1px solid #cc0000; border-radius:6px;
            margin-bottom:15px; font-weight:bold;">
            <%= error %>
        </div>
    %>
    %>
    }

```

%>

```
<form action="servletusuario" method="POST">
```

```
  <label>Nombres:</label>
```

```
  <input type="text" name="nombres">
```

```
  <label>Apellido Paterno:</label>
```

```
  <input type="text" name="appaterno">
```

```
  <label>Apellido Materno:</label>
```

```
  <input type="text" name="apmaterno">
```

```
  <label>Edad:</label>
```

```
  <input type="text" name="edad">
```

```
  <label>Sexo:</label>
```

```
  <select name="sexo">
```

```
    <option value="M">Masculino</option>
```

```
    <option value="F">Femenino</option>
```

```
    <option value="O">Otro</option>
```

```
  </select>
```

```
  <label>Telefono:</label>
```

```
  <input type="text" name="telefono">
```

```
  <label>Direccion:</label>
```

```
  <input type="text" name="direccion">
```

```
  <label>Tipo de Contacto:</label>
```

```
  <select name="tipoContacto">
```

```
    <option value="CASA">Casa</option>
```

```
    <option value="TRABAJO">Trabajo</option>
```

```
  </select>
```

```
<button type="submit" class="btn-guardar" name="action"
value="guardar">Guardar</button>
```

```
</form>
```

```
<h3>Lista de Contactos</h3>
```

```
<%
```

```
List<EntidadUsuario> usuarios = (List<EntidadUsuario>)
request.getAttribute("listausuarios");
```

```
if (usuarios == null || usuarios.isEmpty()) {
```

```
%>
```

```
<div class="mensaje-vacio">No hay usuarios registrados aún.</div>
```

```
<% } else { %>
```

```
<table>
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>Nombres</th>
```

```
<th>Apellido Paterno</th>
```

```
<th>Apellido Materno</th>
```

```
<th>Edad</th>
```

```
<th>Sexo</th>
```

```
<th>Telefono</th>
```

```
<th>Direccion</th>
```

```
<th>Tipo de Contacto</th>
```

```
</tr>
```

```
<% for (EntidadUsuario usuario : usuarios) { %>
```

```
<tr>
```

```
<td><%= usuario.getId() %></td>
```

```
<td><%= usuario.getNombres() %></td>
```



```
<td><%= usuario.getAppaterno() %></td>
<td><%= usuario.getApmaterno() %></td>
<td><%= usuario.getEdad() %></td>
<td><%= usuario.getSexo() %></td>
<td><%= usuario.getTelefono() %></td>
<td><%= usuario.getDireccion() %></td>
<td><%= usuario.getTipoContacto() %></td>
</tr>
<% } %>
</table>

<% } %>

</div>

</body>
</html>
```

modificarUsuarios.jsp

<%--

Document : modificarUsuarios

Created on : 17 nov 2025, 10:17:15 p.m.

Author : melin

--%>

<%@page import="java.util.List"%>

<%@page import="Agenda.EntidadUsuario"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Modificar Contactos</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

background-color: #f3f4f6;

margin: 0;

padding: 0;

color: #333;

}

h1 {

text-align: center;

margin-top: 40px;

color: #222;

```
}
```

```
.contenedor {
```

```
width: 90%;
```

```
max-width: 950px;
```

```
margin: 30px auto;
```

```
background: #ffffff;
```

```
padding: 25px 35px;
```

```
border-radius: 10px;
```

```
box-shadow: 0 4px 12px rgba(0,0,0,0.1);
```

```
}
```

```
.btn-regresar {
```

```
display: inline-block;
```

```
margin-bottom: 20px;
```

```
padding: 10px 18px;
```

```
background-color: #2563eb;
```

```
color: white;
```

```
text-decoration: none;
```

```
border-radius: 6px;
```

```
font-size: 14px;
```

```
transition: 0.25s;
```

```
}
```

```
.btn-regresar:hover {
```

```
background-color: #1e4f8e;
```

```
}
```

```
label {
```

```
display: block;
```

```
margin-top: 12px;  
font-weight: bold;  
}
```

```
input[type="text"] {  
  width: 60%;  
  padding: 10px;  
  border: 1px solid #ccc;  
  border-radius: 6px;  
  margin-top: 5px;  
  font-size: 14px;  
  transition: 0.2s;  
}
```

```
input[type="text"]:focus {  
  border-color: #2563eb;  
  outline: none;  
  background: #eef3ff;  
}
```

```
.btn-guardar {  
  margin-top: 20px;  
  padding: 12px 25px;  
  background-color: #2563eb;  
  border: none;  
  color: white;  
  border-radius: 6px;  
  cursor: pointer;  
  font-size: 15px;  
  transition: 0.2s;
```

```
}
```

```
.btn-guardar:hover {  
    background-color: #1e4fbe;  
}
```

```
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin-top: 25px;  
}
```

```
th {  
    background-color: #2563eb;  
    color: white;  
    padding: 12px;  
    font-size: 14px;  
}
```

```
td {  
    padding: 10px;  
    background: #fafafa;  
    border-bottom: 1px solid #ddd;  
    text-align: center;  
}
```

```
tr:hover td {  
    background-color: #f0f4ff;  
    cursor: pointer;  
}
```

```
.mensaje-vacio {
    text-align: center;
    padding: 15px;
    color: #555;
}

select {
    width: 60%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 6px;
    font-size: 14px;
    background: white;
    transition: 0.2s;
}

select:focus {
    border-color: #2563eb;
    background: #f0f4ff;
    outline: none;
}

</style>
</head>

<body>

<h1>Modificar Contactos</h1>

<div class="contenedor">
```

```
<a class="btn-regresar" href="index.jsp">Regresar al Menú</a>
```

```
<%
```

```
String error = (String) request.getAttribute("error");
```

```
if (error != null) {
```

```
%>
```

```
<div style="background:#ffdddd; color:#a33; padding:12px;
```

```
border:1px solid #cc0000; border-radius:6px;
```

```
margin-bottom:15px; font-weight:bold;">
```

```
<%= error %>
```

```
</div>
```

```
<% } %>
```

```
<form action="servletusuario" method="POST">
```

```
<label>ID:</label>
```

```
<input type="text" name="id" id="idid" readonly>
```

```
<label>Nombres:</label>
```

```
<input type="text" name="nombres" id="idnombres">
```

```
<label>Apellido Paterno:</label>
```

```
<input type="text" name="appaterno" id="idappaterno">
```

```
<label>Apellido Materno:</label>
```

```
<input type="text" name="apmaterno" id="idapmaterno">
```

```
<label>Edad:</label>
```

```
<input type="text" name="edad" id="idedad">
```

```
<label>Sexo:</label>
```

```
<select name="sexo" id="idsexo">
```

```
  <option value="M">Masculino</option>
```

```
  <option value="F">Femenino</option>
```

```
  <option value="O">Otro</option>
```

```
</select>
```

```
<label>Telefono:</label>
```

```
<input type="text" name="telefono" id="idtelefono">
```

```
<label>Direccion:</label>
```

```
<input type="text" name="direccion" id="iddireccion">
```

```
<label>Tipo de Contacto:</label>
```

```
<select name="tipoContacto" id="idtipoContacto">
```

```
  <option value="CASA">Casa</option>
```

```
  <option value="TRABAJO">Trabajo</option>
```

```
</select>
```

```
  <button type="submit" class="btn-guardar" name="action" value="modificar">Guardar  
Cambios</button>
```

```
</form>
```

```
<h3>Lista de Contactos</h3>
```

```
<%
```

```
  List<EntidadUsuario> usuarios = (List<EntidadUsuario>)  
request.getAttribute("listausuarios");
```



```

        if (usuarios == null || usuarios.isEmpty()) {
%>

<div class="mensaje-vacio">No hay usuarios registrados.</div>

<% } else { %>

<table id="mitabla">
    <tr>
        <th>ID</th>
        <th>Nombres</th>
        <th>Apellido Paterno</th>
        <th>Apellido Materno</th>
        <th>Edad</th>
        <th>Sexo</th>
        <th>Telefono</th>
        <th>Direccion</th>
        <th>Tipo de Contacto</th>
    </tr>

    <% for (EntidadUsuario usuario : usuarios) { %>
        <tr>
            <td><%= usuario.getId() %></td>
            <td><%= usuario.getNombres() %></td>
            <td><%= usuario.getAppaterno() %></td>
            <td><%= usuario.getApmaterno() %></td>
            <td><%= usuario.getEdad() %></td>
            <td><%= usuario.getSexo() %></td>
            <td><%= usuario.getTelefono() %></td>
            <td><%= usuario.getDireccion() %></td>

```

```

        <td><%= usuario.getTipoContacto() %></td>

    </tr>

    <% } %>

</table>

<% } %>

</div>

<script>
    window.onload = function() {
        var table = document.getElementById("mitabla");

        for (var i = 1; i < table.rows.length; i++) {
            table.rows[i].onclick = function() {
                document.getElementById("idid").value = this.cells[0].innerHTML;
                document.getElementById("idnombres").value = this.cells[1].innerHTML;
                document.getElementById("idappaterno").value = this.cells[2].innerHTML;
                document.getElementById("idapmaterno").value = this.cells[3].innerHTML;
                document.getElementById("idedad").value = this.cells[4].innerHTML;
                document.getElementById("idsexo").value = this.cells[5].innerHTML;
                document.getElementById("idtelefono").value = this.cells[6].innerHTML;
                document.getElementById("iddireccion").value = this.cells[7].innerHTML;
                document.getElementById("idtipoContacto").value = this.cells[8].innerHTML;
            };
        }
    };
</script>

```

</body>
</html>

Melina SA

eliminarUsuarios.jsp

<%--

Document : eliminarUsuarios

Created on : 17 nov 2025, 10:19:18 p.m.

Author : melin

--%>

<%@page import="java.util.List"%>

<%@page import="Agenda.EntidadUsuario"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Eliminar Contactos</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;

background-color: #f3f4f6;

margin: 0;

padding: 0;

color: #333;

}

h1 {

text-align: center;

margin-top: 40px;

color: #222;

}

.contenedor {

```
width: 90%;
max-width: 950px;
margin: 30px auto;
background: #ffffff;
padding: 25px 35px;
border-radius: 10px;
box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}
```

```
.btn-regresar {
display: inline-block;
margin-bottom: 20px;
padding: 10px 18px;
background-color: #2563eb;
color: white;
text-decoration: none;
border-radius: 6px;
font-size: 14px;
transition: 0.25s;
}
```

```
.btn-regresar:hover {
background-color: #1e4f8e;
}
```

```
label {
display: block;
margin-top: 12px;
font-weight: bold;
}
```

```
input[type="text"] {
width: 60%;
```

```
padding: 10px;
border: 1px solid #ccc;
border-radius: 6px;
margin-top: 5px;
font-size: 14px;
background: #f7f7f7;
}
```

```
#idid {
  background: #e5e5e5;
}
```

```
.btn-eliminar {
  margin-top: 20px;
  padding: 12px 25px;
  background-color: #b91c1c;
  border: none;
  color: white;
  border-radius: 6px;
  cursor: pointer;
  font-size: 15px;
  transition: 0.2s;
}
```

```
.btn-eliminar:hover {
  background-color: #991b1b;
}
```

```
table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 25px;
}
```

```
th {  
  background-color: #2563eb;  
  color: white;  
  padding: 12px;  
  font-size: 14px;  
}
```

```
td {  
  padding: 10px;  
  background: #fafafa;  
  border-bottom: 1px solid #ddd;  
  text-align: center;  
}
```

```
tr:hover td {  
  background-color: #f0f4ff;  
  cursor: pointer;  
}
```

```
.mensaje-vacio {  
  text-align: center;  
  padding: 15px;  
  color: #555;  
}
```

```
select {  
  width: 60%;  
  padding: 10px;  
  border: 1px solid #ccc;  
  border-radius: 6px;  
  font-size: 14px;  
  background: white;  
  transition: 0.2s;
```

```

    }

    select:focus {
        border-color: #2563eb;
        background: #f0f4ff;
        outline: none;
    }
</style>
</head>

<body>

<h1>Eliminar Contactos</h1>

<div class="contenedor">

    <a class="btn-regresar" href="index.jsp">Regresar al Menú</a>
    <%
        String error = (String) request.getAttribute("error");
        if (error != null) {
    %>
        <div style="background:#ffdddd; color:#a33; padding:12px;
            border:1px solid #cc0000; border-radius:6px;
            margin-bottom:15px; font-weight:bold;">
            <%= error %>
        </div>
    <% } %>

    <form action="servletusuario" method="POST">

        <label>ID:</label>

        <input type="text" name="id" id="idid" readonly>

```


<label>Nombres:</label>

<input type="text" name="nombres" id="idnombres" readonly>

<label>Apellido Paterno:</label>

<input type="text" name="appaterno" id="idappaterno" readonly>

<label>Apellido Materno:</label>

<input type="text" name="apmaterno" id="idapmaterno" readonly>

<label>Edad:</label>

<input type="text" name="edad" id="idedad" readonly>

<label>Sexo:</label>

<select name="sexo" id="idsexo" disabled>

<option value="M">Masculino</option>

<option value="F">Femenino</option>

<option value="O">Otro</option>

</select>

<label>Telefono:</label>

<input type="text" name="telefono" id="idtelefono" readonly>

<label>Direccion:</label>

<input type="text" name="direccion" id="iddireccion" readonly>

<label>Tipo de Contacto:</label>

<select name="tipoContacto" id="idtipoContacto" disabled>

<option value="CASA">Casa</option>

<option value="TRABAJO">Trabajo</option>

</select>

<button type="submit" class="btn-eliminar" name="action" value="eliminar">

Eliminar Contacto

</button>

</form>

<h3>Lista de Contactos</h3>

<%

```
List<EntidadUsuario> usuarios = (List<EntidadUsuario>)
request.getAttribute("listausuarios");
```

```
if (usuarios == null || usuarios.isEmpty()) {
%>
```

<div class="mensaje-vacio">No hay contactos registrados.</div>

<% } else { %>

<table id="mitabla">

<tr>

<th>ID</th>

<th>Nombres</th>

<th>Apellido Paterno</th>

<th>Apellido Materno</th>

<th>Edad</th>

<th>Sexo</th>

<th>Telefono</th>

<th>Direccion</th>

<th>Tipo de Contacto</th>

</tr>

<% for (EntidadUsuario usuario : usuarios) { %>

<tr>

<td><%= usuario.getId() %></td>

<td><%= usuario.getNombres() %></td>

```

        <td><%= usuario.getAppaterno() %></td>
        <td><%= usuario.getApmaterno() %></td>
        <td><%= usuario.getEdad() %></td>
        <td><%= usuario.getSexo() %></td>
        <td><%= usuario.getTelefono() %></td>
        <td><%= usuario.getDireccion() %></td>
        <td><%= usuario.getTipoContacto() %></td>
    </tr>
    <% } %>
</table>

<% } %>

</div>

<script>
window.onload = function() {
    var table = document.getElementById("mitabla");

    for (var i = 1; i < table.rows.length; i++) {
        table.rows[i].onclick = function() {
            document.getElementById("idid").value = this.cells[0].innerHTML;
            document.getElementById("idnombres").value = this.cells[1].innerHTML;
            document.getElementById("idappaterno").value = this.cells[2].innerHTML;
            document.getElementById("idapmaterno").value = this.cells[3].innerHTML;
            document.getElementById("idedad").value = this.cells[4].innerHTML;
            document.getElementById("idsexo").value = this.cells[5].innerHTML;
            document.getElementById("idtelefono").value = this.cells[6].innerHTML;
            document.getElementById("iddireccion").value = this.cells[7].innerHTML;
            document.getElementById("idtipoContacto").value = this.cells[8].innerHTML;
        };
    }
};

```

</script>

</body>

</html>

Melina SA

Código Java (Source Packages - Agenda)

ConexionDAO.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Agenda;

import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author melin
 */
public class ConexionDAO {
    private static final String url="jdbc:mysql://localhost:3306/dbusur";
    private static final String username="root";
    private static final String pass="root";

    public static Connection obtenerConexion(){
        Connection conn = null;
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url,username,pass);
        }
        catch(Exception e){
            System.out.println("Error al conectarnos: " + e.toString());
        }
        return conn;
    }
}
```

```
public static void cerrarConexion(Connection conn){
    try{
        if(conn != null){
            conn.close();
        }
    }
    catch(Exception e){
        System.out.println("Error al conectarnos: " + e.toString());
    }
}

public static boolean probarConexion() {
    Connection conn = null;
    try {
        conn = obtenerConexion();
        return conn != null && !conn.isClosed();
    } catch (Exception e) {
        return false;
    } finally {
        cerrarConexion(conn);
    }
}
}
```

EntidadUsuario.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Agenda;

/**
 *
 * @author melin
 */
public class EntidadUsuario {
    int id;
    String nombres;
    String appaterno;
    String apmaterno;
    int edad;

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexo) {
        this.sexo = sexo;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}
```

```
}
```

```
public String getDireccion() {  
    return direccion;  
}
```

```
public void setDireccion(String direccion) {  
    this.direccion = direccion;  
}
```

```
public String getTipoContacto() {  
    return tipoContacto;  
}
```

```
public void setTipoContacto(String tipoContacto) {  
    this.tipoContacto = tipoContacto;  
}
```

```
String sexo;  
String telefono;  
String direccion;  
String tipoContacto;
```

```
public int getId() {  
    return id;  
}
```

```
public void setId(int id) {  
    this.id = id;  
}
```

```
public String getNombres() {  
    return nombres;
```



```
}

public void setNombres(String nombres) {
    this.nombres = nombres;
}

public String getAppaterno() {
    return appaterno;
}

public void setAppaterno(String appaterno) {
    this.appaterno = appaterno;
}

public String getApmaterno() {
    return apmaterno;
}

public void setApmaterno(String apmaterno) {
    this.apmaterno = apmaterno;
}

public int getEdad() {
    return edad;
}

public void setEdad(int edad) {
    this.edad = edad;
}
}
```

UsuarioDAO.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Agenda;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author melin
 */
public class UsuarioDAO {

    public void insertarUsuario(String nombre, String appaterno, String apmaterno, int edad, String
    sexo,String telefono,String direccion,String tipoContacto){

        Connection conn = null;
        PreparedStatement stmt = null;

        try{

            conn = ConexionDAO.obtenerConexion();

            String sql = "insert into
usuarios(nombres,appaterno,apmaterno,edad,sexo,telefono,direccion,tipoContacto) values
(?,?,?,?,?,?,?,?)";

            stmt = conn.prepareStatement(sql);

            stmt.setString(1, nombre);
            stmt.setString(2, appaterno);
```

```

        stmt.setString(3, apmaterno);
        stmt.setInt(4, edad);
        stmt.setString(5, sexo);
        stmt.setString(6, telefono);
        stmt.setString(7, direccion);
        stmt.setString(8, tipoContacto);

        stmt.executeUpdate();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    finally{
        ConexionDAO.cerrarConexion(conn);
        try{
            if(stmt != null)stmt.close();
        }
        catch(SQLException ex){
            ex.printStackTrace();
        }
    }
}

public List<EntidadUsuario> obtenenerUsuarios(){
    List<EntidadUsuario> usuarios = new ArrayList<>();
    Connection conn = null;
    PreparedStatement stmt = null;
    ResultSet rs=null;

    try{
        conn = ConexionDAO.obtenerConexion();
        String sql = "select * from usuarios;";
        stmt = conn.prepareStatement(sql);
    }
    catch(Exception e){
        e.printStackTrace();
    }
}

```

```

rs = stmt.executeQuery();

while(rs.next()){
    EntidadUsuario usuario = new EntidadUsuario();
    usuario.setId(rs.getInt("id"));
    usuario.setNombres(rs.getString("nombres"));
    usuario.setAppaterno(rs.getString("appaterno"));
    usuario.setApmaterno(rs.getString("apmaterno"));
    usuario.setEdad(rs.getInt("edad"));
    usuario.setSexo(rs.getString("sexo"));
    usuario.setTelefono(rs.getString("telefono"));
    usuario.setDireccion(rs.getString("direccion"));
    usuario.setTipoContacto(rs.getString("tipoContacto"));
    usuarios.add(usuario);

}
}
catch(Exception e){
    e.printStackTrace();
}
finally{
    ConexionDAO.cerrarConexion(conn);
    try{
        if(stmt != null)stmt.close();
    }
    catch(SQLException ex){
        ex.printStackTrace();
    }
    try{
        if(rs != null)rs.close();
    }
    catch(SQLException ex){
        ex.printStackTrace();
    }
}

```

```

        }
    }
    return usuarios;

}

public void eliminarUsuario(int id){
    Connection conn = null;
    PreparedStatement stmt = null;

    try{
        conn = ConexionDAO.obtenerConexion();
        String sql = "delete from usuarios where id = ?;";
        stmt = conn.prepareStatement(sql);

        stmt.setInt(1, id);

        stmt.executeUpdate();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    finally{
        ConexionDAO.cerrarConexion(conn);
        try{
            if(stmt != null)stmt.close();
        }
        catch(SQLException ex){
            ex.printStackTrace();
        }
    }
}

public void modificarUsuario(String nombre, String appaterno, String apmaterno, int edad, String
sexo,String telefono,String direccion,String tipoContacto, int id){
    Connection conn = null;

```

```

PreparedStatement stmt = null;

try{
    conn = ConexionDAO.obtenerConexion();

    String sql = "update usuarios set nombres = ?, appaterno = ?, apmaterno = ?, edad = ?,
sexo = ?, telefono = ?, direccion = ?, tipoContacto = ? where id = ?";

    stmt = conn.prepareStatement(sql);

    stmt.setString(1, nombre);
    stmt.setString(2, appaterno);
    stmt.setString(3, apmaterno);
    stmt.setInt(4, edad);
    stmt.setString(5, sexo);
    stmt.setString(6, telefono);
    stmt.setString(7, direccion);
    stmt.setString(8, tipoContacto);
    stmt.setInt(9, id);

    stmt.executeUpdate();
}
catch(Exception e){
    e.printStackTrace();
}
finally{
    ConexionDAO.cerrarConexion(conn);
    try{
        if(stmt != null)stmt.close();
    }
    catch(SQLException ex){
        ex.printStackTrace();
    }
}
}
}

```

Servletusuario.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */
package Agenda;

import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author melin
 */
@WebServlet(name = "servletusuario", urlPatterns = {"/servletusuario"})
public class servletusuario extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String action = request.getParameter("action");
        if("mostrar".equals(action)){
```

```

        mostrarUsuarios(request, response);
    }
    else if("guardar".equals(action)){
        mostrarUsuariosGuardar(request, response);
    }
    else if("modificar".equals(action)){
        mostrarUsuariosModificar(request, response);
    }
    else if("eliminar".equals(action)){
        mostrarUsuariosEliminar(request, response);
    }
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String action = request.getParameter("action");
    if("guardar".equals(action)){
        guardarUsuarios(request, response);
    }
    else if("modificar".equals(action)){
        modificarUsuarios(request, response);
    }
    else if("eliminar".equals(action)){
        eliminarUsuarios(request, response);
    }
}

@Override
public String getServletInfo() {
    return "Short description";
}

```



```

public void guardarUsuarios(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String nombres = request.getParameter("nombres");
    String appaterno = request.getParameter("appaterno");
    String apmaterno = request.getParameter("apmaterno");
    String edadStr = request.getParameter("edad");
    String sexo = request.getParameter("sexo");
    String telefono = request.getParameter("telefono");
    String direccion = request.getParameter("direccion");
    String tipoContacto = request.getParameter("tipoContacto");

    // ===== VALIDACIONES =====

    // Validación de campos vacíos
    if (nombres == null || nombres.trim().isEmpty()) {
        request.setAttribute("error", "El nombre no puede estar vacío.");
        mostrarUsuariosGuardar(request, response);
        return;
    }
    if (appaterno == null || appaterno.trim().isEmpty()) {
        request.setAttribute("error", "El apellido paterno no puede estar vacío.");
        mostrarUsuariosGuardar(request, response);
        return;
    }
    if (apmaterno == null || apmaterno.trim().isEmpty()) {
        request.setAttribute("error", "El apellido materno no puede estar vacío.");
        mostrarUsuariosGuardar(request, response);
        return;
    }

    // Validación de edad numérica
    int edad = 0;

```

```

try {
    edad = Integer.parseInt(edadStr);
    if (edad <= 0 || edad > 120) {
        request.setAttribute("error", "La edad debe ser un número entre 1 y 120.");
        mostrarUsuariosGuardar(request, response);
        return;
    }
} catch (NumberFormatException e) {
    request.setAttribute("error", "La edad debe ser un número válido.");
    mostrarUsuariosGuardar(request, response);
    return;
}

// Validación teléfono
// Validación teléfono
if (telefono == null || telefono.trim().isEmpty()) {
    request.setAttribute("error", "El teléfono no puede estar vacío.");
    mostrarUsuariosGuardar(request, response);
    return;
}

// Solo números
if (!telefono.matches("\\d+")) {
    request.setAttribute("error", "El teléfono solo puede contener números.");
    mostrarUsuariosGuardar(request, response);
    return;
}

// Longitud de 7 a 15 dígitos
if (telefono.length() < 7 || telefono.length() > 15) {
    request.setAttribute("error", "El teléfono debe tener entre 7 y 15 dígitos.");
    mostrarUsuariosGuardar(request, response);
    return;
}

```

```

    }

    // Validación dirección
    if (direccion == null || direccion.trim().isEmpty()) {
        request.setAttribute("error", "La dirección no puede estar vacía.");
        mostrarUsuariosGuardar(request, response);
        return;
    }

    // ===== SI TODO ES VÁLIDO, GUARDAR =====

    UsuarioDAO usuarioDAO = new UsuarioDAO();
    usuarioDAO.insertarUsuario(nombres, appaterno, apmaterno, edad, sexo, telefono, direccion,
    tipoContacto);

    mostrarUsuariosGuardar(request, response);
}

public void mostrarUsuarios(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    UsuarioDAO usuarioDAO = new UsuarioDAO();
    List<EntidadUsuario> usuarios = usuarioDAO.obtenerUsuarios();
    request.setAttribute("listausuarios", usuarios);

    request.getRequestDispatcher("mostrarUsuarios.jsp").forward(request, response);
}

public void mostrarUsuariosGuardar(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {

```

```

        UsuarioDAO usuarioDAO = new UsuarioDAO();
        List<EntidadUsuario> usuarios = usuarioDAO.obtenerUsuarios();
        request.setAttribute("listausuarios", usuarios);

        request.getRequestDispatcher("guardarUsuarios.jsp").forward(request, response);
    }

    public void modificarUsuarios(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {

        String idStr = request.getParameter("id");

        // Validación: si no seleccionaron un usuario
        if (idStr == null || idStr.trim().isEmpty()) {
            request.setAttribute("error", "Debe seleccionar un usuario de la tabla antes de modificar.");
            mostrarUsuariosModificar(request, response);
            return;
        }

        int id = Integer.parseInt(idStr);

        String nombres = request.getParameter("nombres");
        String appaterno = request.getParameter("appaterno");
        String apmaterno = request.getParameter("apmaterno");
        String edadStr = request.getParameter("edad");
        String sexo = request.getParameter("sexo");
        String telefono = request.getParameter("telefono");
        String direccion = request.getParameter("direccion");
        String tipoContacto = request.getParameter("tipoContacto");

        // ===== VALIDACIONES =====

```

```

if (nombres == null || nombres.trim().isEmpty()) {
    request.setAttribute("error", "El nombre no puede estar vacío.");
    mostrarUsuariosModificar(request, response);
    return;
}
if (appaterno == null || appaterno.trim().isEmpty()) {
    request.setAttribute("error", "El apellido paterno no puede estar vacío.");
    mostrarUsuariosModificar(request, response);
    return;
}
if (apmaterno == null || apmaterno.trim().isEmpty()) {
    request.setAttribute("error", "El apellido materno no puede estar vacío.");
    mostrarUsuariosModificar(request, response);
    return;
}

int edad = 0;
try {
    edad = Integer.parseInt(edadStr);
    if (edad <= 0 || edad > 120) {
        request.setAttribute("error", "La edad debe ser un número entre 1 y 120.");
        mostrarUsuariosModificar(request, response);
        return;
    }
} catch (NumberFormatException e) {
    request.setAttribute("error", "La edad debe ser un número válido.");
    mostrarUsuariosModificar(request, response);
    return;
}

// Validación teléfono
if (telefono == null || telefono.trim().isEmpty()) {
    request.setAttribute("error", "El teléfono no puede estar vacío.");

```

```
        mostrarUsuariosModificar(request, response);
        return;
    }

    // Solo números
    if (!telefono.matches("\\d+")) {
        request.setAttribute("error", "El teléfono solo puede contener números.");
        mostrarUsuariosModificar(request, response);
        return;
    }

    // Longitud de 7 a 15 dígitos
    if (telefono.length() < 7 || telefono.length() > 15) {
        request.setAttribute("error", "El teléfono debe tener entre 7 y 15 dígitos.");
        mostrarUsuariosModificar(request, response);
        return;
    }

    if (direccion == null || direccion.trim().isEmpty()) {
        request.setAttribute("error", "La dirección no puede estar vacía.");
        mostrarUsuariosModificar(request, response);
        return;
    }

    // ===== SI TODO ES VÁLIDO, MODIFICAR =====

    UsuarioDAO usuarioDAO = new UsuarioDAO();

    usuarioDAO.modificarUsuario(nombres, appaterno, apmaterno, edad, sexo, telefono,
    direccion, tipoContacto, id);

    mostrarUsuariosModificar(request, response);
}
```

```

public void eliminarUsuarios(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String idStr = request.getParameter("id");

    // Validación: si no se seleccionó un usuario
    if (idStr == null || idStr.trim().isEmpty()) {
        request.setAttribute("error", "Debe seleccionar un usuario de la tabla antes de eliminar.");
        mostrarUsuariosEliminar(request, response);
        return;
    }

    int id = Integer.parseInt(idStr);

    UsuarioDAO usuarioDAO = new UsuarioDAO();
    usuarioDAO.eliminarUsuario(id);

    mostrarUsuariosEliminar(request, response);
}

public void mostrarUsuariosModificar(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    UsuarioDAO usuarioDAO = new UsuarioDAO();
    List<EntidadUsuario> usuarios = usuarioDAO.obtenerUsuarios();
    request.setAttribute("listausuarios", usuarios);

    request.getRequestDispatcher("modificarUsuarios.jsp").forward(request, response);
}

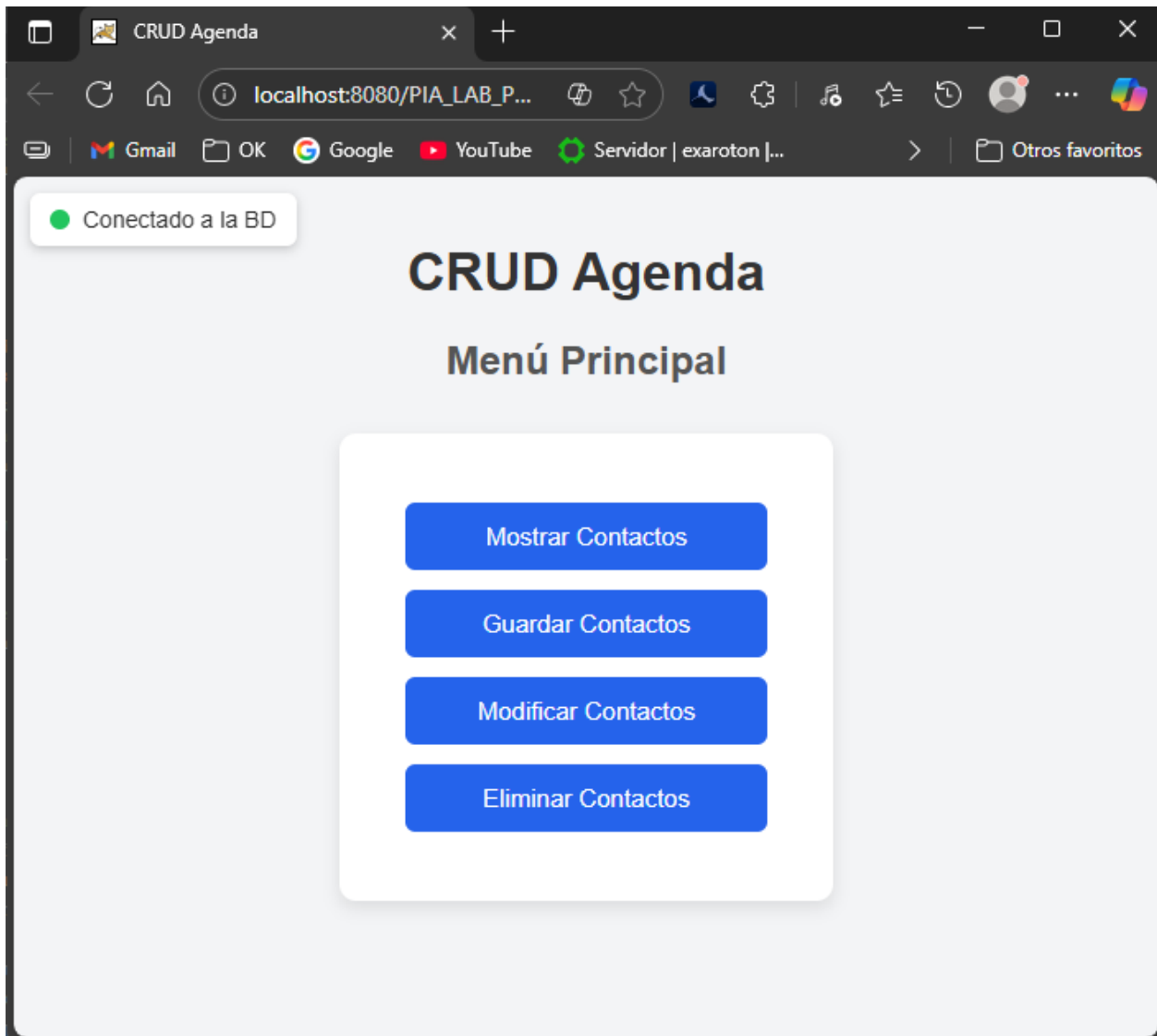
public void mostrarUsuariosEliminar(HttpServletRequest request, HttpServletResponse
response)

```

```
throws ServletException, IOException {  
    UsuarioDAO usuarioDAO = new UsuarioDAO();  
    List<EntidadUsuario> usuarios = usuarioDAO.obtenerUsuarios();  
    request.setAttribute("listausuarios", usuarios);  
  
    request.getRequestDispatcher("eliminarUsuarios.jsp").forward(request, response);  
}  
}
```


Capturas de Pantallas de la Pagina Web

Menú con conexión



Menú sin conexión

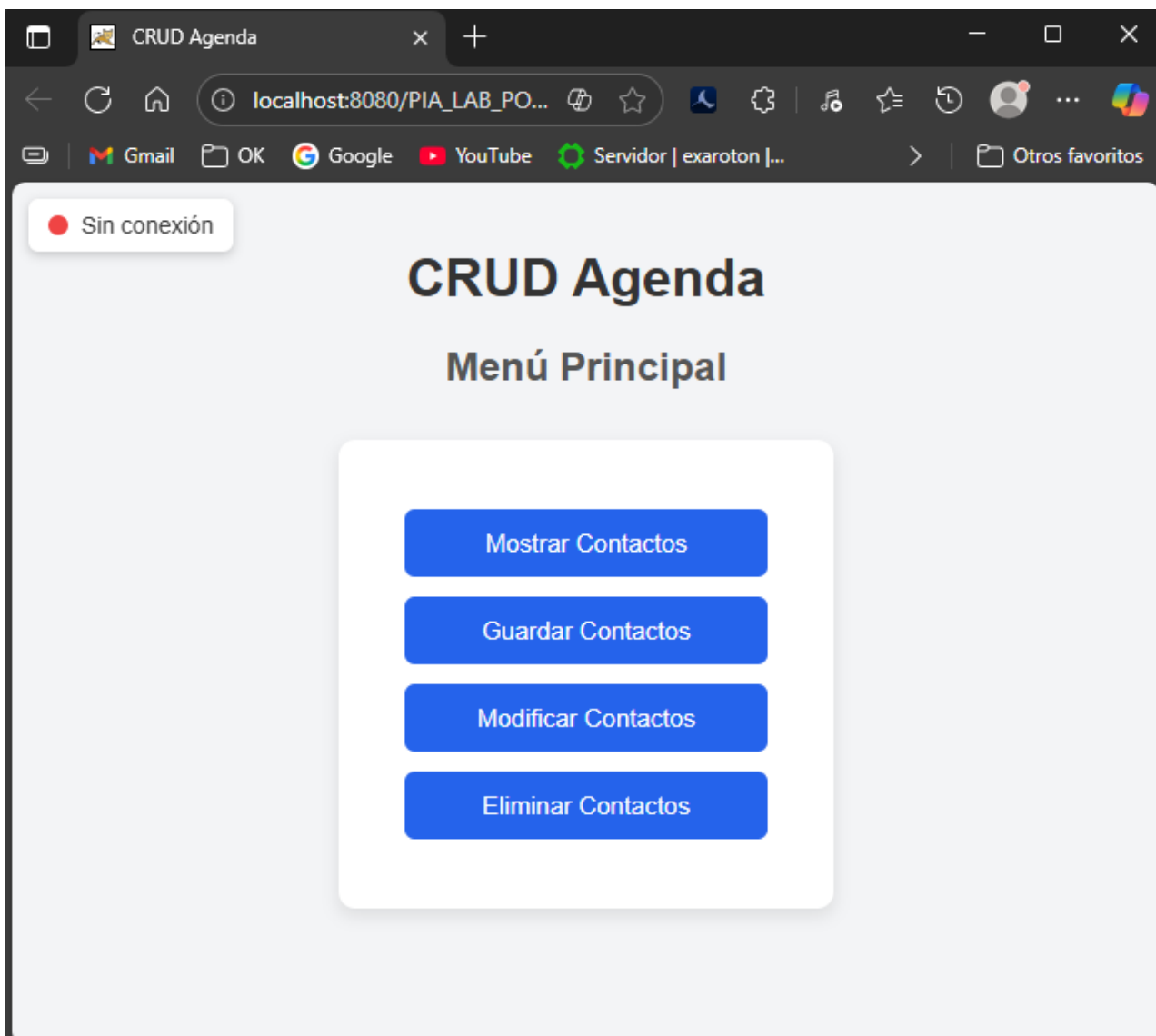
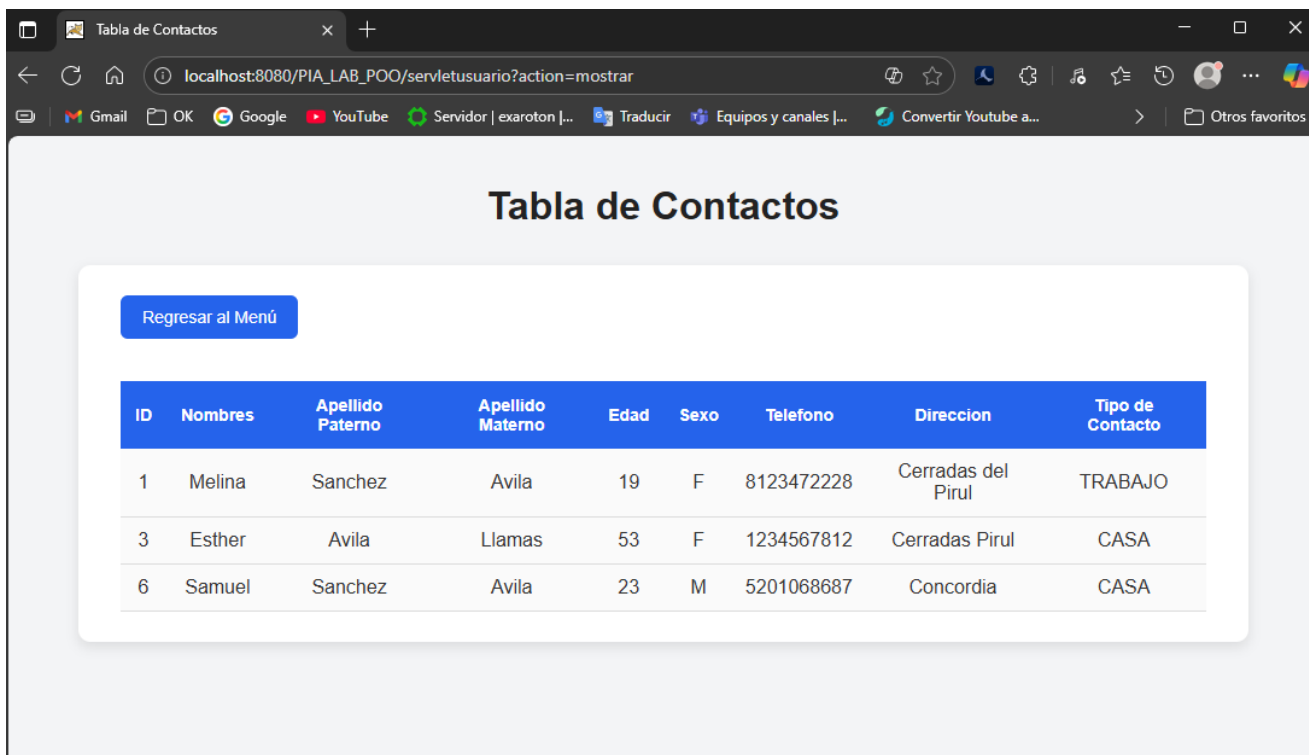


Tabla de Contactos



The screenshot shows a web browser window with the title 'Tabla de Contactos'. The address bar displays 'localhost:8080/PIA_LAB_POO/servletusuario?action=mostrar'. The browser's toolbar includes icons for back, forward, home, search, and various extensions like Gmail, OK, Google, YouTube, and a server status indicator. The main content area features a light gray background with a white box containing a blue button labeled 'Regresar al Menú' and a table with contact information.

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	19	F	8123472228	Cerradas del Pirul	TRABAJO
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA

Guardar Contactos

[Regresar al Menú](#)

Nombres:

Apellido Paterno:

Apellido Materno:

Edad:

Sexo:

Masculino ▼

Telefono:

Direccion:

Tipo de Contacto:

Casa ▼

[Guardar](#)

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	19	F	8123472228	Cerradas del Pirul	TRABAJO
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA

[Regresar al Menú](#)

Nombres:

Esteban

Apellido Paterno:

Sanchez

Apellido Materno:

Avila

Edad:

23

Sexo:

Masculino

Telefono:

12312312313

Direccion:

Cerradas Pirul

Tipo de Contacto:

Casa

[Guardar](#)

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	19	F	8123472228	Cerradas del Pirul	TRABAJO
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA
7	Esteban	Sanchez	Avila	23	M	12312312313	Cerradas Pirul	CASA

Modificar de Contactos

Modificar Contactos

Regresar al Menú

ID:

Nombres:

Apellido Paterno:

Apellido Materno:

Edad:

Sexo:

Masculino

Telefono:

Direccion:

Tipo de Contacto:

Casa

Guardar Cambios

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	19	F	8123472228	Cerradas del Pirul	TRABAJO
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA
7	Esteban	Sanchez	Avila	23	M	12312312313	Cerradas Pirul	CASA

[Regresar al Menú](#)

ID:

Nombres:

Apellido Paterno:

Apellido Materno:

Edad:

Sexo:

Telefono:

Direccion:

Tipo de Contacto:

[Guardar Cambios](#)

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	24	F	8123472228	Cerradas del Pirul	TRABAJO

Eliminar de Contactos

Eliminar Contactos

Regresar al Menú

ID:

1

Nombres:

Melina

Apellido Paterno:

Sanchez

Apellido Materno:

Avila

Edad:

24

Sexo:

Femenino

Telefono:

8123472228

Direccion:

Cerradas del Pirul

Tipo de Contacto:

Trabajo

Eliminar Contacto

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
1	Melina	Sanchez	Avila	24	F	8123472228	Cerradas del Pirul	TRABAJO
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA
7	Esteban	Sanchez	Avila	23	M	12312312313	Cerradas Pirul	CASA

Eliminar Contactos

[Regresar al Menú](#)

ID:

Nombres:

Apellido Paterno:

Apellido Materno:

Edad:

Sexo:

Telefono:

Direccion:

Tipo de Contacto:

[Eliminar Contacto](#)

Lista de Contactos

ID	Nombres	Apellido Paterno	Apellido Materno	Edad	Sexo	Telefono	Direccion	Tipo de Contacto
3	Esther	Avila	Llamas	53	F	1234567812	Cerradas Pirul	CASA
6	Samuel	Sanchez	Avila	23	M	5201068687	Concordia	CASA
7	Esteban	Sanchez	Avila	23	M	12312312313	Cerradas Pirul	CASA

Validaciones visuales

Regresar al Menú

El nombre no puede estar vacío.

Regresar al Menú

El apellido paterno no puede estar vacío.

Nombre:

Regresar al Menú

El apellido paterno no puede estar vacío.

Regresar al Menú

La edad debe ser un número válido.

Regresar al Menú

La edad debe ser un número entre 1 y 120.

Regresar al Menú

El teléfono no puede estar vacío.

Nombre:

Regresar al Menú

El teléfono debe tener entre 7 y 15 dígitos.

Regresar al Menú

El teléfono solo puede contener números.

Regresar al Menú

El teléfono solo puede contener números.

Validaciones visuales extra de modificar y eliminar

Regresar al Menú

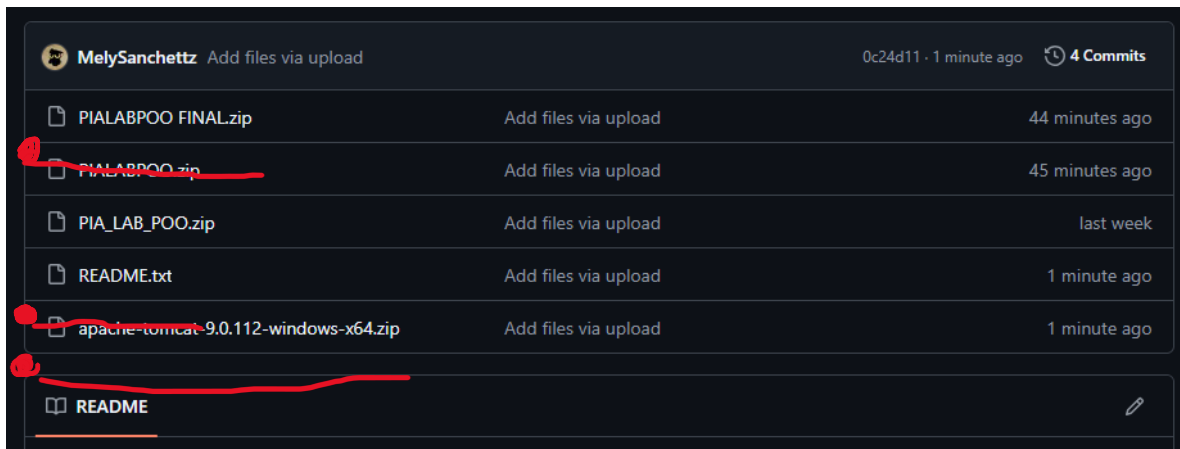
Debe seleccionar un usuario de la tabla antes de modificar.

Regresar al Menú

Debe seleccionar un usuario de la tabla antes de eliminar.

Repositorio GitHub

https://github.com/MelySanchettz/LAB_POO



Conclusión

El desarrollo de este proyecto permitió comprender y aplicar de manera práctica los conceptos fundamentales del desarrollo web con Java utilizando Servlets, JSP y MySQL. A través de la implementación de un CRUD para la gestión de usuarios, fue posible integrar correctamente la arquitectura **Modelo–Vista–Controlador (MVC)**, separando claramente la lógica de negocio, la presentación y el control del flujo de la aplicación.

El uso de JSP para la parte visual facilitó la creación de interfaces dinámicas, mientras que el servlet centralizó toda la lógica relacionada con la validación y procesamiento de datos. Por su parte, la interacción con MySQL mediante el patrón DAO aseguró un manejo ordenado y modular de la información almacenada en la base de datos. Esta estructura no solo hizo que el sistema fuera funcional, sino también escalable y más fácil de mantener.

En general, este proyecto fortaleció habilidades clave como el manejo de Servlets, la comunicación entre capas, la conexión a bases de datos y la correcta organización de un sistema web siguiendo buenas prácticas. Además, aportó una visión más clara de cómo funcionan las aplicaciones web del lado del servidor y sentó bases sólidas para futuros desarrollos utilizando frameworks más avanzados como Spring o Hibernate.