



PROYECTO BIBLIOTECH

ÍNDICE

1. *¿Qué es y cómo funciona Bibliotech?*
2. *Ficheros importantes de la aplicación:*
 - a. *Funciones.php*
 - b. *Index.php*
 - c. *Registrar.inc.php*
 - d. *Login.inc.php*
 - e. *Menu.inc.php*
3. *Resto de ficheros de la aplicación:*
 - a. *Buscador.inc.php*
 - b. *Crearlibros.inc.php*
 - c. *Editarlibro.inc.php*
 - d. *Eliminarlibro.inc.php*
 - e. *Restaurarlibro.inc.php*
 - f. *Prestamos.inc.php*
4. *Fichero CSS*
5. *Fichero BBDD*
6. *Conclusión y Justificación*
7. *Biografía*

1- ¿Qué es y cómo funciona Bibliotech?

Bibliotech es una aplicación creada con los lenguajes de programación de PHP, HTML, CSS y JavaScript la cual tiene su login y registro para poder acceder a ella. Dentro de la aplicación tenemos muchas pestañas con las que podemos interactuar y realizar muchas operaciones. Es una aplicación sencilla, básica y fácil de moverse por la aplicación web.

A continuación, una pequeña vista de lo que tiene Bibliotech en el entorno cliente:

The image displays three screenshots of the Bibliotech web application interface, arranged vertically. Each screenshot has a blue header bar with the text "Bienvenido/a a la Aplicación de Bibliotech:" and a blue footer bar with the text "© 2023 IES VillaVerde - Creado por Tomas Cano 2ºDAW".

Top Screenshot: Login Page
The login page features a central form titled "Iniciar Sesión". It includes two input fields: "Escribe tu usuario" (labeled "Usuario") and "Escribe una contraseña" (labeled "Contraseña"). Below these fields are two green buttons: "Enviar" and "Registrarse".

Middle Screenshot: Registration Page
The registration page features a central form titled "Registro". It includes several input fields: "Nombre", "Apellido Paterno", "Apellido Materno", "Correo Electrónico", "Usuario", and "Contraseña". Below these fields is a dropdown menu labeled "Alumno" and two green buttons: "Registrarse" and "Iniciar Sesión".

Bottom Screenshot: User Dashboard
The user dashboard is titled "Bienvenido/a a la Aplicación de Bibliotech: tomas123". It features a dark blue sidebar on the left with a "Menu User--" section containing links for "Perfil", "Buscador", "Libros", "Prestamos", and "Contacto". A red "Cerrar Sesión" button is located at the bottom of the sidebar. The main content area on the right contains a welcome message, a section titled "¿Qué ofrecemos en nuestra Pagina Web?" with three paragraphs of text, and a large circular logo on the right side. The logo features a blue silhouette of a head with a pair of headphones inside, and the text "BIBLIOTECH THE LIBRARY OF WONDERS" below it.

2- Ficheros importantes de la aplicación

Funciones.php:

Function HashPassword(\$password) {}

→ Esta función en PHP, llamada 'hashPassword', utiliza la función 'password_hash' que de manera resumida, devuelve un hash seguro de la contraseña proporcionada.

Function verificarInicioSesion(\$usuario, \$password) {}

→ Preparación de la consulta SQL: La función prepara una consulta SQL para seleccionar información de usuario (id, usuario, contraseña, rol) basada en el nombre de usuario proporcionado.

→ Ejecución de la consulta: La función ejecuta la consulta, vincula los resultados a variables y recupera la información del usuario desde la base de datos.

→ Verificación de la contraseña: Utiliza la función 'password_verify' para verificar si la contraseña proporcionada coincide con el hash almacenado en la base de datos.

→ Retorno de resultados: Si la verificación de la contraseña es exitosa, la función devuelve un arreglo asociativo con la información del usuario. En caso contrario, devuelve false.

→ Cierre de la declaración preparada: La función cierra la declaración preparada después de su ejecución.

Function registrarUsuario(\$nombre, \$apellido1, \$apellido2, \$correo, \$usuario, \$contrasena, \$rol) {}

→ Hashing de la contraseña: La función utiliza la función hashPassword para generar un hash seguro de la contraseña antes de almacenarla en la base de datos.

→ Preparación de la consulta SQL: La función prepara una consulta SQL para insertar nuevos registros en la tabla de usuarios con la información proporcionada.

→ Vinculación de parámetros y ejecución: La función vincula los parámetros de la consulta a las variables proporcionadas como argumentos, ejecuta la consulta y devuelve el resultado de la ejecución.

→ Retorno del resultado de la ejecución: La función devuelve true si la inserción en la base de datos fue exitosa, y false en caso contrario.

→ Cierre de la declaración preparada: Es importante señalar que el cierre de la declaración preparada (\$stmt->close()) nunca se alcanzará debido al return que precede a esa línea. El código después del return no se ejecutará.

HashPassword:

```
function hashPassword($password) {  
    return password_hash($password, PASSWORD_DEFAULT);  
}
```

VerificarInicioSesion:

```
function verificarInicioSesion($usuario, $password) {  
    global $conn;  
  
    $stmt = $conn->prepare("SELECT id, usuario, contrasena, rol FROM usuarios WHERE usuario = ?");  
    $stmt->bind_param("s", $usuario);  
    $stmt->execute();  
    $stmt->bind_result($dbId, $dbUsuario, $dbContrasena, $rol);  
  
    if ($stmt->fetch() && password_verify($password, $dbContrasena)) {  
        $userData = [];  
        'id' => $dbId,  
        'usuario' => $dbUsuario,  
        'contrasena' => $dbContrasena,  
        'rol' => $rol  
    ];  
    return $userData;  
    } else {  
        return false;  
    }  
  
    $stmt->close();  
}
```

RegistrarUsuario:

```
function registrarUsuario($nombre, $apellido1, $apellido2, $correo, $usuario, $contrasena, $rol) {  
    global $conn;  
  
    $contrasena = hashPassword($contrasena);  
  
    $stmt = $conn->prepare("INSERT INTO usuarios (nombre, apellido1, apellido2, correo, usuario, contrasena, rol) VALUES (?, ?, ?, ?, ?, ?, ?)");  
    $stmt->bind_param("sssssss", $nombre, $apellido1, $apellido2, $correo, $usuario, $contrasena, $rol);  
  
    return $stmt->execute();  
  
    $stmt->close();  
}
```

Index.php:

Este código PHP se encarga de manejar la lógica del inicio de sesión, registro de usuarios y enrutamiento de páginas web:

- Manejo de sesiones: Inicia la sesión y verifica si el formulario de inicio de sesión ha sido enviado (`isset($_POST["enviarLogin"])`) para autenticar al usuario utilizando la función `verificarInicioSesion`.
- Registro de usuarios: Si se ha enviado el formulario de registro (`isset($_POST["registrarse"])`), recopila los datos del formulario y utiliza la función `registrarUsuario` para insertar un nuevo usuario en la base de datos.
- Redirección después de inicio de sesión: Si la autenticación es exitosa, se establecen variables de sesión y se redirige al usuario a la página principal.
- Manejo de roles: Verifica el rol del usuario y redirige si es necesario (por ejemplo, si un alumno intenta acceder a la ruta de administrador).
- Páginas HTML: La parte final del código representa la estructura HTML de la página web, incluyendo la inclusión de diferentes fragmentos de código dependiendo de la ruta y la sesión del usuario.

Parte de Login y Registro de index.php:

```
1 <?php
2 require_once("funciones.php"); // Incluye las funciones necesarias
3 session_start();
4
5 if (isset($_POST["enviarLogin"])) {
6     $usuario = $_POST["usuario"];
7     $password = $_POST["password"];
8
9     $rol = verificarInicioSesion($usuario, $password);
10
11     if ($rol) {
12         $_SESSION["usuario"] = $rol["usuario"];
13         $_SESSION["rol"] = $rol["rol"];
14         $_SESSION["id"] = $rol["id"];
15
16         header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=principal");
17         exit;
18     } else {
19         $errValido = true;
20     }
21 }
22
23 if (isset($_POST["registrarse"])) {
24     $nombre = $_POST["nombre"];
25     $apellido1 = $_POST["apellido1"];
26     $apellido2 = $_POST["apellido2"];
27     $correo = $_POST["correo"];
28     $usuario = $_POST["usuario"];
29     $contrasena = $_POST["contrasena"];
30     $rol = $_POST["rol"];
31
32     if (registrarUsuario($nombre, $apellido1, $apellido2, $correo, $usuario, $contrasena, $rol)) {
33         echo '<script>alert("Los datos se han insertado correctamente.");</script>';
34     }
35 }
36
37 if (isset($_SESSION["usuario"]) && isset($_SESSION["rol"])) {
38     $rol = $_SESSION["rol"];
39     if ($rol === "alumno" && isset($_GET["ruta"]) && $_GET["ruta"] === "admin") {
40         // Redirige a una página de error o a la página principal si el usuario intenta acceder a la ruta de administrador
41         header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=principal");
42         exit;
43     }
44 }
45
46 ?>
```

Parte de Redirecciones de la aplicación:

```
49 <!DOCTYPE html>
50 <html lang="es">
51 <head>
52     <meta charset="UTF-8">
53     <meta http-equiv="X-UA-Compatible" content="IE=edge">
54     <title>Bibliotech</title>
55     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
56     <link rel="stylesheet" href="css/estilos.css">
57     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
58 </head>
59 <body>
60 <?php include_once "estructuraWeb/cabezera.inc.php"; ?>
61 <div class="container">
62
63
64 <?php include_once "estructuraWeb/menu.inc.php"; ?>
65 <?php if(empty($_GET["ruta"]) || !isset($_SESSION["usuario"])) {
66     if (isset($_POST["registrar"])) {
67         include_once "estructuraWeb/registrar.inc.php";
68     } else {
69         include_once "estructuraWeb/login.inc.php";
70     }
71 }
72 else if(isset($_SESSION["usuario"])) {
73     if (isset($_GET["ruta"]) && ($_GET["ruta"] == "principal")) {
74         include_once "estructuraWeb/principal.inc.php";
75     }
76     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "perfil")) {
77         include_once "estructuraWeb/perfil.inc.php";
78     }
79     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "logout")) {
80         include_once "estructuraWeb/logout.inc.php";
81     }
82     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "contacto")) {
83         include_once "estructuraWeb/contacto.inc.php";
84     }
85     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "libros")) {
86         include_once "estructuraWeb/libros.inc.php";
87     }
88     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "prestamos")) {
89         include_once "estructuraWeb/prestamos.inc.php";
90     }
91     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "usuarios")) {
92         include_once "estructuraWeb/usuarios.inc.php";
93     }
94     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "crearusuarios")) {
95         include_once "estructuraWeb/crearusuarios.inc.php";
96     }
97     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "editarlibros")) {
98         include_once "estructuraWeb/editarlibros.inc.php";
99     }
100     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "crearlibros")) {
101         include_once "estructuraWeb/crearlibros.inc.php";
102     }
103     else if (isset($_GET["ruta"]) && ($_GET["ruta"] == "editarprestamo")) {
104         include_once "estructuraWeb/editarprestamo.inc.php";
105     }
106 }
```

Registrar.inc.php:

Este código PHP maneja el registro de usuarios a través de un formulario HTML. Aquí tienes una descripción resumida:

→ **Registro de usuarios:** Verifica si el formulario de registro ha sido enviado (`isset($_POST["registrarse"])`). Si es así, recopila los datos del formulario y utiliza la función `registrarUsuario` para insertar un nuevo usuario en la base de datos.

→ **Alerta de éxito:** Si el registro es exitoso, muestra una alerta JavaScript indicando que los datos se han insertado correctamente.

→ **Formulario HTML de registro:** Muestra un formulario HTML que recopila información como nombre, apellidos, correo electrónico, usuario, contraseña y rol del usuario. El formulario incluye opciones para seleccionar el rol (en este caso, "alumno") y un botón de registro.

→ **Formulario de inicio de sesión:** También se presenta un segundo formulario que redirige a "index.php" cuando se envía, pero este fragmento de código no maneja la lógica de inicio de sesión aquí.

Parte Registrar.php:

```
1  <?php
2  require_once("funciones.php");
3
4  if (isset($_POST["registrarse"])) {
5      $nombre = $_POST["nombre"];
6      $apellido1 = $_POST["apellido1"];
7      $apellido2 = $_POST["apellido2"];
8      $correo = $_POST["correo"];
9      $usuario = $_POST["usuario"];
10     $contrasena = $_POST["contrasena"];
11     $rol = $_POST["rol"];
12
13     if (registrarUsuario($nombre, $apellido1, $apellido2, $correo, $usuario, $contrasena, $rol)) {
14         echo '<script>alert("Los datos se han insertado correctamente.");</script>';
15     }
16 }
17 ?>
18
19 <div class="login-section">
20     <h2>Registro</h2>
21     <form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="post">
22         <input type="text" name="nombre" placeholder="Nombre" required>
23         <input type="text" name="apellido1" placeholder="Apellido Paterno" required>
24         <input type="text" name="apellido2" placeholder="Apellido Materno" required>
25         <input type="email" name="correo" placeholder="Correo Electrónico" required>
26         <input type="text" name="usuario" placeholder="Usuario" required>
27         <input type="password" name="contrasena" placeholder="Contraseña" required>
28         <select name="rol" required>
29             <option value="alumno">Alumno</option>
30         </select> <br>
31         <input type="submit" id="registrarse" name="registrarse" value="Registrarse">
32     </form>
33     <form action="index.php" method="post">
34         <input type="submit" id="iniciarsesion" name="iniciarsesion" value="Iniciar Sesión">
35     </form>
36 </div>
```


Login.inc.php:

Este código PHP maneja el inicio de sesión a través de un formulario HTML. Aquí tienes una descripción resumida:

→ **Inicio de sesión:** Verifica si el formulario de inicio de sesión ha sido enviado (`isset($_POST["enviarLogin"])`). Si es así, recopila los datos del formulario (`$usuario` y `$password`) y utiliza la función `verificarInicioSesion` para autenticar al usuario.

→ **Manejo de errores:** Si la autenticación falla, establece la variable `$errValido` a `true` para mostrar un mensaje de error en el formulario HTML.

→ **Formulario HTML de inicio de sesión:** Muestra un formulario HTML que recopila el nombre de usuario y la contraseña. Si se ha establecido un error (`$errValido`), se muestra un mensaje de error en el formulario.

→ **Botón de registro:** Además del botón para enviar el formulario de inicio de sesión (`enviarLogin`), hay un segundo botón para registrarse (`registrar`). Sin embargo, en el código proporcionado, no hay una lógica específica para manejar el registro. Puede ser necesario implementar una funcionalidad adicional para el registro.

→ **Redirección después del inicio de sesión:** Si la autenticación es exitosa, se establecen variables de sesión (`$_SESSION["usuario"]` y `$_SESSION["rol"]`) y se redirige al usuario a la página principal.

Parte Login.inc.php:

```
1  <?php
2  require_once("funciones.php");
3
4  if (isset($_POST["enviarLogin"])) {
5      $usuario = $_POST["usuario"];
6      $password = $_POST["password"];
7
8      $rol = verificarInicioSesion($usuario, $password);
9
10     if ($rol) {
11         $_SESSION["usuario"] = $usuario;
12         $_SESSION["rol"] = $rol;
13
14         header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=principal");
15         exit;
16     } else {
17         $errValido = true;
18     }
19 }
20 ?>
21
22
23
24 <section class="login-section">
25     <h2>Iniciar Sesión</h2>
26     <?php if (isset($errValido) && $errValido == true) { ?>
27         <p class="error">Usuario o contraseña incorrecta</p>
28     <?php } ?>
29     <form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="post">
30         <?php if (isset($errUsuario) && $errUsuario == true) { ?>
31             <p class="error">Usuario incorrecto</p>
32         <?php } ?>
33         <label for="usuario">Escribe tu usuario</label>
34         <input type="text" id="usuario" name="usuario" placeholder="Usuario" value="<?php if (isset($usuario)) echo $usuario; ?>">
35         <?php if (isset($errPassword) && $errPassword == true) { ?>
36             <p class="error">Contraseña incorrecta</p>
37         <?php } ?>
38         <label for="password">Escribe una contraseña</label>
39         <input type="password" id="password" name="password" placeholder="Contraseña">
40         <input type="submit" id="enviarLogin" name="enviarLogin" value="Enviar">
41         <input type="submit" id="registrar" name="registrar" value="Registrarse">
42     </form>
43 </section>
```

Menu.inc.php:

Este código PHP maneja la generación dinámica de un menú de navegación basado en la sesión del usuario y su rol. Aquí tienes una descripción resumida:

- **Menú del Usuario:** Muestra diferentes elementos del menú si hay un usuario autenticado (`isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])`). Estos elementos incluyen enlaces a las páginas de "Perfil", "Buscador", "Libros", "Préstamos" y "Contacto".
- **Menú de Administrador:** Si el usuario tiene el rol de administrador (`$_SESSION["rol"]=="admin"`), se muestran elementos adicionales en el menú, incluyendo enlaces a las páginas de administración como "Crear Usuarios", "Editar Usuarios", "Eliminar Usuario", "Restaurar Usuario", "Crear Libros", "Editar Libros", "Eliminar Libro", "Restaurar Libro" y "Editar Préstamos".
- **Botón de Cerrar Sesión:** Se muestra un botón para cerrar sesión si hay un usuario autenticado. El botón redirige al usuario a la página de cerrar sesión (`?ruta=logout`).

Parte Usuario Menu.inc.php:

```
1 <nav class="menu">
2   <ul>
3     <!-- PARTE DE USUARIO ----->
4
5     <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
6       <li class="color"> --Menu User-- </li>
7     <?php endif; ?>
8
9     <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
10      <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=perfil"; ?>">Perfil</a></li>
11    <?php endif; ?>
12
13    <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
14      <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=buscador"; ?>">Buscador</a></li>
15    <?php endif; ?>
16
17    <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
18      <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=libros"; ?>">Libros</a></li>
19    <?php endif; ?>
20
21    <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
22      <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=prestamos"; ?>">Prestamos</a></li>
23    <?php endif; ?>
24
25    <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
26      <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=contacto"; ?>">Contacto</a></li>
27    <?php endif; ?>
28  </ul>
```

Parte Admin Menu.inc.php:

```
30 <!-- PARTE DE ADMINISTRACIÓN ----->
31
32 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
33 | <li class="color"> --Menu Admin-- </li>
34 <?php endif; ?>
35
36 <!-- USUARIO DE ADMINISTRACIÓN ----->
37
38 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
39 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=crearusuarios"; ?>">Crear Usuarios</a></li>
40 <?php endif; ?>
41
42 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
43 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=usuarios"; ?>">Editar Usuarios</a></li>
44 <?php endif; ?>
45
46 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
47 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=eliminarusuario"; ?>">Eliminar Usuario</a></li>
48 <?php endif; ?>
49
50
51 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
52 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=restaurarusuario"; ?>">Restaurar Usuario</a></li>
53 <?php endif; ?>
54
55 <!-- LIBRO DE ADMINISTRACIÓN ----->
56
57 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
58 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=crearlibros"; ?>">Crear Libros</a></li>
59 <?php endif; ?>
60
61 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
62 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=editarlibros"; ?>">Editar Libros</a></li>
63 <?php endif; ?>
64
65 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
66 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=eliminarlibro"; ?>">Eliminar Libro</a></li>
67 <?php endif; ?>
68
69
70 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
71 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=restaurarlibro"; ?>">Restaurar Libro</a></li>
72 <?php endif; ?>
73
74 <!-- PRESTAMOS DE ADMINISTRACIÓN ----->
75
76 <?php if(isset($_SESSION["rol"]) && !empty($_SESSION["usuario"]) && $_SESSION["rol"]=="admin"): ?>
77 | <li><a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=editarprestamo"; ?>">Editar Prestamos</a></li>
78 <?php endif; ?>
79
80 <!-- CERRAR SESION ACCESO RAPIDO ----->
81 </ul>
82 <?php if(isset($_SESSION["usuario"]) && !empty($_SESSION["usuario"])): ?>
83 | <button class="logoutmenu"> <a href="<?php echo $_SERVER["PHP_SELF"]. "?ruta=logout"; ?>"> Cerrar Sesión</button> </a>
84 <?php endif; ?>
85
86 </nav>
```

3- Resto de ficheros de la aplicación

Buscador.inc.php:

Este código PHP maneja un formulario de búsqueda de libros y muestra los resultados en una tabla HTML. Aquí tienes una descripción resumida:

→ **Procesamiento del formulario:** Si el método de solicitud es POST (`$_SERVER["REQUEST_METHOD"] == "POST"`), se recopilan y limpian los datos del formulario utilizando la función `'limpiarEntrada'` y se realiza una búsqueda de libros utilizando la función `'buscarLibros'`. El resultado se almacena en la variable `'$librosFiltrados'`.

→ **Formulario de Búsqueda:** Muestra un formulario HTML con campos para el ISBN, título, autor y editorial, y un botón de búsqueda.

→ **Mostrar Resultados:** Dependiendo de si hay resultados en `'$librosFiltrados'`, se muestra una tabla HTML con la información de los libros filtrados. Si no hay resultados, se muestra una tabla vacía.

→ **Mensajes predeterminados:** Si no se ha enviado el formulario o no hay resultados, se muestra una tabla vacía como mensaje predeterminado.

Parte Buscador.inc.php:

```
1  <?php
2  require_once("funciones.php");
3  if ($_SERVER["REQUEST_METHOD"] == "POST") {
4      $isbn = limpiarEntrada($_POST["isbn"]);
5      $titulo = limpiarEntrada($_POST["titulo"]);
6      $autor = limpiarEntrada($_POST["autor"]);
7      $editorial = limpiarEntrada($_POST["editorial"]);
8      $librosFiltrados = buscarLibros($isbn, $titulo, $autor, $editorial);
9  }
10 ?>
11 <form method="post" action="" class="filtrador">
12     <label for="isbn">ISBN:</label>
13     <input type="text" name="isbn">
14     <label for="titulo">Título:</label>
15     <input type="text" name="titulo">
16     <label for="autor">Autor:</label>
17     <input type="text" name="autor">
18     <label for="editorial">Editorial:</label>
19     <input type="text" name="editorial">
20     <input type="submit" value="Buscar">
21 </form>
22
23 <?php
24 if (isset($librosFiltrados) && !empty($librosFiltrados)) {
25     echo '<div class="contenedor3">';
26     echo '<table class="tabla-libros">';
27     echo '<tr>';
28     echo '<th>ID</th>';
29     echo '<th>ISBN</th>';
30     echo '<th>Título</th>';
31     echo '<th>Autor</th>';
32     echo '<th>Editorial</th>';
33     echo '</tr>';
34
35     foreach ($librosFiltrados as $libro) {
36         echo '<tr>';
37         echo '<td>' . $libro['id'] . '</td>';
38         echo '<td>' . $libro['isbn'] . '</td>';
39         echo '<td>' . $libro['titulo'] . '</td>';
40         echo '<td>' . $libro['autor'] . '</td>';
41         echo '<td>' . $libro['editorial'] . '</td>';
42         echo '</tr>';
43     }
44
45     echo '</table>';
46     echo '</div>';
47 } elseif (isset($librosFiltrados) && empty($librosFiltrados)) {
48     echo '<div class="contenedor3">';
49     echo '<table class="tabla-libros">';
50     echo '<tr>';
51     echo '<th>ID</th>';
52     echo '<th>ISBN</th>';
53     echo '<th>Título</th>';
54     echo '<th>Autor</th>';
55     echo '<th>Editorial</th>';
56     echo '</tr>';
57     echo '</table>';
58     echo '</div>';
```

Crearlibros.inc.php:

Este código PHP maneja la creación de un nuevo libro a través de un formulario, pero solo permite el acceso a usuarios con el rol de administrador. Aquí tienes una descripción resumida:

→ **Verificación de Permisos:** Verifica si el usuario tiene el rol de administrador (`'isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin'`). Si el usuario no tiene el rol de administrador, muestra un mensaje indicando que no tiene permisos.

→ **Procesamiento del Formulario:** Si el método de solicitud es POST (`'$_SERVER["REQUEST_METHOD"] == "POST"'`) y se ha enviado el formulario (`'isset($_POST["crearLibro"]')`), recopila los datos del formulario (ISBN, título, autor, editorial) y utiliza las funciones `'isbnExistente'` y `'crearLibro'` para verificar la existencia del ISBN y crear el libro, respectivamente.

→ **Mensajes de Éxito o Error:** Después de procesar el formulario, muestra mensajes de éxito o error mediante alertas de JavaScript. Si el libro se crea con éxito, se muestra una alerta informando al usuario y se recarga la página.

→ **Formulario de Creación de Libro:** Muestra un formulario HTML con campos para el ISBN, título, autor y editorial, y un botón para crear el libro.

→ **Mensaje de Permiso Denegado:** Si el usuario no tiene permisos de administrador, muestra un mensaje indicando que no tiene acceso a la página.

Parte Crearlibros.inc.php:

```
1  <?php
2  require_once("funciones.php");
3  if (isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin') {
4
5      if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["crearLibro"])) {
6          $isbn = $_POST["isbn"];
7          $titulo = $_POST["titulo"];
8          $autor = $_POST["autor"];
9          $editorial = $_POST["editorial"];
10
11          if (!isbnExistente($isbn) && crearLibro($isbn, $titulo, $autor, $editorial)) {
12              echo '<script>alert("Libro creado con éxito.");</script>';
13              echo '<script>>window.location.href = window.location.href;</script>';
14          } else {
15              echo '<script>alert("Error al crear el libro. El ISBN ya existe.") </script>';
16              echo '<script>>window.location.href = window.location.href;</script>';
17          }
18      }
19  ?>
20
21  <div class="contenedor-crearusuario">
22      <h2>Crear Nuevo Libro</h2>
23      <form method="post" action="">
24          <label for="isbn">ISBN:</label>
25          <input type="text" id="isbn" name="isbn" required>
26
27          <label for="titulo">Título:</label>
28          <input type="text" id="titulo" name="titulo" required>
29
30          <label for="autor">Autor:</label>
31          <input type="text" id="autor" name="autor" required>
32
33          <label for="editorial">Editorial:</label>
34          <input type="text" id="editorial" name="editorial" required>
35
36          <input type="submit" name="crearLibro" value="Crear Libro">
37      </form>
38  </div>
39
40  <?php
41
42  } else {
43      echo "No tienes permisos para acceder a esta página.";
44  }
45
46  ?>
```

Editarlibro.inc.php:

Este código PHP maneja la visualización de todos los libros en una tabla, la modificación de un libro seleccionado y la actualización de la información del libro modificado. Aquí tienes una descripción resumida:

- **Verificación de Permisos:** Verifica si el usuario tiene el rol de administrador (`'isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin'`). Si el usuario no tiene el rol de administrador, muestra un mensaje indicando que no tiene permisos.
- **Obtención de Libros:** Obtiene todos los libros de la base de datos utilizando la función `'obtenerTodosLosLibros()'` y los muestra en una tabla HTML. Cada fila de la tabla incluye un botón "Modificar" que permite acceder a la opción de modificar el libro.
- **Modificación de Libro:** Si se hace clic en el botón "Modificar" para un libro específico, se muestra un formulario prellenado con la información del libro seleccionado, utilizando la función `'obtenerLibroPorId'`.
- **Guardar Cambios:** Si se envía el formulario de modificación (`'isset($_POST['guardarCambiosLibro'])'`), se recopilan los datos modificados y se utiliza la función `'actualizarLibro'` para guardar los cambios en la base de datos.
- **Mensajes de Éxito o Error:** Después de procesar la modificación, se muestra un mensaje de éxito o error mediante alertas de JavaScript. Si los cambios se guardan con éxito, se muestra una alerta informando al usuario y se recarga la página.

Parte I Editarlibro.inc.php:

```
44 if (isset($_POST['modificarLibro'])) {
45     $idLibroModificar = $_POST['id_libro'];
46     $libroModificar = obtenerLibroPorId($idLibroModificar);
47
48     if ($libroModificar) {
49         echo '<div class="contenedor-libro">';
50         echo '<h3>Modificar Libro</h3>';
51         echo '<form method="post" action="">';
52         echo '<input type="hidden" name="id_libro_modificar" value="' . $idLibroModificar . '">';
53         echo 'ISBN: <input type="text" name="isbn" value="' . $libroModificar['isbn'] . '" required><br>';
54         echo 'Título: <input type="text" name="titulo" value="' . $libroModificar['titulo'] . '" required><br>';
55         echo 'Autor: <input type="text" name="autor" value="' . $libroModificar['autor'] . '" required><br>';
56         echo 'Editorial: <input type="text" name="editorial" value="' . $libroModificar['editorial'] . '" required><br>';
57         echo '<input type="submit" name="guardarCambiosLibro" value="Guardar Cambios">';
58         echo '</form>';
59         echo '</div>';
60     }
61 }
62
63 if (isset($_POST['guardarCambiosLibro'])) {
64     $idLibroModificar = $_POST['id_libro_modificar'];
65     $isbn = $_POST['isbn'];
66     $titulo = $_POST['titulo'];
67     $autor = $_POST['autor'];
68     $editorial = $_POST['editorial'];
69
70     if (actualizarLibro($idLibroModificar, $isbn, $titulo, $autor, $editorial)) {
71         echo '<script>alert("Los cambios han sido guardados.");</script>';
72         echo '<script>>window.location.href = window.location.href;</script>';
73     } else {
74         echo "Error al guardar los cambios.";
75     }
76 }
77 ?>
```


Parte II Editarlibro.inc.php:

```
1  <?php
2  require_once("funciones.php");
3
4  if (isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin') {
5      $libros = obtenerTodosLosLibros();
6
7      if ($libros) {
8          echo '<div class="contenedor3">';
9          echo '<table class="tabla-libros">';
10         echo '<tr>';
11         echo '<th>ID</th>';
12         echo '<th>ISBN</th>';
13         echo '<th>Título</th>';
14         echo '<th>Autor</th>';
15         echo '<th>Editorial</th>';
16         echo '<th>Acciones</th>';
17         echo '</tr>';
18
19         foreach ($libros as $libro) {
20             echo '<tr>';
21             echo '<td>' . $libro['id'] . '</td>';
22             echo '<td>' . $libro['isbn'] . '</td>';
23             echo '<td>' . $libro['titulo'] . '</td>';
24             echo '<td>' . $libro['autor'] . '</td>';
25             echo '<td>' . $libro['editorial'] . '</td>';
26             echo '<td>';
27             echo '<form method="post" action="">';
28             echo '<input type="hidden" name="id_libro" value="' . $libro['id'] . '">';
29             echo '<input type="submit" name="modificarLibro" value="Modificar">';
30             echo '</form>';
31             echo '</td>';
32             echo '</tr>';
33         }
34
35         echo '</table>';
36         echo '</div>';
37     } else {
38         echo "No se encontraron libros en la base de datos.";
39     }
40 } else {
41     echo "No tienes permisos para acceder a esta página.";
42 }
43
```

Eliminarlibro.inc.php:

Este código PHP maneja la visualización de todos los libros en una tabla, incluida la información sobre si un libro está disponible o no, y proporciona la opción de eliminar libros. Aquí tienes una descripción resumida:

→ **Verificación de Permisos:** Verifica si el usuario tiene el rol de administrador (`'isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin'`). Si el usuario no tiene el rol de administrador, muestra un mensaje indicando que no tiene permisos.

→ **Obtención de Libros:** Obtiene todos los libros de la base de datos utilizando la función `'obtenerTodosLosLibrosAdmin()'` y los muestra en una tabla HTML. Cada fila de la tabla incluye un botón "Eliminar" que permite eliminar el libro correspondiente.

→ **Eliminación de Libro:** Si se hace clic en el botón "Eliminar" para un libro específico (`'isset($_POST['eliminarLibro'])'`), se recopila la información del libro seleccionado utilizando la función `'obtenerLibroPorID'` y se utiliza la función `'eliminarLibro'` para eliminar el libro de la base de datos.

→ **Mensajes de Éxito o Error:** Después de procesar la eliminación, se muestra un mensaje de éxito o error mediante alertas de JavaScript. Si el libro se elimina con éxito, se muestra una alerta informando al usuario y se recarga la página.

Parte Eliminarlibro.inc.php:

```
1  <?php
2  require_once("funciones.php");
3  if (isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin') {
4      $libros = obtenerTodosLosLibrosAdmin();
5
6      if ($libros) {
7          echo '<div class="contenedor3">';
8          echo '<table class="tabla-libros">';
9          echo '<tr>';
10         echo '<th>ID</th>';
11         echo '<th>ISBN</th>';
12         echo '<th>Título</th>';
13         echo '<th>Autor</th>';
14         echo '<th>Editorial</th>';
15         echo '<th>Disponible</th>';
16         echo '<th>Acciones</th>';
17         echo '</tr>';
18
19         foreach ($libros as $libro) {
20             echo '<tr>';
21             echo '<td>' . $libro['id'] . '</td>';
22             echo '<td>' . $libro['isbn'] . '</td>';
23             echo '<td>' . $libro['titulo'] . '</td>';
24             echo '<td>' . $libro['autor'] . '</td>';
25             echo '<td>' . $libro['editorial'] . '</td>';
26             echo '<td>' . ($libro['disponible'] ? 'Sí' : 'No') . '</td>';
27             echo '<td>';
28             echo '<form method="post" action="">';
29             echo '<input type="hidden" name="id_libro_eliminar" value="' . $libro['id'] . '">';
30             echo '<input type="submit" name="eliminarLibro" value="Eliminar">';
31             echo '</form>';
32             echo '</td>';
33             echo '</tr>';
34         }
35
36         echo '</table>';
37         echo '</div>';
38     } else {
39         echo "No se encontraron libros en la base de datos.";
40     }
41 } else {
42     echo "No tienes permisos para acceder a esta página.";
43 }
44 if (isset($_POST['eliminarLibro'])) {
45     $idLibroEliminar = $_POST['id_libro_eliminar'];
46     $libroEliminar = obtenerLibroPorID($idLibroEliminar);
47
48     if ($libroEliminar) {
49         if (eliminarLibro($libroEliminar)) {
50             echo '<script>alert("El libro ha sido eliminado.");</script>';
51             echo '<script>>window.location.href = window.location.href;</script>';
52         } else {
53             echo "Error al eliminar el libro.";
54         }
55     }
56 }
57 ?>
```

Restaurarlibro.inc.php:

Este código PHP gestiona la visualización y restauración de libros eliminados para usuarios con el rol de administrador. Aquí tienes una descripción resumida:

→ **Verificación de Permisos:** Verifica si el usuario tiene el rol de administrador (`'isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin'`). Si el usuario no tiene el rol de administrador, muestra un mensaje indicando que no tiene permisos.

→ **Obtención de Libros Eliminados:** Obtiene todos los libros eliminados de la base de datos utilizando la función `'obtenerLibrosEliminados()'` y los muestra en una tabla HTML. Cada fila de la tabla incluye un botón "Restaurar" que permite restaurar el libro correspondiente.

→ **Restauración de Libro:** Si se hace clic en el botón "Restaurar" para un libro específico (`'isset($_POST['restaurarLibro'])'`), se recopila la información del libro seleccionado utilizando la función `'restaurarLibro'` para restaurar el libro en la base de datos.

→ **Mensajes de Éxito o Error:** Después de procesar la restauración, se muestra un mensaje de éxito o error mediante alertas de JavaScript. Si el libro se restaura con éxito, se muestra una alerta informando al usuario y se recarga la página.

Parte Restaurar libro.inc.php:

```
2 require_once("funciones.php");
3 if (isset($_SESSION['rol']) && $_SESSION['rol'] === 'admin') {
4     $librosEliminados = obtenerLibrosEliminados();
5     if ($librosEliminados) {
6         echo '<div class="contenedor3">';
7         echo '<table class="tabla-libros">';
8         echo '<tr>';
9         echo '<th>ID</th>';
10        echo '<th>ISBN</th>';
11        echo '<th>Título</th>';
12        echo '<th>Autor</th>';
13        echo '<th>Editorial</th>';
14        echo '<th>Acciones</th>';
15        echo '</tr>';
16        foreach ($librosEliminados as $libroEliminado) {
17            echo '<tr>';
18            echo '<td>' . $libroEliminado['id'] . '</td>';
19            echo '<td>' . $libroEliminado['isbn'] . '</td>';
20            echo '<td>' . $libroEliminado['titulo'] . '</td>';
21            echo '<td>' . $libroEliminado['autor'] . '</td>';
22            echo '<td>' . $libroEliminado['editorial'] . '</td>';
23            echo '<td>';
24            echo '<form method="post" action="">';
25            echo '<input type="hidden" name="id_libro_restaurar" value="' . $libroEliminado['id'] . '">';
26            echo '<input type="submit" name="restaurarLibro" value="Restaurar">';
27            echo '</form>';
28            echo '</td>';
29            echo '</tr>';
30        }
31        echo '</table>';
32        echo '</div>';
33    } else {
34        echo '<div class="contenedor3">';
35        echo '<table class="tabla-libros">';
36        echo '<tr>';
37        echo '<th>ID</th>';
38        echo '<th>ISBN</th>';
39        echo '<th>Título</th>';
40        echo '<th>Autor</th>';
41        echo '<th>Editorial</th>';
42        echo '<th>Acciones</th>';
43        echo '</tr>';
44        echo '</table>';
45        echo '</div>';
46    }
47 } else {
48     echo "No tienes permisos para acceder a esta página.";
49 }
50 if (isset($_POST['restaurarLibro'])) {
51     $idLibroRestaurar = $_POST['id_libro_restaurar'];
52
53     if (restaurarLibro($idLibroRestaurar)) {
54         echo '<script>alert("El libro ha sido restaurado.");</script>';
55         echo '<script>window.location.href = window.location.href;</script>';
56     } else {
57         echo "Error al restaurar el libro.";
58     }
59 }
```

Prestamos.inc.php:

Este código PHP gestiona la visualización y cancelación de préstamos de libros para usuarios que han iniciado sesión. Aquí tienes una descripción resumida:

- **Cancelación de Alquiler:** Si se recibe un formulario para cancelar un alquiler (`'isset($_POST['cancelarAlquiler'])'`), se procesa utilizando la función `'desalquilarLibro'`. Se muestra una alerta de JavaScript informando sobre el resultado.
- **Visualización de Préstamos del Usuario:** Si el usuario ha iniciado sesión (`'isset($_SESSION['usuario'])'`), se obtienen los préstamos del usuario con la función `'obtenerLibrosAlquilados'`. Luego, se muestran los préstamos en una tabla HTML, incluyendo información como ID de usuario, ID de libro, título, autor y fecha de préstamo. Además, se incluye un botón "Cancelar Alquiler" para cada préstamo.
- **Mensajes de Éxito o Error:** Después de procesar la cancelación de alquiler, se muestra un mensaje de éxito o error mediante alertas de JavaScript. Si el alquiler se cancela con éxito, se muestra una alerta informando al usuario.
- **Mensaje de Inicio de Sesión Requerido:** Si el usuario no ha iniciado sesión, se muestra un mensaje indicando que deben iniciar sesión para ver sus préstamos.

Parte Prestamos.inc.php:

```
2  require_once("funciones.php");
3  if (isset($_POST['cancelarAlquiler'])) {
4      $idUser = $_POST['id_usuario'];
5      $idLibro = $_POST['id_libro'];
6      if (desalquilarLibro($idUser, $idLibro)) {
7          echo '<script>alert("El alquiler ha sido cancelado.");</script>';
8      } else {
9          echo "Error al cancelar el alquiler.";
10     }
11 }
12 if (isset($_SESSION['usuario'])) {
13     $idUser = $_SESSION['id'];
14     $prestamos = obtenerLibrosAlquilados($idUser);
15     if ($prestamos) {
16         echo '<div class="contenedor3">';
17         echo '<table class="tabla-libros">';
18         echo '<tr>';
19         echo '<th>ID Usuario</th>';
20         echo '<th>ID Libro</th>';
21         echo '<th>Título</th>';
22         echo '<th>Autor</th>';
23         echo '<th>Fecha de Préstamo</th>';
24         echo '<th>Acciones</th>';
25         echo '</tr>';
26         foreach ($prestamos as $prestamo) {
27             echo '<tr>';
28             echo '<td>' . $prestamo['id_usuario'] . '</td>';
29             echo '<td>' . $prestamo['id_libro'] . '</td>';
30             echo '<td>' . $prestamo['titulo'] . '</td>';
31             echo '<td>' . $prestamo['autor'] . '</td>';
32             echo '<td>' . $prestamo['fin_prestamo'] . '</td>';
33             echo '<td>';
34             echo '<form method="post" action="">';
35             echo '<input type="hidden" name="id_usuario" value="' . $prestamo['id_usuario'] . '">';
36             echo '<input type="hidden" name="id_libro" value="' . $prestamo['id_libro'] . '">';
37             echo '<input type="submit" name="cancelarAlquiler" value="Cancelar Alquiler">';
38             echo '</form>';
39             echo '</td>';
40             echo '</tr>';
41         }
42         echo '</table>';
43         echo '</div>';
44     } else {
45         echo '<div class="contenedor3">';
46         echo '<table class="tabla-libros">';
47         echo '<tr>';
48         echo '<th>ID Usuario</th>';
49         echo '<th>ID Libro</th>';
50         echo '<th>Título</th>';
51         echo '<th>Autor</th>';
52         echo '<th>Fecha de Préstamo</th>';
53         echo '<th>Acciones</th>';
54         echo '</tr>';
55         echo '</table>';
56         echo '</div>';
57     }
58 } else {
59     echo "Debes iniciar sesión para ver tus préstamos.";
```

4- Fichero CSS

Estilos.css:

A continuación, una pequeña vista de los estilos que hay en toda la aplicación (+470 Líneas de estilos):

```
1  /*Fondo de Pantalla Blanco*/
2  body {
3      background-image: url("/DWES/ProyectoFinal/img/fondoPantalla.jpeg");
4      background-size: cover; /* Ajusta la imagen para que cubra todo el fondo */
5      background-repeat: no-repeat; /* Evita que la imagen se repita */
6      background-attachment: fixed; /* Fija la imagen en su lugar, de modo que no se desplace con el contenido */
7      font-family: Verdana, Geneva, Tahoma, sans-serif
8  }
9
10 /*-----LOGIN-----*/
11
12 .login-section {
13     background-color: #f0f0f0;
14     border: 1px solid #ddd;
15     padding: 20px;
16     text-align: center;
17     max-width: 400px;
18     margin: 0 auto;
19     border-radius: 5px;
20     float: left;
21     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
22 }
23
24 /* Estilos para el título "Iniciar Sesión" */
25 .login-section h2 {
26     font-size: 24px;
27     color: #333;
28     margin-bottom: 20px;
29 }
30
31 /* Estilos para los mensajes de error */
32 .error {
33     color: #d9534f;
34     font-weight: bold;
35     margin-bottom: 10px;
36 }
37
38 /* Estilos para los campos del formulario */
39 .login-section label {
40     display: block;
41     font-weight: bold;
42     margin-top: 10px;
43 }
44
45 .login-section select{
46     width: 90%;
47     padding: 10px;
48     margin-top: 5px;
49     border: 1px solid #ccc;
50     border-radius: 3px;
51 }
52
```


5- Fichero BBDD

A continuación, una pequeña vista de las tablas SQL que hay en toda la aplicación:

Tabla Libro:

```
30 CREATE TABLE `libro` (  
31   `id` int(11) NOT NULL,  
32   `isbn` varchar(13) DEFAULT NULL,  
33   `titulo` varchar(100) DEFAULT NULL,  
34   `autor` varchar(100) DEFAULT NULL,  
35   `editorial` varchar(100) DEFAULT NULL,  
36   `disponible` tinyint(4) DEFAULT 1  
37 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Tabla Libroeliminados:

```
84 CREATE TABLE `libroseliminados` (  
85   `id` int(11) NOT NULL,  
86   `isbn` varchar(13) DEFAULT NULL,  
87   `titulo` varchar(100) DEFAULT NULL,  
88   `autor` varchar(100) DEFAULT NULL,  
89   `editorial` varchar(100) DEFAULT NULL  
90 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Tabla Prestamo:

```
98 CREATE TABLE `prestamo` (  
99   `id_usuario` int(11) NOT NULL,  
100  `id_libro` int(11) NOT NULL,  
101  `fin_prestamo` date DEFAULT NULL  
102 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
103
```

Tabla Usuarios:

```
110 CREATE TABLE `usuarios` (  
111   `id` int(11) NOT NULL,  
112   `nombre` varchar(50) DEFAULT NULL,  
113   `apellido1` varchar(50) DEFAULT NULL,  
114   `apellido2` varchar(50) DEFAULT NULL,  
115   `correo` varchar(100) DEFAULT NULL,  
116   `usuario` varchar(50) DEFAULT NULL,  
117   `contrasena` varchar(255) DEFAULT NULL,  
118   `fecha_registro` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),  
119   `rol` enum('alumno','admin') DEFAULT NULL  
120 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Tabla Usuarioeliminados:

```
168 CREATE TABLE `usuarioseliminados` (  
169   `id` int(11) NOT NULL,  
170   `nombre` varchar(50) DEFAULT NULL,  
171   `apellido1` varchar(50) DEFAULT NULL,  
172   `apellido2` varchar(50) DEFAULT NULL,  
173   `correo` varchar(100) DEFAULT NULL,  
174   `usuario` varchar(50) DEFAULT NULL,  
175   `contrasena` varchar(255) DEFAULT NULL,  
176   `fecha_registro` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),  
177   `rol` enum('alumno','admin') DEFAULT NULL  
178 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

6- Conclusión y Justificación

En la aplicación he usado ciertos recursos que no hemos llegado a dar, pero gracias a ellos, la aplicación es más eficiente:

-En la parte de funciones.php, como has podido observar, he estado usando PDO ya que es más seguro que el MySQLi y más sencillo de realizar las consultas a la base de datos.

Es cierto que se debería de usar MySQLi con lo explicado, pero estuve mirando videos y foros que es mucho mejor y esta es la razón por la que la he usado.

-En la parte de eliminar libros o eliminar usuarios, tuve una idea la cual cuando se borrara un usuario (por ejemplo), se hiciera un insert a una tabla llamada "usuario eliminado" y un delete en la parte de usuarios. De esta manera nos aseguramos que no estaba en la tabla usuarios. Y cuando quisiera restablecer ese usuario, tendría que hacer lo mismo de manera inversa.

Es cierto que se podría poner visible y no visible y nos ahorramos una tabla, pero en su momento lo vi de esta manera y funciona sin problema.

-En la parte de préstamos, cuando alguien alquila el libro y luego lo desalquila, no se queda ningún tipo de registro de que esa persona ha alquilado ese libro. Para ello se podría crear una tabla de registro de préstamos la cual coja la información de ese usuario, el libro alquilado, y las veces que lo ha estado alquilando ese usuario ese libro. Por falta de tiempo no lo he implementado, pero se cómo se podría hacer.

En resumen, hay cosas que se podrían mejorar, cosas que se podrían depurar y quede la aplicación mucho mejor de lo que ya está de por sí.

7- Bibliografía

He estado usando todas estas webs la cual gracias a ellas me ha sido de apoyo para realizar este proyecto:

-<https://www.youtube.com/> (Funcionalidad de PDO para poder realizar las Funciones.php)

-<https://chat.openai.com/> (Errores que me salian y consultas SQL de base de datos, estilos CSS de la página)

-<https://www.php.net/> (Propiedades, funciones...)