

**Universidad Tecnológica de Panamá  
Facultad de Ingeniería de Sistemas Computacionales  
Licenciatura en Ingeniería de Software  
Curso: Programación de Software II**

Integrantes:

Juan Rojas 8-993-1590

José Centella 8-1009-1127

Jonathan Reyes 20-23-7541

Maybeth Moreno 8- 960-27

Amid Ramirez 8- 980-312

Dilam Jimenez 8-976-637

Proyecto Final

Profesor: Paulo Picota Grupo: 1SF-131

05/07/2024



# Índice

<i>Índice</i>	<b>3</b>
<i>Introducción</i>	<b>4</b>
<i>Desarrollo</i>	<b>5</b>
<i>Wireframe</i>	<b>5</b>
Home	5
Panel de Usuario	5
Panel Admin	7
Login	7
<i>Front end</i>	<b>8</b>
Inicio	8
Panel de Usuario	8
Portada de Libro	9
Login	9
Panel de Bibliotecario	10
<i>Anexos</i>	<b>11</b>
<i>Base de datos</i>	<b>11</b>
<i>Código Fuente</i>	<b>12</b>
<i>Estilo</i>	
Login	
JSP	21
Connect cliente	
Connect	
Connect Login	
Estilos	44
Retos y dificultades	68
<i>Conclusión</i>	<b>69</b>
<i>Referencias bibliográficas</i>	<b>70</b>

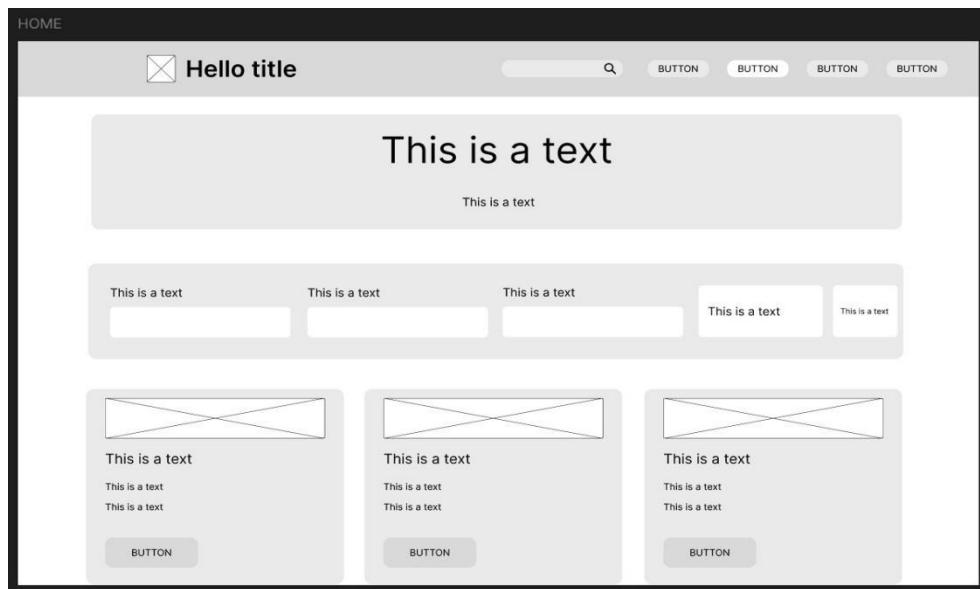
## Introducción

El proyecto consiste en diseñar e implementar una base de datos para una biblioteca universitaria, con el objetivo de centralizar y optimizar la gestión de recursos bibliográficos. Se manejará información clave como libros, usuarios, préstamos, autores, editoriales, bibliotecarios, sedes y carreras. El diseño se basará en un modelo entidad-relación y su implementación en un modelo lógico relacional, utilizando claves primarias y foráneas para asegurar la integridad. Se aplicarán formas normales para evitar redundancias y se emplearán comandos y funciones PL/SQL para automatizar la administración y consultas del sistema.

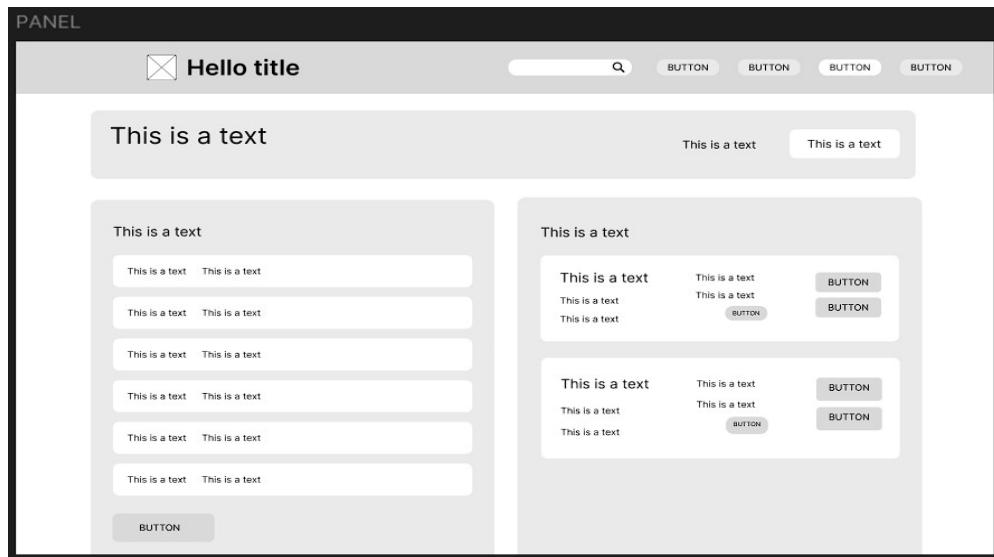
# Desarrollo

## Wireframe

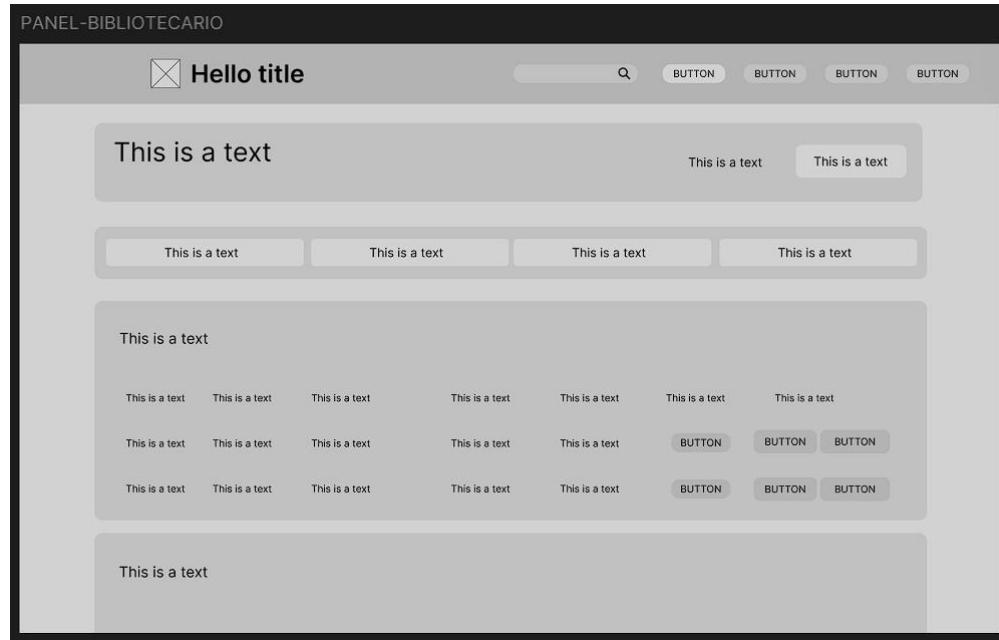
### Home



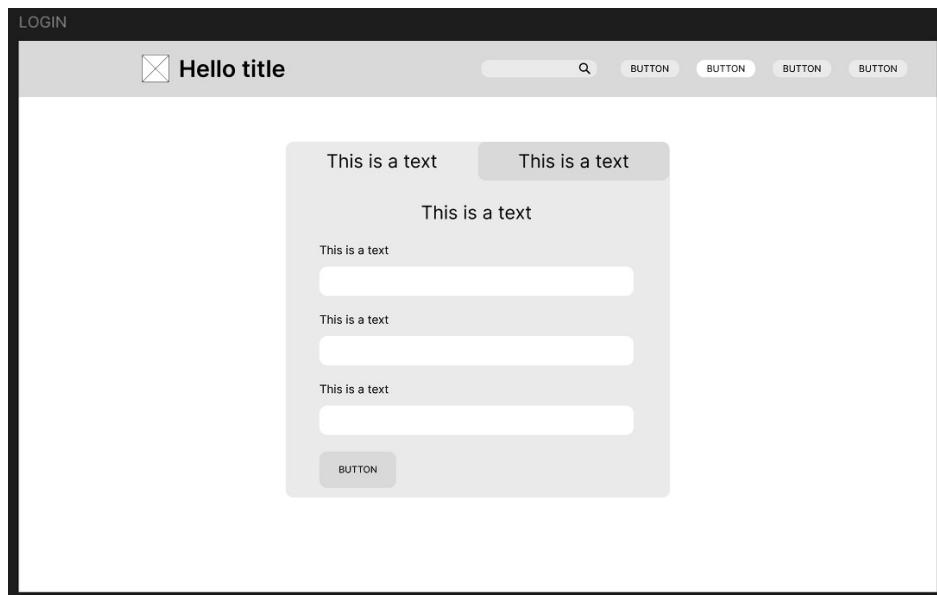
### Panel de Usuario



## Panel Admin



## Login



## Front end

### Inicio

The screenshot shows the homepage of the University Library. At the top, there is a blue header bar with the library's logo, name, and navigation links for 'Inicio', 'Catálogo', and 'Login'. Below the header, a main title 'Bienvenido a la Biblioteca Universitaria' is displayed, followed by a subtitle 'Explora nuestra colección de recursos bibliográficos'. There are two buttons: 'Explorar Catálogo' and 'Iniciar Sesión'. A search bar labeled 'Buscar libros, autores...' is present. Below the search bar is a 'Búsqueda Rápida' section with dropdown menus for 'Todas las sedes' and 'Todas las categorías', and a text input for 'Título del libro...'. A 'Buscar' button is located at the bottom of this section. The main content area features a section titled 'Libros Destacados' with three book covers: 'VIOLETA' by Isabel Allende, 'MARINA' by Marina, and 'El Alquimista'.

### Panel de Usuario

The screenshot shows the User Panel page. The top navigation bar is identical to the homepage. The main content area has a title 'Panel de Usuario' and a welcome message 'Bienvenido, Jonathan Reyes Estudiante'. It includes a 'Cerrar Sesión' link. On the left, there is a 'Datos Personales' section listing personal information: ID: 4, Nombre: Jonathan Reyes, Edad: 23 años, Carrera: Desarrollo de Software, Sede: Sede Este, Email: example@gmail.com, and Teléfono: 52187446. On the right, there is a 'Reservas Activas' section which states 'No tienes reservas activas.' and a 'Explorar Catálogo' button.

### Portada de Libro

The screenshot shows a book detail page for 'Cien años de soledad' by Gabriel García Márquez. At the top, there's a blue header bar with the library logo and navigation links for Inicio, Login, Mi Panel, and Bibliotecario. Below the header, a breadcrumb trail shows 'Inicio > Catálogo > Detalle del Libro'. The main content area features the book's title 'Cien años de soledad' in large bold letters, followed by the author's name 'Gabriel García Márquez'. Below the title, there are several data fields: ISBN: 978-3-16-148410-0, Editorial: Editorial Sudamericana, Año de Publicación: 1967; Edición: Primera, Páginas: 471, Idioma: Español. To the left of the title is a thumbnail image of the book cover. A section titled 'Descripción' provides a brief summary of the novel, mentioning it's a work by Gabriel García Márquez, Nobel laureate in Literature, set in Macondo across seven generations. A small note at the bottom right says 'Activar Windows'.

### Login

The screenshot shows the login page of the library system. The top navigation bar includes the library logo and links for Inicio, Login, Mi Panel, and Bibliotecario. The main content area has two buttons: 'Iniciar Sesión' and 'Registrarse'. Below these buttons is a form titled 'Iniciar Sesión' with fields for 'Identificación' (with a placeholder input field), 'Contraseña' (with a placeholder input field), and 'Tipo de Usuario' (with a dropdown menu showing 'Seleccionar...'). At the bottom of the form is a blue 'Iniciar Sesión' button. The background of the page features a faint watermark of a person reading a book.

## Panel de Bibliotecario

## Catálogo

# Anexos

## Base de datos

### Diagrama



# Código Fuente

## HTML

Index.HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>La Biblioteca del Conocimiento Universitario - Inicio</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header class="header">
<div class="container">
<div class="Logo">[REDACTED]

<h1>La Biblioteca del Conocimiento Universitario</h1>
</div>
<!-- Search Bar en Header -->
<div class="header-search">[REDACTED]
<form method="GET" action="catalogo-libros.jsp" class="search-form">
<input type="text" name="busqueda" placeholder="Buscar Libros, autores..." class="header-search-input">
<button type="submit" class="header-search-btn">
<svg width="20" height="20" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
<circle cx="11" cy="11" r="8"></circle>
<path d="m21 21-4.35-4.35"></path>
</svg>
</button>
</form>
</div>
<nav class="nav" id="main-nav">
<!-- La navegación se cargará dinámicamente -->
</nav>
</div>
</header>

<main class="main">
<div class="container">
<section class="hero">
<h2>Bienvenido a la Biblioteca del Conocimiento Universitario</h2>
```

```

<p>Explora nuestra colección de recursos bibliográficos</p>
<div class="hero-actions">
<a href="catalogo-libros.jsp" class="btn btn-primary">Explorar Catálogo</a>
<a href="Login.html" class="btn btn-outline" id="login-action">Iniciar Sesión</a>
</div>
</section>

<section class="quick-search">
<h3>Búsqueda Rápida</h3>
<form method="GET" action="catalogo-libros.jsp" class="quick-search-form">
<div class="search-filters">
<select name="sede" class="filter-select">
<option value="">Todas las sedes</option>
<option value="0">Sede Central</option>
<option value="2">Sede Este</option>
<option value="3">Sede Oeste</option>
<option value="4">Sede Norte</option>
</select>
<select name="categoria" class="filter-select">
<option value="">Todas las categorías</option>
<option value="Literatura">Literatura</option>
<option value="Ciencias">Ciencias</option>
<option value="Historia">Historia</option>
<option value="Tecnología">Tecnología</option>
<option value="Arte">Arte</option>
</select>
<input type="text" name="busqueda" placeholder="Título del libro..." class="search-input">
<button type="submit" class="btn btn-primary">Buscar</button>
</div>
</form>
</section>

<!-- Libros Destacados desde la Base de Datos -->
<section class="featured-books">
<h3>Libros Destacados</h3>
<div class="books-grid" id="featured-books-container">
<!-- Los libros se cargarán dinámicamente -->
</div>
</section>
</div>
</main>

<footer class="footer">
<div class="container">
<div class="footer-content">
<div class="footer-section">
<h3>Contacto</h3>
<p>Email: biblioteca@universidad.edu</p>
<p>Teléfono: +507 123-4567</p>

```

```
</div>
<div class="footer-section">
<h3>Horarios</h3>
<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>
<p>Sábados: 8:00 AM - 5:00 PM</p>
</div>
<div class="footer-section">
<h3>Sedes</h3>
<p>Sede Central - Panamá</p>
<p>Sede Este - Darién</p>
<p>Sede Oeste - Panamá Oeste</p>
<p>Sede Norte - Panamá</p>
</div>
</div>
</div>
</footer>

<script>
// Cargar navegación dinámica
document.addEventListener('DOMContentLoaded', function() {
loadNavigation();
loadFeaturedBooks();
});

function loadNavigation() {
fetch('api/check-session.jsp')
.then(response => response.json())
.then(data => {
const nav = document.getElementById('main-nav');
const loginAction = document.getElementById('login-action');
if (data.authenticated) {
// Usuario autenticado
let navHTML = '<a href="index.html" class="nav-link active">Inicio</a>';
navHTML += '<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>';
if (data.userType === 'bibliotecario') {
navHTML += '<a href="panel-bibliotecario.jsp" class="nav-link">Panel Bibliotecario</a>';
} else {
navHTML += '<a href="panel-usuario.jsp" class="nav-link">Mi Panel</a>';
}
nav.innerHTML = navHTML;
// Cambiar el botón de acción del hero
if (data.userType === 'bibliotecario') {
loginAction.href = 'panel-bibliotecario.jsp';
loginAction.textContent = 'Ir a Panel';
} else {
loginAction.href = 'panel-usuario.jsp';
loginAction.textContent = 'Mi Panel';
}
} else {
// Usuario no autenticado
nav.innerHTML = `

`});
}


```

```

<a href="index.html" class="nav-link active">Inicio</a>
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<a href="login.html" class="nav-link">Login</a>
`];
}
})
})
.catch(error => {
console.error('Error checking session:', error);
// Fallback para usuarios no autenticados
const nav = document.getElementById('main-nav');
nav.innerHTML = `
<a href="index.html" class="nav-link active">Inicio</a>
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<a href="login.html" class="nav-link">Login</a>
`;
});
}

function loadFeaturedBooks() {
fetch('api/libros-destacados.jsp')
.then(response => response.text())
.then(html => {
document.getElementById('featured-books-container').innerHTML = html;
})
.catch(error => {
console.error('Error cargando libros destacados:', error);
document.getElementById('featured-books-container').innerHTML =
'<p>Error al cargar los libros destacados.</p>';
});
}
</script>
</body>
</html>

```

## Login.HTML:

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login - La Biblioteca del Conocimiento Universitario</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header class="header">
<div class="container">
<div class="Logo">


```

```

<h1>La Biblioteca del Conocimiento Universitario</h1>
</div>
<!-- Search Bar en Header -->
<div class="header-search">
<form method="GET" action="catalogo-libros.jsp" class="search-form">
<input type="text" name="busqueda" placeholder="Buscar Libros, autores..." class="header-search-input">
<button type="submit" class="header-search-btn">
<svg width="20" height="20" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
<circle cx="11" cy="11" r="8"></circle>
<path d="m21 21-4.35-4.35"></path>
</svg>
</button>
</form>
</div>
<nav class="nav">
<a href="index.html" class="nav-link">Inicio</a>
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<a href="Login.html" class="nav-link active">Login</a>
</nav>
</div>
</header>

<main class="main">
<div class="container">
<div class="auth-container">
<div class="auth-tabs">
<button class="tab-btn active" onclick="showLogin(this)">Iniciar Sesión</button>
<button class="tab-btn" onclick="showRegister(this)">Registrarse</button>
</div>
<!-- Mostrar mensajes de error -->
<div id="message-container"></div>
<!-- Formulario de Login -->
<div id="Login-form" class="auth-form">
<h2>Iniciar Sesión</h2>
<form action="Login-process.jsp" method="POST">
<div class="form-group">
<label for="Login-email">Correo Electrónico:</label>
<input type="email" id="Login-email" name="email" required placeholder="ejemplo@correo.com">
</div>
<div class="form-group">
<label for="Login-password">Contraseña:</label>
<input type="password" id="Login-password" name="password" required placeholder="Ingrese su contraseña">
</div>
<div class="form-group">

```

```

<label for="user-type">Tipo de Usuario:</label>
<select id="user-type" name="tipo" required>
<option value="">Seleccionar tipo de usuario...</option>
<option value="estudiante">Estudiante</option>
<option value="bibliotecario">Bibliotecario</option>
</select>
</div>
<button type="submit" class="btn btn-primary">Iniciar Sesión</button>
<div class="login-help">
<p>¿No tienes cuenta? <a href="#" onclick="showRegister()">Regístrate aquí</a></p>
<div class="demo-accounts">
<h4>Cuentas de Prueba:</h4>
<p><strong>Estudiante:</strong> example@gmail.com / 123</p>
<p><strong>Bibliotecario:</strong> Ana.Martinez@correo.com / wsrg421GH</p>
</div>
</div>
</form>
</div>

<!-- Formulario de Registro -->
<div id="register-form" class="auth-form hidden">
<h2>Registrarse</h2>
<form action="registro-process.jsp" method="POST">
<div class="form-row">
<div class="form-group">
<label for="reg-nombre">Nombre:</label>
<input type="text" id="reg-nombre" name="nombre" required placeholder="Tu nombre">
</div>
<div class="form-group">
<label for="reg-apellido">Apellido:</label>
<input type="text" id="reg-apellido" name="apellido" required placeholder="Tu apellido">
</div>
</div>
<div class="form-group">
<label for="reg-fecha-nac">Fecha de Nacimiento:</label>
<input type="date" id="reg-fecha-nac" name="fecha_nacimiento" required>
</div>
<div class="form-row">
<div class="form-group">
<label for="reg-carrera">Carrera:</label>
<select id="reg-carrera" name="carrera" required>
<option value="">Seleccionar carrera...</option>
<!-- Las carreras se cargarán dinámicamente -->
</select>
</div>
<div class="form-group">
<label for="reg-sede">Sede:</label>
<select id="reg-sede" name="sede" required>
<option value="">Seleccionar sede...</option>

```

```

<!-- Las sedes se cargarán dinámicamente -->
</select>
</div>
</div>
<div class="form-row">
<div class="form-group">
<label for="reg-telefono">Teléfono:</label>
<input type="tel" id="reg-telefono" name="telefono" required
placeholder="60000000">
</div>
<div class="form-group">
<label for="reg-email">Correo Electrónico:</label>
<input type="email" id="reg-email" name="email" required
placeholder="ejemplo@correo.com">
</div>
</div>
<div class="form-group">
<label for="reg-password">Contraseña:</label>
<input type="password" id="reg-password" name="password" required
placeholder="Ingrese una contraseña">
</div>
<button type="submit" class="btn btn-primary">Registrarse</button>
<div class="login-help">
<p>¿Ya tienes cuenta? <a href="#" onclick="showLogin()">Inicia sesión aquí</a></p>
</div>
</form>
</div>
</div>
</div>
</main>

```

```

<footer class="footer">
<div class="container">
<div class="footer-content">
<div class="footer-section">
<h3>Contacto</h3>
<p>Email: biblioteca@universidad.edu</p>
<p>Teléfono: +507 123-4567</p>
</div>
<div class="footer-section">
<h3>Horarios</h3>
<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>
<p>Sábados: 8:00 AM - 5:00 PM</p>
</div>
<div class="footer-section">
<h3>Sedes</h3>
<p>Sede Central - Panamá</p>
<p>Sede Este - Darién</p>
<p>Sede Oeste - Panamá Oeste</p>
<p>Sede Norte - Panamá</p>
</div>

```

```

</div>
</div>
</footer>

<script>
function showLogin(element) {
document.getElementById("login-form").classList.remove("hidden");
document.getElementById("register-form").classList.add("hidden");
// Actualizar tabs
document.querySelectorAll(".tab-btn").forEach(btn =>
btn.classList.remove("active"));
if (element) {
element.classList.add("active");
} else {
document.querySelector(".tab-btn").classList.add("active");
}
}

function showRegister(element) {
document.getElementById("login-form").classList.add("hidden");
document.getElementById("register-form").classList.remove("hidden");
// Actualizar tabs
document.querySelectorAll(".tab-btn").forEach(btn =>
btn.classList.remove("active"));
if (element) {
element.classList.add("active");
} else {
document.querySelectorAll(".tab-btn")[1].classList.add("active");
}
// Cargar datos dinámicos para el registro
loadCarreras();
loadSedes();
}

function loadCarreras() {
console.log('Cargando carreras...');
fetch('api/get-carreras.jsp')
.then(response => {
console.log('Respuesta carreras:', response.status);
return response.text();
})
.then(html => {
console.log('HTML carreras:', html);
const select = document.getElementById('reg-carrera');
if (select) {
select.innerHTML = '<option value="">Seleccionar carrera...</option>' + html;
}
})
.catch(error => {
console.error('Error cargando carreras:', error);
const select = document.getElementById('reg-carrera');
}
)
}

```

```
if (select) {
    select.innerHTML = '<option value="">Error al cargar carreras</option>';
}
});
}

function loadSedes() {
    console.log('Cargando sedes...');
    fetch('api/get-sedes.jsp')
        .then(response => {
            console.log('Respuesta sedes:', response.status);
            return response.text();
        })
        .then(html => {
            console.log('HTML sedes:', html);
            const select = document.getElementById('reg-sede');
            if (select) {
                select.innerHTML = '<option value="">Seleccionar sede...</option>' + html;
            }
        })
        .catch(error => {
            console.error('Error cargando sedes:', error);
            const select = document.getElementById('reg-sede');
            if (select) {
                select.innerHTML = '<option value="">Error al cargar sedes</option>';
            }
        });
}

// Mostrar mensajes de error/éxito
document.addEventListener('DOMContentLoaded', function() {
    const urlParams = new URLSearchParams(window.location.search);
    const error = urlParams.get('error');
    const success = urlParams.get('success');
    const messageContainer = document.getElementById('message-container');
    if (error) {
        let message = '';
        switch(error) {
            case 'usuario_no_encontrado':
                message = 'Correo electrónico no encontrado o no corresponde a un estudiante';
                break;
            case 'bibliotecario_no_encontrado':
                message = 'Correo electrónico no encontrado o no corresponde a un bibliotecario';
                break;
            case 'contraseña_incorrecta':
                message = 'Contraseña incorrecta';
                break;
            case 'tipo_usuario_invalido':
                message = 'Tipo de usuario no válido';
                break;
            default:
        }
    }
});
```

```

message = 'Error: ' + decodeURIComponent(error);
}
messageContainer.innerHTML = '<div class="error-message">' + message +
'</div>';
}
if (success) {
let message = '';
switch(success) {
case 'registro':
message = 'Registro exitoso. Ya puedes iniciar sesión.';
break;
case 'logout':
message = 'Sesión cerrada correctamente.';
break;
default:
message = 'Operación exitosa';
}
messageContainer.innerHTML = '<div class="success-message">' + message +
'</div>';
}
});
</script>
</body>
</html>

```

## JSP

### Agregar-Libro.JSP:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<%
// Verificar autenticación de bibliotecario
if (session.getAttribute("usuario_id") == null ||
!"bibliotecario".equals(session.getAttribute("tipo_usuario"))) {
response.sendRedirect("login.html");
return;
}
// Procesar formulario si es POST
if (request.getMethod().equals("POST")) {
Connection conn = null;
CallableStatement cstmt = null;
try {
conn = getConnection();
String sql = "{CALL carga_libro(?, ?, ?, ?, ?, ?, ?, ?, ?)}";
cstmt = conn.prepareCall(sql);
cstmt.setString(1, request.getParameter("titulo"));
cstmt.setInt(2, Integer.parseInt(request.getParameter("ano_publicacion")));
}

```

```

cstmt.setInt(3, Integer.parseInt(request.getParameter("edicion")));
cstmt.setString(4, request.getParameter("isbn"));
cstmt.setInt(5, Integer.parseInt(request.getParameter("sede")));
cstmt.setInt(6, Integer.parseInt(request.getParameter("editorial")));
cstmt.setString(7, request.getParameter("autor_nombre"));
cstmt.setString(8, request.getParameter("autor_apellido"));
cstmt.setInt(9, Integer.parseInt(request.getParameter("cantidad")));
cstmt.execute();
response.sendRedirect("panel-bibliotecario.jsp?success=libro_agregado");
return;
} catch (Exception e) {
request.setAttribute("error", "Error al agregar libro: " + e.getMessage());
} finally {
try {
if (cstmt != null) cstmt.close();
if (conn != null) conn.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
}
%>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Agregar Libro - Biblioteca Universitaria</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header class="header">
<div class="container">
<div class="Logo">

<h1>Biblioteca Universitaria</h1>
</div>
<nav class="nav">
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<a href="panel-bibliotecario.jsp" class="nav-link">Panel Bibliotecario</a>
<a href="Logout.jsp" class="nav-link">Cerrar Sesión</a>
</nav>
</div>
</header>

<main class="main">
<div class="container">
<div class="auth-container">
<h2>Agregar Nuevo Libro</h2>
<%
if (request.getAttribute("error") != null) {

```

```
out.println("<div class='error-message'>" + request.getAttribute("error") +
"</div>");}
%>
<form method="POST" class="auth-form">
<div class="form-row">
<div class="form-group">
<label for="titulo">Título del Libro:</label>
<input type="text" id="titulo" name="titulo" required>
</div>
<div class="form-group">
<label for="isbn">ISBN:</label>
<input type="text" id="isbn" name="isbn" required>
</div>
</div>
<div class="form-row">
<div class="form-group">
<label for="autor_nombre">Nombre del Autor:</label>
<input type="text" id="autor_nombre" name="autor_nombre" required>
</div>
<div class="form-group">
<label for="autor_apellido">Apellido del Autor:</label>
<input type="text" id="autor_apellido" name="autor_apellido" required>
</div>
</div>
<div class="form-group">
<label for="editorial">Editorial:</label>
<select id="editorial" name="editorial" required>
<option value="">Seleccionar editorial...</option>
<%
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;
try {
conn = getConnection();
String sql = "SELECT cod_editorial, edi_nombre FROM editorial ORDER BY
edi_nombre";
pstmt = conn.prepareStatement(sql);
rs = pstmt.executeQuery();
while (rs.next()) {
out.println("<option value='" + rs.getInt("cod_editorial") + "'>" +
rs.getString("edi_nombre") + "</option>");
}
} catch (Exception e) {
e.printStackTrace();
} finally {
closeResources(conn, pstmt, rs);
}
%>
</select>
</div>
<div class="form-row">
```

```
<div class="form-group">
<label for="ano_publicacion">Año de Publicación:</label>
<input type="number" id="ano_publicacion" name="ano_publicacion" min="1000"
max="2024" required>
</div>
<div class="form-group">
<label for="edicion">Edición:</label>
<input type="number" id="edicion" name="edicion" min="1" required>
</div>
</div>
<div class="form-row">
<div class="form-group">
<label for="sede">Sede:</label>
<select id="sede" name="sede" required>
<option value="">Seleccionar sede...</option>
<%
try {
conn = getConnection();
String sql = "SELECT cod_sede, sed_nombre FROM sede ORDER BY sed_nombre";
pstmt = conn.prepareStatement(sql);
rs = pstmt.executeQuery();
while (rs.next()) {
out.println("<option value='" + rs.getInt("cod_sede") + "'>" +
rs.getString("sed_nombre") + "</option>");
}
} catch (Exception e) {
e.printStackTrace();
} finally {
closeResources(conn, pstmt, rs);
}
%>
</select>
</div>
<div class="form-group">
<label for="cantidad">Cantidad de Ejemplares:</label>
<input type="number" id="cantidad" name="cantidad" min="1" required>
</div>
</div>
<div class="form-actions">
<a href="panel-bibliotecario.jsp" class="btn btn-secondary">Cancelar</a>
<button type="submit" class="btn btn-primary">Agregar Libro</button>
</div>
</div>
</main>

<footer class="footer">
<div class="container">
<div class="footer-content">
<div class="footer-section">
<h3>Contacto</h3>
```

```

<p>Email: biblioteca@universidad.edu</p>
<p>Teléfono: +507 123-4567</p>
</div>
<div class="footer-section">
<h3>Horarios</h3>
<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>
<p>Sábados: 8:00 AM - 5:00 PM</p>
</div>
<div class="footer-section">
<h3>Sedes</h3>
<p>Sede Central - Panamá</p>
<p>Sede Este - Darién</p>
<p>Sede Oeste - Panamá Oeste</p>
</div>
</div>
</div>
</body>
</html>

```

## Catalogo-libros.JSP:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Catálogo de Libros - La Biblioteca del Conocimiento Universitario</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header class="header">
<div class="container">
<div class="Logo">

<h1>La Biblioteca del Conocimiento Universitario</h1>
</div>
<!-- Search Bar en Header -->
<div class="header-search">
<form method="GET" action="catalogo-libros.jsp" class="search-form">
<input type="text" name="busqueda" placeholder="Buscar Libros, autores..." class="header-search-input">

```

```

value=<%= request.getParameter("busqueda") != null ?  

request.getParameter("busqueda") : "" %>>  

<button type="submit" class="header-search-btn">  

<svg width="20" height="20" viewBox="0 0 24 24" fill="none"  

stroke="currentColor" stroke-width="2">  

<circle cx="11" cy="11" r="8"></circle>  

<path d="m21 21-4.35-4.35"></path>  

</svg>  

</button>  

</form>  

</div>  

<nav class="nav">  

<a href="index.html" class="nav-link">Inicio</a>  

<a href="catalogo-libros.jsp" class="nav-link active">Catálogo</a>  

<%  

if (session.getAttribute("usuario_id") != null) {  

String tipoUsuario = (String) session.getAttribute("tipo_usuario");  

if ("bibliotecario".equals(tipoUsuario)) {  

out.println("<a href='panel-bibliotecario.jsp' class='nav-link'>Panel  

Bibliotecario</a>");  

} else {  

out.println("<a href='panel-usuario.jsp' class='nav-link'>Mi Panel</a>");  

}  

out.println("<a href='logout.jsp' class='nav-link'>Cerrar Sesión</a>");  

} else {  

out.println("<a href='login.html' class='nav-link'>Login</a>");  

}  

%>  

</nav>  

</div>  

</header>

<main class="main">  

<div class="container">  

<section class="hero">  

<h2>Catálogo de Libros Disponibles</h2>  

<p>Explora nuestra colección de recursos bibliográficos</p>  

</section>

<section class="filters">  

<form method="GET" action="catalogo-libros.jsp">  

<div class="filter-group">  

<label for="sede-filter">Sede:</label>  

<select id="sede-filter" name="sede" class="filter-select">  

<option value="">Todas las sedes</option>  

<%  

Connection conn = null;  

PreparedStatement pstmt = null;  

ResultSet rs = null;  

try {  

conn = getConnection();  

String sql = "SELECT cod_sede, sed_nombre FROM sede ORDER BY sed_nombre";  


```

```

 pstmt = conn.prepareStatement(sql);
 rs = pstmt.executeQuery();
 String sedeSelected = request.getParameter("sede");
 while (rs.next()) {
 String selected = String.valueOf(rs.getInt("cod_sede")).equals(sedeSelected) ?
 "selected" : "";
 out.println("<option value='" + rs.getInt("cod_sede") + "' " + selected + ">" +
 rs.getString("sed_nombre") + "</option>");
 }
 } catch (Exception e) {
 e.printStackTrace();
 } finally {
 closeResources(conn, pstmt, rs);
 }
%>
</select>
</div>

<div class="filter-group">
<label for="autor-filter">Autor:</label>
<select id="autor-filter" name="autor" class="filter-select">
<option value="">Todos los autores</option>
<%
try {
conn = getConnection();
String sql = "SELECT cod_autor, CONCAT(au_nombre, ' ', au_apellido) as
nombre_completo FROM autor ORDER BY au_nombre";
pstmt = conn.prepareStatement(sql);
rs = pstmt.executeQuery();
String autorSelected = request.getParameter("autor");
while (rs.next()) {
String selected = String.valueOf(rs.getInt("cod_autor")).equals(autorSelected) ?
"selected" : "";
out.println("<option value='" + rs.getInt("cod_autor") + "' " + selected + ">" +
rs.getString("nombre_completo") + "</option>");
}
} catch (Exception e) {
e.printStackTrace();
} finally {
closeResources(conn, pstmt, rs);
}
%>
</select>
</div>

<div class="search-group">
<input type="text" name="busqueda" placeholder="Buscar Libros..." 
class="search-input" 
value="<%= request.getParameter("busqueda") != null ? 
request.getParameter("busqueda") : "" %>">

```

```

<button type="submit" class="search-btn">Buscar</button>
</div>
</form>
</section>

<section class="books-grid">
<%
try {
conn = getConnection();
// Construir consulta dinámica
StringBuilder sqlBuilder = new StringBuilder();
sqlBuilder.append("SELECT l.* , a.au_nombre, a.au_apellido, s.sed_nombre,
e.edi_nombre, l.url_portada, ");
sqlBuilder.append("i.cantidad_libros as cantidad ");
sqlBuilder.append("FROM libro l ");
sqlBuilder.append("JOIN autor a ON l.cod_autor = a.cod_autor ");
sqlBuilder.append("JOIN sede s ON l.cod_sede = s.cod_sede ");
sqlBuilder.append("JOIN editorial e ON l.cod_editorial = e.cod_editorial ");
sqlBuilder.append("LEFT JOIN inventario i ON l.cod_libro = i.cod_libro ");
sqlBuilder.append("WHERE i.cantidad_libros > 0 ");
String sedeParam = request.getParameter("sede");
String autorParam = request.getParameter("autor");
String busquedaParam = request.getParameter("busqueda");
if (sedeParam != null && !sedeParam.isEmpty()) {
sqlBuilder.append("AND l.cod_sede = ? ");
}
if (autorParam != null && !autorParam.isEmpty()) {
sqlBuilder.append("AND l.cod_autor = ? ");
}
if (busquedaParam != null && !busquedaParam.isEmpty()) {
sqlBuilder.append("AND (l.lib_nombre LIKE ? OR CONCAT(a.au_nombre, ' ',
a.au_apellido) LIKE ? ) ");
}
sqlBuilder.append("ORDER BY l.lib_nombre");
pstmt = conn.prepareStatement(sqlBuilder.toString());
int paramIndex = 1;
if (sedeParam != null && !sedeParam.isEmpty()) {
pstmt.setInt(paramIndex++, Integer.parseInt(sedeParam));
}
if (autorParam != null && !autorParam.isEmpty()) {
pstmt.setInt(paramIndex++, Integer.parseInt(autorParam));
}
if (busquedaParam != null && !busquedaParam.isEmpty()) {
pstmt.setString(paramIndex++, "%" + busquedaParam + "%");
pstmt.setString(paramIndex++, "%" + busquedaParam + "%");
}
rs = pstmt.executeQuery();
boolean hasResults = false;
while (rs.next()) {
hasResults = true;
String urlPortada = rs.getString("url_portada");
if (urlPortada == null || urlPortada.trim().isEmpty()) {

```

```

urlPortada = "/placeholder.svg?height=200&width=150";
}%
out.println("<div class='book-card'>");
out.println("<img src='" + urlPortada + "' alt=''" + rs.getString("lib_nombre") +
+ "' onerror='this.src='/placeholder.svg?height=200&width=150\'>");%
out.println("<div class='book-info'>");%
out.println("<h4>" + rs.getString("lib_nombre") + "</h4>");%
out.println("<p>" + rs.getString("au_nombre") + " " +
rs.getString("au_apellido") + "</p>");%
out.println("<div class='book-meta'>");%
out.println("<span>ISBN: " + rs.getString("lib_isbn") + "</span>");%
out.println("<span class='stock'>" + rs.getInt("cantidad") + " "
disponibles</span>");%
out.println("<span>Sede: " + rs.getString("sed_nombre") + "</span>");%
out.println("<span>Editorial: " + rs.getString("edi_nombre") + "</span>");%
out.println("</div>");%
if (session.getAttribute("usuario_id") != null) {%
out.println("<a href='detalle-libro.jsp?id=" + rs.getInt("cod_libro") + "' "
class='btn btn-outline'>Ver Detalle</a>");%
} else {%
out.println("<a href='login.html' class='btn btn-outline'>Iniciar Sesión para "
Reservar</a>");%
}%
out.println("</div>");%
out.println("</div>");%
}%
if (!hasResults) {%
out.println("<div class='no-results'>");%
out.println("<h3>No se encontraron libros</h3>");%
out.println("<p>Intenta con otros filtros de búsqueda.</p>");%
out.println("</div>");%
}%
} catch (Exception e) {%
out.println("<p>Error al cargar los libros: " + e.getMessage() + "</p>");%
e.printStackTrace();%
} finally {%
closeResources(conn, pstmt, rs);%
}%
%>
</section>%
</div>%
</main>

<footer class="footer">
<div class="container">
<div class="footer-content">
<div class="footer-section">
<h3>Contacto</h3>%
<p>Email: biblioteca@universidad.edu</p>%
<p>Teléfono: +507 123-4567</p>%
</div>%
<div class="footer-section">

```

```

<h3>Horarios</h3>
<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>
<p>Sábados: 8:00 AM - 5:00 PM</p>
</div>
<div class="footer-section">
<h3>Sedes</h3>
<p>Sede Central - Panamá</p>
<p>Sede Este - Darién</p>
<p>Sede Oeste - Panamá Oeste</p>
</div>
</div>
</div>
</body>
</html>

```

## Detalle-Cliente.JSP:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Detalle del Libro - Biblioteca Universitaria</title>
<link rel="stylesheet" href="styles.css">
<style>
.book-cover {
width: 300px;
height: 400px;
object-fit: cover;
border: 1px solid #ccc;
}
.separator {
margin: 40px 0;
height: 2px;
background-color: #e0e0e0;
}
</style>
</head>
<body>
<header class="header">
<div class="container">
<div class="logo">

<h1>Biblioteca Universitaria</h1>

```

```

</div>
<nav class="nav">
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<%
if (session.getAttribute("usuario_id") != null) {
String tipoUsuario = (String) session.getAttribute("tipo_usuario");
if ("bibliotecario".equals(tipoUsuario)) {
out.println("<a href='panel-bibliotecario.jsp' class='nav-link'>Panel
Bibliotecario</a>");
} else {
out.println("<a href='panel-usuario.jsp' class='nav-link'>Mi Panel</a>");
}
} else {
out.println("<a href='login.html' class='nav-link'>Login</a>");
}
%>
</nav>
</div>
</header>

<main class="main">
<div class="container">
<div class="breadcrumb">
<a href="catalogo-libros.jsp">Catálogo</a> > Detalle del Libro
</div>

<%
String libroId = request.getParameter("id");
if (libroId != null) {
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;

try {
conn = getConnection();
String sql = "SELECT l.*, a.au_nombre, a.au_apellido, s.sed_nombre,
e.edi_nombre, " +
"i.cantidad_libros as cantidad, l.url_portada, l.lib_descripcion " +
"FROM libro l " +
"JOIN autor a ON l.cod_autor = a.cod_autor " +
"JOIN sede s ON l.cod_sede = s.cod_sede " +
"JOIN editorial e ON l.cod_editorial = e.cod_editorial " +
"LEFT JOIN inventario i ON l.cod_libro = i.cod_libro " +
"WHERE l.cod_libro = ?;";
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, Integer.parseInt(libroId));
rs = pstmt.executeQuery();

if (rs.next()) {
String urlPortada = rs.getString("url_portada");
if (urlPortada == null || urlPortada.trim().isEmpty()) {

```

```
urlPortada = "/placeholder.svg?height=400&width=300";
}

String descripcion = rs.getString("lib_descripcion");
if (descripcion == null || descripcion.trim().isEmpty()) {
    descripcion = "Descripción no disponible para este libro.";
}

out.println("<div class='book-detail'>");
out.println("<div class='book-image-section'>");
out.println("<img src='" + urlPortada + "' alt='" + rs.getString("lib_nombre") +
+ "' class='book-cover'>");[REDACTED]
out.println("</div>");

out.println("<div class='book-info-section'>");[REDACTED]
out.println("<h1>" + rs.getString("lib_nombre") + "</h1>");[REDACTED]
out.println("<h2>" + rs.getString("au_nombre") + " " + [REDACTED]
rs.getString("au_apellido") + "</h2>");[REDACTED]

out.println("<div class='book-meta'>");[REDACTED]
out.println("<div class='meta-item'><strong>ISBN:</strong> " + [REDACTED]
rs.getString("lib_isbn") + "</div>");[REDACTED]
out.println("<div class='meta-item'><strong>Editorial:</strong> " + [REDACTED]
rs.getString("edi_nombre") + "</div>");[REDACTED]
out.println("<div class='meta-item'><strong>Año de Publicación:</strong> " + [REDACTED]
rs.getInt("lib_año_publicacion") + "</div>");[REDACTED]
out.println("<div class='meta-item'><strong>Edición:</strong> " + [REDACTED]
rs.getInt("lib_edicion") + "</div>");[REDACTED]
out.println("<div class='meta-item'><strong>Sede:</strong> " + [REDACTED]
rs.getString("sed_nombre") + "</div>");[REDACTED]
out.println("</div>");

out.println("<div class='book-description'>");[REDACTED]
out.println("<h3>Descripción</h3>");[REDACTED]
out.println("<p>" + descripcion + "</p>");[REDACTED]
out.println("</div>");

out.println("<div class='availability-section'>");[REDACTED]
out.println("<h3>Disponibilidad</h3>");[REDACTED]
out.println("<div class='availability-item'>");[REDACTED]
out.println("<div class='sede-name'>" + rs.getString("sed_nombre") + [REDACTED]
"</div>");[REDACTED]
out.println("<div class='stock-info'>");[REDACTED]

int cantidad = rs.getInt("cantidad");
if (cantidad > 0) {[REDACTED]
out.println("<span class='available'>" + cantidad + " disponibles</span>");[REDACTED]
} else {[REDACTED]
out.println("<span class='unavailable'>No disponible</span>");[REDACTED]
}[REDACTED]
```

```

out.println("</div>");

if (session.getAttribute("usuario_id") != null &&
"estudiante".equals(session.getAttribute("tipo_usuario")) && cantidad > 0) {
out.println("<a href='reservar-libro.jsp?libro_id=" + rs.getInt("cod_libro") +
'" class='btn btn-primary'>Reservar</a>");
} else if (cantidad == 0) {
out.println("<button class='btn btn-disabled' disabled>No
Disponible</button>");
} else if (session.getAttribute("usuario_id") == null) {
out.println("<a href='login.html' class='btn btn-primary'>Iniciar Sesión para
Reservar</a>");
}

out.println("</div>");
out.println("</div>");
out.println("</div>");
out.println("</div>");

out.println("<div class='separator'></div>");

out.println("<section class='related-books'>");
out.println("<h3>Otros libros del mismo autor</h3>");
out.println("<div class='books-grid'>");

String sqlRelated = "SELECT 1.*, a.au_nombre, a.au_apellido,
i.cantidad_libros, l.url_portada " +
"FROM libro l " +
"JOIN autor a ON l.cod_autor = a.cod_autor " +
"LEFT JOIN inventario i ON l.cod_libro = i.cod_libro " +
"WHERE l.cod_autor = ? AND l.cod_libro != ? AND i.cantidad_libros > 0 LIMIT
4";
PreparedStatement pstmtRelated = conn.prepareStatement(sqlRelated);
pstmtRelated.setInt(1, rs.getInt("cod_autor"));
pstmtRelated.setInt(2, Integer.parseInt(libroId));
ResultSet rsRelated = pstmtRelated.executeQuery();

while (rsRelated.next()) {
String urlPortadaRelated = rsRelated.getString("url_portada");
if (urlPortadaRelated == null || urlPortadaRelated.trim().isEmpty()) {
urlPortadaRelated = "/placeholder.svg?height=400&width=300";
}
}

out.println("<div class='book-card'>");
out.println("<img src='" + urlPortadaRelated + "' alt='"
+ rsRelated.getString("lib_nombre") + "' class='book-cover'>");
out.println("<div class='book-info'>[REDACTED]");
out.println("<h4>" + rsRelated.getString("lib_nombre") + "</h4>");
out.println("<p>" + rsRelated.getString("au_nombre") + " " +
rsRelated.getString("au_apellido") + "</p>");
out.println("<div class='book-meta'>[REDACTED]");

```

```

out.println("<span>ISBN: " + rsRelated.getString("lib_isbn") + "</span>");
out.println("<span class='stock'>" + rsRelated.getInt("cantidad_libros") + " "
disponibles</span>");■
out.println("</div>");■
out.println("<a href='detalle-libro.jsp?id=" + rsRelated.getInt("cod_libro") + 
"' class='btn btn-outline'>Ver Detalle</a>");■
out.println("</div>");■
out.println("</div>");■
}

rsRelated.close();■
stmtRelated.close();■

out.println("</div>");■
out.println("</section>");■

} else {■
out.println("<div class='no-results'><h3>Libro no encontrado</h3><a 
href='catalogo-libros.jsp' class='btn'>Volver</a></div>");■
}
} catch (Exception e) {■
out.println("<div class='no-results'><h3>Error</h3><p>" + e.getMessage() + 
"</p></div>");■
} finally {■
closeResources(conn, stmt, rs);■
}
} else {■
out.println("<div class='no-results'><h3>Libro no especificado</h3><a 
href='catalogo-libros.jsp' class='btn'>Volver</a></div>");■
}
%>
</div>■
</main>

```

```

<footer class="footer">
<div class="container">
<div class="footer-content">
<div class="footer-section">
<h3>Contacto</h3>■
<p>Email: biblioteca@universidad.edu</p>
<p>Teléfono: +507 123-4567</p>
</div>■
<div class="footer-section">
<h3>Horarios</h3>■
<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>
<p>Sábados: 8:00 AM - 5:00 PM</p>
</div>■
<div class="footer-section">
<h3>Sedes</h3>■
<p>Sede Central - Panamá</p>
<p>Sede Este - Darién</p>

```

```
<p>Sede Oeste - Panamá Oeste</p>
</div>
</div>
</div>
</footer>
</body>
</html>
```

## Devolver-libro.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<%
// Verificar autenticación
if (session.getAttribute("usuario_id") == null) {
response.sendRedirect("login.html");
return;
}
String reservaId = request.getParameter("id");
if (reservaId != null) {
Connection conn = null;
CallableStatement cstmt = null;
try {
conn = getConnection();
String sql = "{CALL devolucion_prestamo(?)}";
cstmt = conn.prepareCall(sql);
cstmt.setInt(1, Integer.parseInt(reservaId));
cstmt.execute();
String tipoUsuario = (String) session.getAttribute("tipo_usuario");
if ("bibliotecario".equals(tipoUsuario)) {
response.sendRedirect("panel-bibliotecario.jsp?success=devolucion");
} else {
response.sendRedirect("panel-usuario.jsp?success=devolucion");
}
} catch (Exception e) {
String tipoUsuario = (String) session.getAttribute("tipo_usuario");
if ("bibliotecario".equals(tipoUsuario)) {
response.sendRedirect("panel-bibliotecario.jsp?error=" + e.getMessage());
} else {
response.sendRedirect("panel-usuario.jsp?error=" + e.getMessage());
}
}
} finally {
try {
if (cstmt != null) cstmt.close();
if (conn != null) conn.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
```

```
    }
}
}
}
} else {
response.sendRedirect("panel-usuario.jsp");
}
%>
```

## Login-Process.JSP:

```

} else {
response.sendRedirect("login.html?error=contraseña_incorrecta");
}
} else {
response.sendRedirect("login.html?error=usuario_no_encontrado");
}
} else if ("bibliotecario".equals(tipoUsuario)) {
// Verificar bibliotecario por correo electrónico y contraseña
String sql = "SELECT b.id_bibliotecario, b.bibli_nombre, b.bibli_apellido,
b.bibli_contraseña, " +
"s.sed_nombre, bc.correo_bibli, bt.num_bibliotecario " +
"FROM bibliotecario b " +
"JOIN sede s ON b.cod_sede = s.cod_sede " +
"JOIN bibliotecario_correo bc ON b.id_bibliotecario = bc.id_bibliotecario " +
"LEFT JOIN bibliotecario_telefono bt ON b.id_bibliotecario =
bt.id_bibliotecario " +
"WHERE bc.correo_bibli = ?";
pst = conn.prepareStatement(sql);
pst.setString(1, email);
rs = pst.executeQuery();
if (rs.next()) {
String dbPassword = rs.getString("bibli_contraseña");
// Verificar contraseña (en producción usar hash)
if (password.equals(dbPassword)) {
// Bibliotecario autenticado - crear sesión completa
session.setAttribute("usuario_id", rs.getInt("id_bibliotecario"));
session.setAttribute("usuario_nombre", rs.getString("bibli_nombre"));
session.setAttribute("usuario_apellido", rs.getString("bibli_apellido"));
session.setAttribute("usuario_sede", rs.getString("sed_nombre"));
session.setAttribute("usuario_email", rs.getString("correo_bibli"));
session.setAttribute("usuario_telefono", rs.getString("num_bibliotecario"));
session.setAttribute("tipo_usuario", "bibliotecario");
response.sendRedirect("panel-bibliotecario.jsp?success=login");
} else {
response.sendRedirect("login.html?error=contraseña_incorrecta");
}
} else {
response.sendRedirect("login.html?error=bibliotecario_no_encontrado");
}
} else {
response.sendRedirect("login.html?error=tipo_usuario_invalido");
}
} catch (Exception e) {
e.printStackTrace();
response.sendRedirect("login.html?error=" +
java.net.URLEncoder.encode(e.getMessage(), "UTF-8"));
} finally {
closeResources(conn, pst, rs);
}
%>

```

## Logout.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%
// Invalidar la sesión
session.invalidate();
// Redirigir a la página de inicio
response.sendRedirect("index.html?success=logout");
%>
```

## Registro-Process.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Registro - Biblioteca Universitaria</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<div class="container">
<%
String nombre = request.getParameter("nombre");
String apellido = request.getParameter("apellido");
String fechaNacimiento = request.getParameter("fecha_nacimiento");
String carrera = request.getParameter("carrera");
String sede = request.getParameter("sede");
String telefono = request.getParameter("telefono");
String email = request.getParameter("email");
String password = request.getParameter("password");
Connection conn = null;
PreparedStatement pstmt = null;
try {
conn = getConnection();
// Insertar usuario con contraseña
String sql = "INSERT INTO usuario (usu_nombre, usu_apellido, usu_fecha_naci,
usu_edad, fecha_registro, cod_carrera, cod_sede, usu_contraseña) " +
"VALUES (?, ?, ?, ?, TIMESTAMPDIFF(YEAR, ?, CURDATE()), CURDATE(), ?, ?, ?)";
pstmt = conn.prepareStatement(sql, Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, nombre);
pstmt.setString(2, apellido);
pstmt.setDate(3, java.sql.Date.valueOf(fechaNacimiento));
pstmt.setDate(4, java.sql.Date.valueOf(fechaNacimiento));
pstmt.setInt(5, Integer.parseInt(carrera));
pstmt.setInt(6, Integer.parseInt(sede));
pstmt.setString(7, password);
pstmt.executeUpdate();
```

```
// Obtener el ID del usuario insertado
ResultSet generatedKeys = pstmt.getGeneratedKeys();
int usuarioId = 0;
if (generatedKeys.next()) {
    usuarioId = generatedKeys.getInt(1);
}
// Insertar teléfono
sql = "INSERT INTO usuario_telefono (usu_id, id_telefono, num_usuario) VALUES
(?, 1, ?);"
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, usuarioId);
pstmt.setLong(2, Long.parseLong(telefono));
pstmt.executeUpdate();
// Insertar correo
sql = "INSERT INTO usuario_correo (id_correo, usu_id, correo_usuario) VALUES
(6, ?, ?);"
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, usuarioId);
pstmt.setString(2, email);
pstmt.executeUpdate();
out.println("<div class='success-message'>");
out.println("<h1>¡Registro Exitoso!</h1>");
out.println("<p>Tu cuenta ha sido creada correctamente.</p>");
out.println("<p><strong>Nombre:</strong> " + nombre + " " + apellido +
"</p>"); 
out.println("<p><strong>Email:</strong> " + email + "</p>"); 
out.println("<p><strong>Teléfono:</strong> " + telefono + "</p>"); 
out.println("<p><a href='login.html?success=registro' class='btn btn-primary'>Iniciar Sesión</a></p>"); 
out.println("</div>"); 
} catch (Exception e) {
    out.println("<div class='error-message'>"); 
    out.println("<h1>Error en el Registro</h1>"); 
    out.println("<p>Ha ocurrido un error: " + e.getMessage() + "</p>"); 
    out.println("<p><a href='login.html'>Volver a intentar</a></p>"); 
    out.println("</div>"); 
    e.printStackTrace(); 
} finally {
    try {
        if (pstmt != null) pstmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
%>
</div>
</body>
</html>
```

## Renovar-Reserva.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<%
// Verificar autenticación
if (session.getAttribute("usuario_id") == null) {
response.sendRedirect("login.html");
return;
}
String reservaId = request.getParameter("id");
if (reservaId != null) {
Connection conn = null;
CallableStatement cstmt = null;
try {
conn = getConnection();
String sql = "{CALL renovar_reserva(?)}";
cstmt = conn.prepareCall(sql);
cstmt.setInt(1, Integer.parseInt(reservaId));
cstmt.execute();
String tipoUsuario = (String) session.getAttribute("tipo_usuario");
if ("bibliotecario".equals(tipoUsuario)) {
response.sendRedirect("panel-bibliotecario.jsp?success=renovacion");
} else {
response.sendRedirect("panel-usuario.jsp?success=renovacion");
}
} catch (Exception e) {
String tipoUsuario = (String) session.getAttribute("tipo_usuario");
if ("bibliotecario".equals(tipoUsuario)) {
response.sendRedirect("panel-bibliotecario.jsp?error=" + e.getMessage());
} else {
response.sendRedirect("panel-usuario.jsp?error=" + e.getMessage());
}
} finally {
try {
if (cstmt != null) cstmt.close();
if (conn != null) conn.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
} else {
response.sendRedirect("panel-usuario.jsp");
}
%>
```

## Reservar-Libro.JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="java.sql.*" %>
<%@ include file="config/conexion.jsp" %>
<%
// Verificar autenticación
if (session.getAttribute("usuario_id") == null || !"estudiante".equals(session.getAttribute("tipo_usuario")))) {
response.sendRedirect("login.html");
return;
}
int usuarioId = (Integer) session.getAttribute("usuario_id");
String libroIdParam = request.getParameter("libro_id");
if (libroIdParam != null && request.getMethod().equals("POST")) {
// Procesar reserva
Connection conn = null;
CallableStatement cstmt = null;
try {
conn = getConnection();
String sql = "{CALL carga_reservar(?, ?)}";
cstmt = conn.prepareCall(sql);
cstmt.setInt(1, Integer.parseInt(libroIdParam));
cstmt.setInt(2, usuarioId);
cstmt.execute();
response.sendRedirect("panel-usuario.jsp?success=reserva");
return;
} catch (Exception e) {
request.setAttribute("error", "Error al realizar la reserva: " +
e.getMessage());
} finally {
try {
if (cstmt != null) cstmt.close();
if (conn != null) conn.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
}
%>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Reservar Libro - Biblioteca Universitaria</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header class="header">
<div class="container">
<div class="Logo">


```

```

<h1>Biblioteca Universitaria</h1>
</div>
<nav class="nav">
<a href="catalogo-libros.jsp" class="nav-link">Catálogo</a>
<a href="panel-usuario.jsp" class="nav-link">Mi Panel</a>
<a href="Logout.jsp" class="nav-link">Cerrar Sesión</a>
</nav>
</div>
</header>

<main class="main">
<div class="container">
<div class="reserva-container">
<h2>Reservar Libro</h2>
<%
if (request.getAttribute("error") != null) {
out.println("<div class='error-message'>" + request.getAttribute("error") +
"</div>");
}
%>
<div class="reserva-content">
<%
String libroId = request.getParameter("libro_id");
if (libroId != null) {
Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rs = null;
try {
conn = getConnection();
String sql = "SELECT * FROM v_librosdisponibles WHERE cod_libro = ?";
pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, Integer.parseInt(libroId));
rs = pstmt.executeQuery();
if (rs.next()) {
out.println("<div class='libro-selected'>");
out.println("<div class='libro-image'>");■
out.println("<img src='/placeholder.svg?height=200&width=150' alt=' " +
rs.getString("titulo") + "'>");■
out.println("</div>");■
out.println("<div class='libro-details'>");■
out.println("<h3>" + rs.getString("titulo") + "</h3>");■
out.println("<p><strong>Autor:</strong> " + rs.getString("autor") + "</p>");■
out.println("<p><strong>ISBN:</strong> " + rs.getString("ISBN") + "</p>");■
out.println("<p><strong>Editorial:</strong> " + rs.getString("editorial") + "</p>");■
out.println("<p><strong>Año:</strong> " + rs.getInt("ano_publicacion") + "</p>");■
out.println("<p><strong>Edición:</strong> " + rs.getInt("edicion") + "</p>");■
out.println("<p><strong>Disponibles:</strong> <span class='stock'>" + rs.getInt("cantidad") + " ejemplares</span></p>");■
out.println("</div>");■
out.println("</div>");■
}
}
}
%>

```

```
// Formulario de reserva
out.println("<form method='POST' class='reserva-form'>");
out.println("<input type='hidden' name='libro_id' value='" + libroId + "'>");[REDACTED]
out.println("<div class='form-section'>");[REDACTED]
out.println("<h4>Información de la Reserva</h4>");[REDACTED]
out.println("<div class='form-group'>");[REDACTED]
out.println("<label>Fecha de Préstamo:</label>");[REDACTED]
out.println("<input type='date' value='" + java.time.LocalDate.now() + "'[REDACTED]
readonly>");[REDACTED]
out.println("</div>");[REDACTED]
out.println("<div class='form-group'>");[REDACTED]
out.println("<label>Fecha de Devolución:</label>");[REDACTED]
out.println("<input type='date' value='" + [REDACTED]
java.time.LocalDate.now().plusDays(7) + "' readonly>");[REDACTED]
out.println("</div>");[REDACTED]
out.println("<div class='form-section'>");[REDACTED]
out.println("<h4>Términos y Condiciones</h4>");[REDACTED]
out.println("<div class='terms-box'>");[REDACTED]
out.println("<ul>");[REDACTED]
out.println("<li>El período de préstamo es de 7 días calendario.</li>");[REDACTED]
out.println("<li>Puede renovar el préstamo una vez por 7 días [REDACTED]
 adicionales.</li>");[REDACTED]
out.println("<li>Máximo 5 libros reservados simultáneamente.</li>");[REDACTED]
out.println("<li>La devolución tardía puede generar restricciones.</li>");[REDACTED]
out.println("<li>Debe presentar su identificación al retirar el libro.</li>");[REDACTED]
out.println("</ul>");[REDACTED]
out.println("</div>");[REDACTED]
out.println("<div class='checkbox-group'>");[REDACTED]
out.println("<input type='checkbox' id='accept-terms' required>");[REDACTED]
out.println("<label for='accept-terms'>Acepto los términos y [REDACTED]
 condiciones</label>");[REDACTED]
out.println("</div>");[REDACTED]
out.println("<div class='form-actions'>");[REDACTED]
out.println("<a href='catalogo-libros.jsp' class='btn btn-[REDACTED]
 secondary'>Cancelar</a>");[REDACTED]
out.println("<button type='submit' class='btn btn-primary'>Confirmar [REDACTED]
 Reserva</button>");[REDACTED]
out.println("</div>");[REDACTED]
out.println("</form>");[REDACTED]
} else {[REDACTED]
out.println("<p>Libro no encontrado.</p>");[REDACTED]
}
} catch (Exception e) {[REDACTED]
out.println("<p>Error al cargar información del libro: " + e.getMessage() + [REDACTED]
 "</p>");[REDACTED]
} finally {[REDACTED]
closeResources(conn, pstmt, rs);[REDACTED]
}
} else {[REDACTED]
out.println("<p>No se especificó un libro para reservar.</p>");[REDACTED]
}
```

```

out.println("<a href='catalogo-libros.jsp' class='btn btn-primary'>Volver al  

Catálogo</a>");  

}  

%>  

</div>  

</div>  

</div>  

</main>

<footer class="footer">  

<div class="container">  

<div class="footer-content">  

<div class="footer-section">  

<h3>Contacto</h3>  

<p>Email: biblioteca@universidad.edu</p>  

<p>Teléfono: +507 123-4567</p>  

</div>  

<div class="footer-section">  

<h3>Horarios</h3>  

<p>Lunes a Viernes: 7:00 AM - 9:00 PM</p>  

<p>Sábados: 8:00 AM - 5:00 PM</p>  

</div>  

<div class="footer-section">  

<h3>Sedes</h3>  

<p>Sede Central - Panamá</p>  

<p>Sede Este - Darién</p>  

<p>Sede Oeste - Panamá Oeste</p>  

</div>  

</div>  

</div>
</body>
</html>
```

## Estilos

```

/* Reset y configuración base */  

* {  

margin: 0;  

padding: 0;  

box-sizing: border-box;  

}  
  

body {  

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;  

line-height: 1.6;  

color: #333;  

background-color: #f8f9fa;  

}
```

```
.container {  
    max-width: 1200px;  
    margin: 0 auto;  
    padding: 0 20px;  
}  
  
/* Header */  
.header {  
    background: linear-gradient(135deg, #1e3a8a 0%, #3b82f6 100%);  
    color: white;  
    padding: 1rem 0;  
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}  
  
.header .container {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}  
  
.logo {  
    display: flex;  
    align-items: center;  
    gap: 1rem;  
}  
  
.logo img {  
    width: 50px;  
    height: 50px;  
    border-radius: 8px;  
}  
  
.logo h1 {  
    font-size: 1.5rem;  
    font-weight: 600;  
}  
  
.nav {  
    display: flex;  
    gap: 2rem;  
}  
  
.nav-link {  
    color: white;
```

```
text-decoration: none;
padding: 0.5rem 1rem;
border-radius: 6px;
transition: all 0.3s ease;
font-weight: 500;
}

.nav-link:hover,
.nav-link.active {
  background-color: rgba(255, 255, 255, 0.2);
  transform: translateY(-1px);
}

/* Main Content */
.main {
  min-height: calc(100vh - 200px);
  padding: 2rem 0;
}

/* Hero Section */
.hero {
  text-align: center;
  margin-bottom: 3rem;
  padding: 2rem 0;
  background: white;
  border-radius: 12px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}

.hero h2 {
  font-size: 2.5rem;
  color: #1e3a8a;
  margin-bottom: 1rem;
  font-weight: 700;
}

.hero p {
  font-size: 1.2rem;
  color: #6b7280;
}

/* Filters */
.filters {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
```

```
gap: 1.5rem;
margin-bottom: 3rem;
padding: 2rem;
background: white;
border-radius: 12px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}

.filter-group {
  display: flex;
  flex-direction: column;
  gap: 0.5rem;
}

.filter-group label {
  font-weight: 600;
  color: #374151;
}

.filter-select,
.search-input {
  padding: 0.75rem;
  border: 2px solid #e5e7eb;
  border-radius: 8px;
  font-size: 1rem;
  transition: border-color 0.3s ease;
}

.filter-select:focus,
.search-input:focus {
  outline: none;
  border-color: #3b82f6;
  box-shadow: 0 0 3px rgba(59, 130, 246, 0.1);
}

.search-group {
  display: flex;
  gap: 0.5rem;
}

.search-btn {
  padding: 0.75rem 1.5rem;
  background: #3b82f6;
  color: white;
  border: none;
```

```
border-radius: 8px;
cursor: pointer;
font-weight: 600;
transition: all 0.3s ease;
}

.search-btn:hover {
background: #2563eb;
transform: translateY(-1px);
}

/* Books Grid */
.books-grid {
display: grid;
grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
gap: 2rem;
margin-bottom: 3rem;
}

.book-card {
background: white;
border-radius: 12px;
overflow: hidden;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
transition: all 0.3s ease;
}

.book-card:hover {
transform: translateY(-4px);
box-shadow: 0 8px 25px rgba(0, 0, 0, 0.1);
}

.book-card img {
width: 100%;
height: 200px;
object-fit: cover;
}

.book-info {
padding: 1.5rem;
}

.book-info h4 {
font-size: 1.2rem;
color: #1e3a8a;
```

```
margin-bottom: 0.5rem;  
font-weight: 600;  
}
```

```
.book-info p {  
color: #6b7280;  
margin-bottom: 1rem;  
}
```

```
.book-meta {  
display: flex;  
flex-direction: column;  
gap: 0.25rem;  
margin-bottom: 1rem;  
font-size: 0.9rem;  
}
```

```
.stock {  
color: #059669;  
font-weight: 600;  
}
```

```
/* Buttons */  
.btn {  
padding: 0.75rem 1.5rem;  
border: none;  
border-radius: 8px;  
cursor: pointer;  
font-weight: 600;  
text-decoration: none;  
display: inline-block;  
text-align: center;  
transition: all 0.3s ease;  
font-size: 1rem;  
}
```

```
.btn-primary {  
background: #3b82f6;  
color: white;  
}
```

```
.btn-primary:hover {  
background: #2563eb;  
transform: translateY(-1px);  
}
```

```
.btn-secondary {  
background: #6b7280;  
color: white;  
}  
  
.btn-secondary:hover {  
background: #4b5563;  
}  
  
.btn-outline {  
background: transparent;  
color: #3b82f6;  
border: 2px solid #3b82f6;  
}  
  
.btn-outline:hover {  
background: #3b82f6;  
color: white;  
}  
  
.btn-small {  
padding: 0.5rem 1rem;  
font-size: 0.9rem;  
}  
  
.btn-success {  
background: #059669;  
color: white;  
}  
  
.btn-success:hover {  
background: #047857;  
}  
  
.btn-danger {  
background: #dc2626;  
color: white;  
}  
  
.btn-danger:hover {  
background: #b91c1c;  
}  
  
.btn-disabled {
```

```
background: #d1d5db;
color: #9ca3af;
cursor: not-allowed;
}

/* Auth Forms */
.auth-container {
max-width: 500px;
margin: 0 auto;
background: white;
border-radius: 12px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
overflow: hidden;
}

.auth-tabs {
display: flex;
background: #f3f4f6;
}

.tab-btn {
flex: 1;
padding: 1rem;
background: transparent;
border: none;
cursor: pointer;
font-weight: 600;
transition: all 0.3s ease;
}

.tab-btn.active {
background: white;
color: #3b82f6;
}

.auth-form {
padding: 2rem;
}

.auth-form.hidden {
display: none;
}

.auth-form h2 {
color: #1e3a8a;
```

```
margin-bottom: 2rem;
text-align: center;
}

.form-group {
margin-bottom: 1.5rem;
}

.form-row {
display: grid;
grid-template-columns: 1fr 1fr;
gap: 1rem;
}

.form-group label {
display: block;
margin-bottom: 0.5rem;
font-weight: 600;
color: #374151;
}

.form-group input,
.form-group select,
.form-group textarea {
width: 100%;
padding: 0.75rem;
border: 2px solid #e5e7eb;
border-radius: 8px;
font-size: 1rem;
transition: border-color 0.3s ease;
}

.form-group input:focus,
.form-group select:focus,
.form-group textarea:focus {
outline: none;
border-color: #3b82f6;
box-shadow: 0 0 0 3px rgba(59, 130, 246, 0.1);
}

/* Panel Styles */
.panel-header {
display: flex;
justify-content: space-between;
align-items: center;
```

```
margin-bottom: 2rem;
padding: 2rem;
background: white;
border-radius: 12px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}

.panel-header h2 {
color: #1e3a8a;
font-size: 2rem;
}

.user-info {
display: flex;
align-items: center;
gap: 1rem;
}

.panel-grid {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(400px, 1fr));
gap: 2rem;
}

.panel-card {
background: white;
border-radius: 12px;
padding: 2rem;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}

.panel-card.full-width {
grid-column: 1 / -1;
}

.panel-card h3 {
color: #1e3a8a;
margin-bottom: 1.5rem;
font-size: 1.3rem;
}

.user-details {
display: flex;
flex-direction: column;
gap: 1rem;
```

```
margin-bottom: 2rem;
}

.detail-item {
padding: 0.75rem;
background: #f8f9fa;
border-radius: 8px;
border-left: 4px solid #3b82f6;
}

/* Reservas */
.reservas-list {
display: flex;
flex-direction: column;
gap: 1.5rem;
}

.reserva-item {
display: grid;
grid-template-columns: 1fr auto auto;
gap: 1rem;
padding: 1.5rem;
background: #f8f9fa;
border-radius: 8px;
border-left: 4px solid #3b82f6;
}

.libro-info h4 {
color: #1e3a8a;
margin-bottom: 0.5rem;
}

.libro-info p {
color: #6b7280;
margin-bottom: 0.25rem;
}

.isbn {
font-size: 0.9rem;
color: #9ca3af;
}

.reserva-dates {
text-align: center;
}
```

```
.reserva-dates p {  
    margin-bottom: 0.25rem;  
    font-size: 0.9rem;  
}  
  
.reserva-actions {  
    display: flex;  
    flex-direction: column;  
    gap: 0.5rem;  
}  
  
/* Status */  
.status {  
    padding: 0.25rem 0.75rem;  
    border-radius: 20px;  
    font-size: 0.8rem;  
    font-weight: 600;  
    text-align: center;  
}  
  
.status-active {  
    background: #dcfce7;  
    color: #166534;  
}  
  
.status-completed {  
    background: #e0e7ff;  
    color: #3730a3;  
}  
  
.status-pending {  
    background: #fef3c7;  
    color: #92400e;  
}  
  
.status-warning {  
    background: #fed7d7;  
    color: #c53030;  
}  
  
.status-good {  
    background: #dcfce7;  
    color: #166534;  
}
```

```
/* Tables */
.table-container {
  overflow-x: auto;
}

.history-table,
.admin-table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 1rem;
}

.history-table th,
.history-table td,
.admin-table th,
.admin-table td {
  padding: 1rem;
  text-align: left;
  border-bottom: 1px solid #e5e7eb;
}

.history-table th,
.admin-table th {
  background: #f8f9fa;
  font-weight: 600;
  color: #374151;
}

.history-table tr:hover,
.admin-table tr:hover {
  background: #f8f9fa;
}

/* Admin Tabs */
.admin-tabs {
  display: flex;
  gap: 0.5rem;
  margin-bottom: 2rem;
  background: white;
  padding: 1rem;
  border-radius: 12px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}
```

```
.admin-tabs .tab-btn {  
    padding: 0.75rem 1.5rem;  
    background: #f3f4f6;  
    color: #6b7280;  
    border: none;  
    border-radius: 8px;  
    cursor: pointer;  
    font-weight: 600;  
    transition: all 0.3s ease;  
}  
  
.admin-tabs .tab-btn.active {  
    background: #3b82f6;  
    color: white;  
}  
  
.tab-content {  
    display: none;  
}  
  
.tab-content.active {  
    display: block;  
}  
  
.section-header {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    margin-bottom: 1.5rem;  
}  
  
.search-section {  
    display: flex;  
    gap: 1rem;  
    margin-bottom: 1.5rem;  
}  
  
/* Inventory */  
.inventory-filters {  
    margin-bottom: 2rem;  
}  
  
.inventory-grid {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
}
```

```
gap: 1.5rem;
margin-bottom: 2rem;
}

.inventory-card {
background: #f8f9fa;
padding: 1.5rem;
border-radius: 8px;
text-align: center;
border-left: 4px solid #3b82f6;
}

.inventory-card h4 {
color: #6b7280;
margin-bottom: 0.5rem;
font-size: 0.9rem;
}

.inventory-number {
font-size: 2rem;
font-weight: 700;
color: #1e3a8a;
}

.inventory-number.warning {
color: #dc2626;
}

/* Reports */
.reports-grid {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 2rem;
}

.report-card {
background: #f8f9fa;
padding: 1.5rem;
border-radius: 8px;
border-left: 4px solid #3b82f6;
}

.report-card h4 {
color: #1e3a8a;
margin-bottom: 1rem;
```

```
}
```

.report-list {  
 list-style: none;  
 counter-reset: item;  
}

.report-list li {  
 counter-increment: item;  
 margin-bottom: 0.5rem;  
 padding-left: 2rem;  
 position: relative;  
}

.report-list li::before {  
 content: counter(item);  
 position: absolute;  
 left: 0;  
 top: 0;  
 background: #3b82f6;  
 color: white;  
 width: 1.5rem;  
 height: 1.5rem;  
 border-radius: 50%;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 0.8rem;  
 font-weight: 600;  
}

.stats-list {  
 display: flex;  
 flex-direction: column;  
 gap: 0.75rem;  
}

.stat-item {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 0.5rem 0;  
 border-bottom: 1px solid #e5e7eb;  
}

```
/* Reserva Form */
.reserva-container {
    max-width: 800px;
    margin: 0 auto;
}

.reserva-container h2 {
    color: #1e3a8a;
    margin-bottom: 2rem;
    text-align: center;
    font-size: 2rem;
}

.reserva-content {
    background: white;
    border-radius: 12px;
    padding: 2rem;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
}

.libro-selected {
    display: grid;
    grid-template-columns: auto 1fr;
    gap: 2rem;
    margin-bottom: 3rem;
    padding: 2rem;
    background: #f8f9fa;
    border-radius: 8px;
}

.libro-image img {
    width: 150px;
    height: 200px;
    object-fit: cover;
    border-radius: 8px;
}

.libro-details h3 {
    color: #1e3a8a;
    margin-bottom: 1rem;
    font-size: 1.5rem;
}

.libro-details p {
    margin-bottom: 0.5rem;
```

```
color: #374151;
}

.reserva-form {
  display: flex;
  flex-direction: column;
  gap: 2rem;
}

.form-section {
  padding: 1.5rem;
  border: 1px solid #e5e7eb;
  border-radius: 8px;
}

.form-section h4 {
  color: #1e3a8a;
  margin-bottom: 1.5rem;
  font-size: 1.2rem;
}

.terms-box {
  background: #f8f9fa;
  padding: 1.5rem;
  border-radius: 8px;
  margin-bottom: 1rem;
}

.terms-box ul {
  list-style-type: disc;
  padding-left: 1.5rem;
}

.terms-box li {
  margin-bottom: 0.5rem;
  color: #374151;
}

.checkbox-group {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}

.checkbox-group input[type="checkbox"] {
```

```
width: auto;  
}
```

```
.form-actions {  
display: flex;  
gap: 1rem;  
justify-content: flex-end;  
padding-top: 1rem;  
border-top: 1px solid #e5e7eb;  
}
```

```
/* Book Detail */
```

```
.breadcrumb {  
margin-bottom: 2rem;  
color: #6b7280;  
}
```

```
.breadcrumb a {  
color: #3b82f6;  
text-decoration: none;  
}
```

```
.breadcrumb a:hover {  
text-decoration: underline;  
}
```

```
.book-detail {  
display: grid;  
grid-template-columns: auto 1fr;  
gap: 3rem;  
margin-bottom: 3rem;  
background: white;  
padding: 2rem;  
border-radius: 12px;  
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);  
}
```

```
.book-cover {  
width: 300px;  
height: 400px;  
object-fit: cover;  
border-radius: 8px;  
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}
```

```
.book-info-section h1 {  
    color: #1e3a8a;  
    font-size: 2.5rem;  
    margin-bottom: 0.5rem;  
}  
  
.book-info-section h2 {  
    color: #6b7280;  
    font-size: 1.5rem;  
    margin-bottom: 2rem;  
    font-weight: 400;  
}  
  
.book-meta {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
    gap: 1rem;  
    margin-bottom: 2rem;  
}  
  
.meta-item {  
    padding: 0.75rem;  
    background: #f8f9fa;  
    border-radius: 8px;  
}  
  
.book-description {  
    margin-bottom: 2rem;  
}  
  
.book-description h3 {  
    color: #1e3a8a;  
    margin-bottom: 1rem;  
}  
  
.book-description p {  
    margin-bottom: 1rem;  
    line-height: 1.7;  
    color: #374151;  
}  
  
.availability-section h3 {  
    color: #1e3a8a;  
    margin-bottom: 1.5rem;  
}
```

```
.availability-grid {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 1rem;  
}  
  
.availability-item {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 1rem;  
    background: #f8f9fa;  
    border-radius: 8px;  
    border-left: 4px solid #3b82f6;  
}  
  
.sede-name {  
    font-weight: 600;  
    color: #1e3a8a;  
}  
  
.stock-info {  
    font-size: 0.9rem;  
    color: #6b7280;  
}  
  
.available {  
    color: #059669;  
    font-weight: 600;  
}  
  
.unavailable {  
    color: #dc2626;  
    font-weight: 600;  
}  
  
.related-books {  
    margin-top: 3rem;  
}  
  
.related-books h3 {  
    color: #1e3a8a;  
    margin-bottom: 2rem;  
    font-size: 1.5rem;
```

```
}
```

```
/* Footer */
```

```
.footer {
```

```
background: #1f2937;
```

```
color: white;
```

```
padding: 3rem 0 1rem;
```

```
margin-top: 4rem;
```

```
}
```

```
.footer-content {
```

```
display: grid;
```

```
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
```

```
gap: 2rem;
```

```
margin-bottom: 2rem;
```

```
}
```

```
.footer-section h3 {
```

```
margin-bottom: 1rem;
```

```
color: #3b82f6;
```

```
}
```

```
.footer-section p {
```

```
margin-bottom: 0.5rem;
```

```
color: #d1d5db;
```

```
}
```

```
/* Responsive Design */
```

```
@media (max-width: 768px) {
```

```
.container {
```

```
padding: 0 15px;
```

```
}
```

```
.header .container {
```

```
flex-direction: column;
```

```
gap: 1rem;
```

```
}
```

```
.nav {
```

```
gap: 1rem;
```

```
}
```

```
.hero h2 {
```

```
font-size: 2rem;
```

```
}
```

```
.filters {  
    grid-template-columns: 1fr;  
}  
  
.books-grid {  
    grid-template-columns: 1fr;  
}  
  
.form-row {  
    grid-template-columns: 1fr;  
}  
  
.panel-grid {  
    grid-template-columns: 1fr;  
}  
  
.book-detail {  
    grid-template-columns: 1fr;  
    text-align: center;  
}  
  
.libro-selected {  
    grid-template-columns: 1fr;  
    text-align: center;  
}  
  
.reserva-item {  
    grid-template-columns: 1fr;  
    text-align: center;  
}  
  
.availability-item {  
    flex-direction: column;  
    gap: 1rem;  
    text-align: center;  
}  
  
.form-actions {  
    flex-direction: column;  
}  
  
.admin-tabs {  
    flex-wrap: wrap;  
}
```

```
.search-section {  
    flex-direction: column;  
}  
}  
  
@media (max-width: 480px) {  
    .hero h2 {  
        font-size: 1.5rem;  
    }  
  
    .panel-header {  
        flex-direction: column;  
        gap: 1rem;  
        text-align: center;  
    }  
  
    .user-info {  
        flex-direction: column;  
    }  
  
    .book-info-section h1 {  
        font-size: 2rem;  
    }  
  
    .book-info-section h2 {  
        font-size: 1.2rem;  
    }  
}
```

## Retos y dificultades

RETO	Descripción de la solución
1. Conectar la base de datos con el HTML	Retomando trabajos anteriores, pudimos alcanzar un resultado parecido y adaptarlo de manera efectiva al desarrollo de este proyecto.
2. Crear la base de datos	Adaptación entre lenguajes.

## Conclusión

En conclusión, la implementación de este sistema, que combina una base de datos eficiente con una interfaz en HTML, representa un avance significativo en la gestión bibliotecaria de la universidad. La base de datos permite centralizar y organizar la información clave como libros, usuarios y préstamos, mientras que la interfaz en HTML facilita el acceso y la interacción con el sistema de forma más intuitiva y accesible para los usuarios. Esta integración no solo optimiza los procesos internos, sino que también mejora la experiencia general del usuario, apoyando el aprendizaje, la investigación y la toma de decisiones dentro de la comunidad universitaria.

## Referencias bibliográficas

*W3Schools.com.* (s. f.). <https://www.w3schools.com/html/>

*W3Schools.com.* (s. f.-b). [https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)

Material didáctico

