

DOCUMENTACIÓN

PROYECTO BIBLIOTECA

Hecho por :
Nicolas Guañuna Pillajo

ÍNDICE

- 1. Introducción al proyecto.**
- 2. Explicación de la estructura del proyecto.**
- 3. Funciones.**
- 4. Views.**
 - 4.1. View Registro**
 - 4.2. View Login**
 - 4.3. View Cuerpo**
 - 4.4. View Perfil**
 - 4.5. View MisLibros**
 - 4.6. View AñadirLibro**
 - 4.7. View LibrosPrestados**
 - 4.8. View InfoLibro**
 - 4.9. View 403 Forbidden**
- 5. Conclusión**

1. Introducción del proyecto

El proyecto propuesto se centra en el desarrollo de un sistema gestor de libros que brinda soporte a la administración de libros y préstamos para usuarios registrados. Este sistema se construirá utilizando tecnologías web como PHP, HTML y CSS, con el objetivo de proporcionar una interfaz amigable y funcional para los usuarios.

La estructura de archivos y carpetas será organizada de manera coherente y eficiente para facilitar la administración del proyecto.

En cuanto a la funcionalidad del sistema, se espera que los usuarios puedan registrarse y acceder al sistema para navegar por un catálogo de libros. Dependiendo del tipo de usuario (administrador o estándar), se mostrarán diferentes opciones de navegación y funcionalidades disponibles. Los administradores tendrán acceso a características avanzadas, como la gestión total de libros, mientras que los usuarios estándar podrán buscar y solicitar préstamos de libros.

Se implementarán mecanismos de seguridad para el manejo de sesiones de usuario, validación de formularios y autenticación de acceso a fin de garantizar la protección de datos.

Antes de empezar, es necesario saber que el usuario administrador tiene estas credenciales:

Username:admin' Password:1234

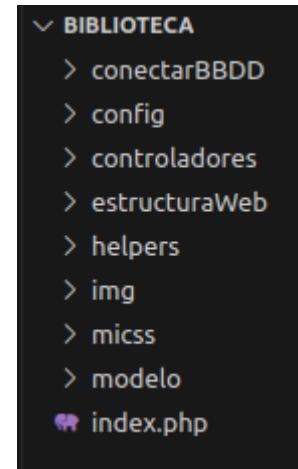
Si no deseas crear un usuario, uno de prueba es:

Username:joan' Password:1234

2. Explicación de la estructura del proyecto.

La estructura de directorios utilizada como se puede ver en la imagen, se divide en varias partes, para ser más específicos : 8 directorios y un fichero php.

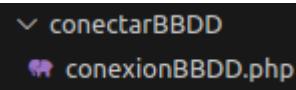
Se ha tratado de distribuir el código de la forma más clara y entendible posible.



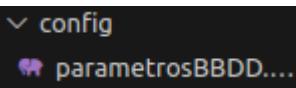
El fichero index.php es el núcleo de la página web, conteniendo la estructura principal. Aquí se ubican tanto el encabezado como las llamadas a diversas vistas en el cuerpo del HTML. Además, este archivo engloba las funciones necesarias para el funcionamiento adecuado de la página y se inicia la sesión.

```
index.php
1  <?php
2  session_start();
3  include './controladores/controlador.php';
4  ?>
5  <!DOCTYPE html>
6  <html lang="es">
7  <head>
8      <meta charset="utf-8">
9      <link rel="icon" type="favicon/x-icon" href="img/icon.png" />
10     <link rel="stylesheet" type="text/css" href="micss/style.css" />
11     <title>Bibliotech</title>
12 </head>
13 <body>
14     <?php include_once "estructuraWeb/cabecera.inc.php";?>
15
16     <div id="container">
17         <?php
18             if (empty($_GET) && !isset($_SESSION["usuario"])) {
19                 include_once "estructuraWeb/login.inc.php";
20             } elseif (!isset($_SESSION["usuario"])) {
21                 if (isset($_GET["ruta"]) && $_GET["ruta"] == "registro") {
22                     include_once "estructuraWeb/registro.inc.php";
23                 } elseif (isset($_GET["ruta"]) && $_GET["ruta"] == "login") {
24                     include_once "estructuraWeb/login.inc.php";
25                 } else {
26                     header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=login");
27                     exit;
28                 }
29             } elseif (isset($_SESSION["usuario"])) {
30                 include_once "estructuraWeb/menu.inc.php";
31             }
32         <?php
33     </div>
34 </body>
35 </html>
```

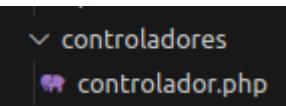
- El directorio ‘conectarBBDD’ contiene un fichero ‘conexionBBDD.php’, desde ese fichero se conecta a la base de datos y se desconecta mediante dos funciones , pasándole los parámetros de la base de datos.



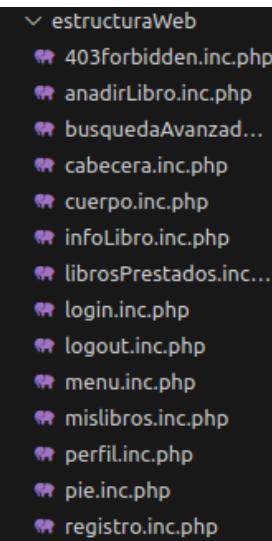
- El directorio ‘config’ contiene un fichero ‘parametrosBBDD.php’ con los datos necesarios para acceder a la base de datos y que son llamados desde ‘conexionBBDD.php’.



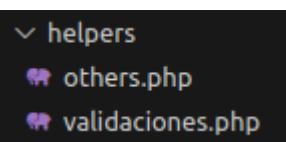
- El directorio ‘controladores’ desde donde se controlan los eventos: darle a un botón,cargar una página,existir un dato...



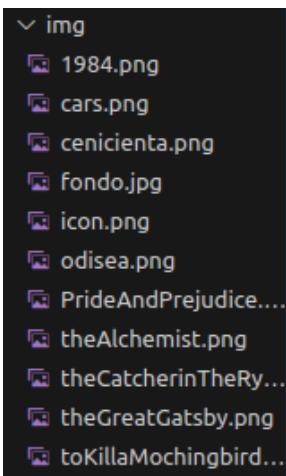
- El directorio ‘Estructura web’ contiene todas las vistas llamadas desde el index.php.



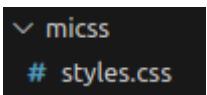
- El directorio ‘helpers’ contiene dos ficheros: ‘others.php’ y ‘validaciones.php’ .El primero como su nombre indica contiene validaciones para los datos introducidos.Por otro lado others.php se utiliza como contenedor de funciones que no tengan que ver con validaciones ni con funciones de acceso a la base de datos.



- El directorio 'img' contiene las imágenes.



- El directorio 'micss' como su nombre indica contiene el css de la página.



- Por último, el directorio 'modelo' incluye un fichero 'BBDDfunctions.php' desde el que creo funciones a la base de datos para hacer más accesible tanto el insert, como el select y el update. Además hay otro fichero desde el que se hacen las verdaderas consultas a mi base de datos.



3. Funciones

Consulta select a la base de datos y devolver un array con los datos recuperados.

```
function selectFunction($query) {
    $mysqli = conectarBBDD();
    $resultado = mysqli_query($mysqli, $query);
    $data = array();
    while ($row = mysqli_fetch_assoc($resultado)) {
        $data[] = $row;
    }
    desconectarBBDD($mysqli);
    return $data;
}
```

Realizar insert o update en la base de datos(es lo mismo pero le doy dos nombres distintos para saber con que estamos trabajando).

```
function insertFunction($query){  
    $mysqli=conectarBBDD();  
    mysqli_query($mysqli,$query);  
    desconectarBBDD($mysqli);  
}  
  
function updateFunction($query){  
    $mysqli=conectarBBDD();  
    mysqli_query($mysqli,$query);  
    desconectarBBDD($mysqli);  
}
```

Verificar las credenciales del usuario en la base de datos e iniciar la sesión de ser correctos los datos.

```
function iniciarSesion($usuario, $password) {  
    $query = "SELECT * FROM usuarios WHERE usuario='$usuario'";  
    $resultado = selectFunction($query);  
  
    if (!empty($resultado)) {  
        $userData = $resultado[0];  
        $contrasenaAlmacenada = $userData['contrasena'];  
  
        // Comprobar si la contraseña ingresada coincide con la almacenada  
        if (password_verify($password, $contrasenaAlmacenada)) {  
            $tipoUsuario = $userData['tipo'];  
            $idUsuario = $userData['id'];  
  
            // Establecer información de sesión para el usuario autenticado  
            $_SESSION["usuario"] = $usuario;  
            $_SESSION["id_usuario"] = $idUsuario;  
            $_SESSION["rol"] = ($tipoUsuario == "admin") ? "administrador" : "estandar";  
            return true; // Inicio de sesión exitoso  
        }  
    }  
  
    return false; // Inicio de sesión fallido  
}
```

En el momento de registrarse, verificar si el usuario existe o no .

```
function verificarUsuarioExistente($usuario) {  
    $query = "SELECT * FROM usuarios WHERE usuario='$usuario'";  
    $result = selectFunction($query);  
    $existeUsuario = count($result) > 0;  
    return $existeUsuario;  
}
```

Agregar al usuario a la base de datos.

```
function agregarNuevoUsuario($nombre, $apellidos, $email, $usuario, $contrasena, $pais) {  
    $contraseniahaseada = password_hash($contrasena, PASSWORD_BCRYPT);  
    $query = "INSERT INTO usuarios (nombre, apellidos, usuario, contrasena, email, pais) VALUES  
    ('$nombre', '$apellidos', '$usuario', '$contraseniahaseada', '$email', '$pais')";  
    insertFunction($query);  
    header("Location: " . $_SERVER["PHP_SELF"]);  
    exit;  
}
```

Recuperar los datos de un usuario pasándole por parámetro el id.

```
function obtenerDatosUsuario($idUsuario) {  
    $query = "SELECT nombre, apellidos, email, pais, tipo FROM usuarios WHERE id = $idUsuario";  
    $resultado = selectFunction($query);  
    return $resultado[0];  
}
```

Obtiene todos los libros disponibles en ese momento.

```
function obtenerLibrosCuerpo($librosPorPagina,$paginaActual) {
    $comienzo = ($paginaActual - 1) * $librosPorPagina;
    $queryLibro = "SELECT * FROM libros WHERE borrado=false LIMIT $comienzo, $librosPorPagina";
    return selectFunction($queryLibro);
}
```

Obtiene los libros prestados al usuario de la sesión.

```
function obtenerMisLibros($librosPorPagina, $idUsuario,$paginaActual) {
    $comienzo = ($paginaActual - 1) * $librosPorPagina;
    $queryLibro = "SELECT l.id, l.ISBN, l.titulo, p.fecha_inicio, p.fecha_fin, l.prestado FROM libros l
                  INNER JOIN prestamos p ON l.id = p.id_libro
                  WHERE p.borrado=false AND l.prestado = true AND p.id_usuario = $idUsuario
                  LIMIT $comienzo, $librosPorPagina";
    return selectFunction($queryLibro);
}
```

Obtiene todos los préstamos en activo en ese momento.

```
function obtenerLibrosPrestados($librosPorPagina,$paginaActual) {
    $comienzo = ($paginaActual - 1) * $librosPorPagina;
    $queryLibro = "SELECT libros.ISBN, prestamos.fecha_fin, usuarios.id AS id_usuario,
                   usuarios.usuario, prestamos.fecha_inicio
                   FROM libros
                   INNER JOIN prestamos ON libros.id = prestamos.id_libro
                   INNER JOIN usuarios ON prestamos.id_usuario = usuarios.id
                   WHERE prestamos.borrado=false AND libros.prestado = true
                   LIMIT $comienzo, $librosPorPagina";
    return selectFunction($queryLibro);
}
```

Esta función me permite obtener el número de páginas totales que se crearán dependiendo de la ruta en la que se encuentre.

```
// calcular el numero de paginas dependiendo de la ruta.
function calcularTotalPaginas($librosPorPagina, $idUsuario){
    if (isset($_GET['ruta'])) {
        if ($_GET['ruta'] == "cuerpo") {
            // Consulta para contar todos los libros no borrados
            $query = "SELECT * FROM libros WHERE borrado=false";
            $totalLibros = count(selectFunction($query));
            $totalPaginas = ceil($totalLibros / $librosPorPagina);
            return $totalPaginas;
        } elseif ($_GET['ruta'] == "mislibros") {
            // Consulta para contar los libros prestados al usuario
            $query = "SELECT l.id, l.ISBN, l.titulo, p.fecha_inicio, p.fecha_fin, l.prestado FROM libros l
                      INNER JOIN prestamos p ON l.id = p.id_libro
                      WHERE p.borrado=false AND l.prestado = true AND p.id_usuario = $idUsuario";
            $totalLibros = count(selectFunction($query));
            $totalPaginas = ceil($totalLibros / $librosPorPagina);
            return $totalPaginas;
        } elseif ($_GET['ruta'] == "librosPrestados") {
            // Consulta para contar todos los libros prestados
            $query = "SELECT libros.ISBN, prestamos.fecha_fin, usuarios.id AS id_usuario, usuarios.usuario, pres
                      FROM libros
                      INNER JOIN prestamos ON libros.id = prestamos.id_libro
                      INNER JOIN usuarios ON prestamos.id_usuario = usuarios.id
                      WHERE prestamos.borrado=false AND libros.prestado = true ";
            $totalLibros = count(selectFunction($query));
            $totalPaginas = ceil($totalLibros / $librosPorPagina);
            return $totalPaginas;
        }
    }
}
```

Verifica si el libro está prestado por ese usuario que se pasa por parámetro.

```
function verificarPrestamo($idUsuario,$idLibro){
    $queryPrestamo = "SELECT * FROM prestamos WHERE id_usuario = $idUsuario AND id_libro = $idLibro AND borrado=f
    return $prestamoExistente = selectFunction($queryPrestamo);
}
```

Verificar si el libro existe.

```
function verificarLibroExistente($ISBN) {
    $query = "SELECT * FROM libros WHERE ISBN='$ISBN'";
    $result = selectFunction($query);
    return count($result) > 0;
}
```

Inserto el libro en la base de datos.

```
function agregarNuevoLibro($titulo, $ISBN, $editorial, $autor, $rutaImagen) {
    $query = "INSERT INTO libros (titulo, ISBN, editorial, autor, imagen) VALUES ('$titulo', '$ISBN', '$editorial'";
    insertFunction($query);
    header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=cuerpo");
    exit;
}
```

Busca un libro en la base de datos mediante su título.

```
function buscarLibrosPorTitulo($titulo) {
    $queryLibro = "SELECT DISTINCT * FROM libros WHERE titulo LIKE '%$titulo%' AND borrado=false LIMIT 0,5";
    $libros = selectFunction($queryLibro);
    return $libros;
}
```

Realiza un update al libro para poner prestado a true e inserta en la tabla prestamos los datos del préstamo.

```
function adquirirLibro($idLibro, $paginaActual) {
    // Fecha actual
    $fecha_inicio = date('Y-m-d');
    $idUsuario = $_SESSION["id_usuario"];

    // Actualizar el estado del libro a prestado en la base de datos
    $update_query = "UPDATE libros SET prestado=true WHERE id=$idLibro";
    updateFunction($update_query);

    // Registrar el préstamo del libro para el usuario en la base de datos
    $guardar_query = "INSERT INTO prestamos(id_usuario, id_libro, fecha_inicio) VALUES ('$idUsuario','$idLibro', '$fecha_inicio')";
    insertFunction($guardar_query);

    // Redireccionar según la ruta y la página actual
    redirectSegunRuta($idLibro, $paginaActual);
}
```

Cambia en el libro el campo 'prestado' a false y en prestamos 'borrado' a true.

```
function devolverLibro($idLibro, $ruta, $paginaActual) {
    // Actualizar el estado del libro a no prestado en la base de datos
    $updateLibroQuery = "UPDATE libros SET prestado=false WHERE id=$idLibro";
    updateFunction($updateLibroQuery);

    // Marcar el préstamo del libro como borrado en la base de datos
    $borrarPrestamoQuery = "UPDATE prestamos SET borrado=true WHERE id_libro=$idLibro";
    updateFunction($borrarPrestamoQuery);

    // Redireccionar según la ruta, el libro y la página actual
    redirectSegunRuta($ruta, $idLibro, $paginaActual);
}
```

Obtener información del libro mediante la id pasada por parámetro.

```
function obtenerInfoLibro($idLibro) {
    $query = "SELECT * FROM libros WHERE id = $idLibro AND borrado = false";
    $libros= selectFunction($query);
    return $libros[0];
}
```

Cambia el libro a borrado=true.

```
function eliminarLibro($idLibro) {
    $query = "UPDATE libros SET borrado=true WHERE id = $idLibro"
    updateFunction($query);
    // Redireccionar a la página principal
    header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=cuerpo");
    exit;
}
```

Me permite realizar la consulta select a mi base de datos y con esa información dependiendo del el número de respuestas , realiza una acción u otra.

```
function realizarBusquedaAvanzada($ISBN, $titulo, $autor, $editorial) {
    // Verifica si todos los campos de búsqueda están vacíos
    if (empty($ISBN) && empty($titulo) && empty($autor) && empty($editorial)) {
        echo '<script>alert("Rellene al menos un campo");</script>';
        return;
    }

    // Construye la consulta SQL
    $query = construirConsulta($ISBN, $titulo, $autor, $editorial);
    $resultadoLibro = selectFunction($query);

    // Verifica si se encontraron resultados de búsqueda
    if ($resultadoLibro) {
        $contador = count($resultadoLibro);

        // Redirige a la información detallada del libro si se encuentra un único resultado
        if ($contador === 1) {
            $libro = $resultadoLibro[0];
            $idLibro = $libro['id'];
            header("Location: " . $_SERVER["PHP_SELF"] . "?ruta=infoLibro&id=" . $idLibro . "&isbn=$ISBN");
            exit;
        } elseif ($contador > 1) {
            // Muestra una alerta si hay más de un libro con las características proporcionadas
            echo '<script>alert("Hay más de un libro con estas características. Introduce más datos.");</script>';
        } else {
            // Muestra una alerta si no se encontraron resultados
            echo '<script>alert("No se encontraron resultados");</script>';
        }
    } else {
        // Muestra una alerta si no se encontró ningún resultado para la búsqueda
        echo '<script>alert("No se encontraron resultados para su búsqueda");</script>';
    }
}
```

Dicha función sirve para construir la consulta teniendo en cuenta los datos introducidos.

```
function construirConsulta($ISBN, $titulo, $autor, $editorial) {
    $query = "SELECT * FROM libros WHERE 1 AND borrado=false";
    if (!empty($ISBN)) {
        $query .= " AND ISBN = '$ISBN'";
    }

    if (!empty($titulo)) {
        $query .= " AND titulo = '$titulo'";
    }

    if (!empty($autor)) {
        $query .= " AND autor = '$autor'";
    }

    if (!empty($editorial)) {
        $query .= " AND editorial = '$editorial'";
    }
    return $query;
}
```

4. Views

4.1. View registro

En html no te permite dejar los campos con * en blanco y el email tiene formato email por lo que no te permite introducir un email no válido.

En PHP valida que los campos Nombre, Apellidos y País no admiten caracteres distintos a letras o espacios. Además se verifica que el usuario no exista haciendo una consulta a la base de datos.

The screenshot shows a registration form titled "Regístrate" on a website. The form fields are labeled: * Nombre, * Apellidos, * Usuario, País, * Email, and * Contraseña. Below the form is a blue button labeled "Crear cuenta" and a link "¿Ya tienes cuenta? Inicia sesión". The website has a dark header and footer with the logo "Bibliotech".

Campo vacío.

The screenshot shows a registration form with an empty "Nombre" field highlighted by a red border. Below the field, a red message says "Rellene este campo."

Formato incorrecto.

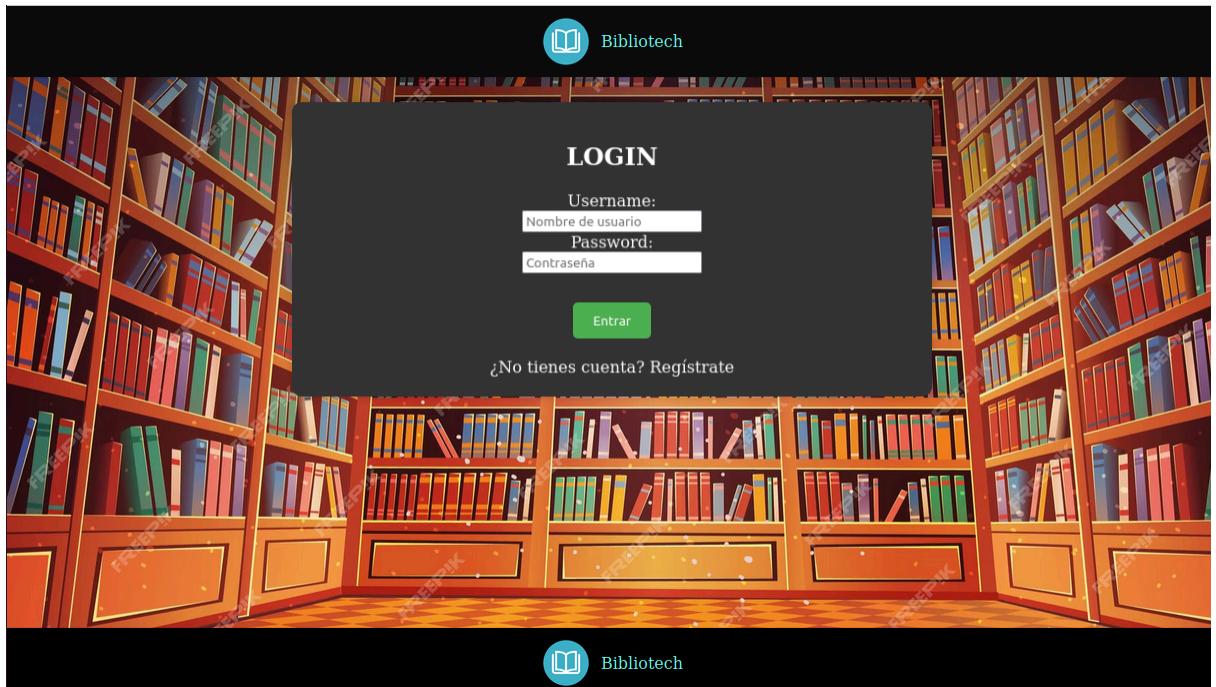
The screenshot shows a registration form with an "Nombre" field containing "Manuel3". Below the field, a red message says "Nombre no válido."

Usuario existente.

The screenshot shows a registration form with a red message box containing the text "Usuario existente."

4.2. View Login

En esta vista se verifica si los campos están vacíos o si las credenciales introducidas son incorrectas.



Campos vacíos

Campo usuario vacío

Username:

Campo contraseña vacío

Password:

Credenciales incorrectas.

Credenciales Incorrectas

4.3. View Cuerpo

View Usuario

Está vista será la que se mostrará a los usuarios. Esta vista contiene un menú con diferentes opciones, un encabezado que te da la bienvenida según tu nombre de usuario y una tabla paginada y con un buscador que funciona mediante el título.

El título es un enlace que te redirige a otra página con la información completa del libro .

El estado puede valer 3 formas diferentes:

- Si el libro no está prestado mostrará “Adquirir”.

978-1400032716	The Great Gatsby	Scribner	F. Scott Fitzgerald	Adquirir
----------------	------------------	----------	---------------------	--------------------------

- Libro prestado por otro usuario mostrará “Prestado”.

978-0061120084	To Kill a Mockingbird	HarperCollins	Harper Lee	Prestado
----------------	-----------------------	---------------	------------	--------------------------

- Libro prestado por mismo usuario de la sesión “Devolver”.

978-0142407332	The Catcher in the Rye	Little, Brown and Company	J.D. Salinger	Devolver
----------------	------------------------	---------------------------	---------------	--------------------------

The screenshot shows the Bibliotech library management system. At the top, there's a dark header with a welcome message "BIENVENIDO joan" and a search bar. Below the header, on the left, is a sidebar with navigation links: "CATÁLOGO", "BÚSQUEDA AVANZADA", "PERFIL", "MIS LIBROS", and "CERRAR SESIÓN". The main content area displays a table of books in a catalog view. The table has columns for ISBN, Título, Editorial, Autor, and Estado. Each book entry includes a green "Adquirir" button. The table shows five entries:

ISBN	Título	Editorial	Autor	Estado
978-0061120084	To Kill a Mockingbird	HarperCollins	Harper Lee	Adquirir
978-0142407332	The Catcher in the Rye	Little, Brown and Company	J.D. Salinger	Adquirir
978-1400032716	The Great Gatsby	Scribner	F. Scott Fitzgerald	Adquirir
978-0141439969	Pride and Prejudice	Penguin Classics	Jane Austen	Adquirir
978-0446310789	To Kill a Mockingbird	Warner Books	Harper Lee	Adquirir

At the bottom of the page, there are navigation buttons for "Página 1", ">", and ">>". The footer features the Bibliotech logo and the word "Bibliotech".

View Administrador

Esta es la vista para el administrador, en el encabezado te incluirá ADMINISTRADOR y como puedes observar en el menú hay más opciones que solo están disponibles para un usuario administrador.

The screenshot shows the administrator view of the Bibliotech library management system. The interface includes a sidebar with navigation options: CÁTALOGO, BÚSQUEDA AVANZADA, PERFIL, MIS LIBROS, AÑADIR LIBRO +, LIBROS PRESTADOS, and CERRAR SESIÓN. At the top, there is a search bar with placeholder text "Busque el título aquí..." and a "Buscar" button. The main content area displays a table of books with columns: ISBN, Título, Editorial, Autor, and Estado. The table contains five entries:

	ISBN	Título	Editorial	Autor	Estado
	978-0061120084	To Kill a Mockingbird	HarperCollins	Harper Lee	<button>Adquirir</button>
	978-0142407332	The Catcher in the Rye	Little, Brown and Company	J.D. Salinger	<button>Adquirir</button>
	978-1400032716	The Great Gatsby	Scribner	F. Scott Fitzgerald	<button>Adquirir</button>
	978-0141439969	Pride and Prejudice	Penguin Classics	Jane Austen	<button>Adquirir</button>
	978-0446310789	To Kill a Mockingbird	Warner Books	Harper Lee	<button>Adquirir</button>

At the bottom of the page, there is a footer with a logo and the text "Bibliotech".

4.4.View Perfil

Esta vista mostrará la información del usuario recogida en la base de datos y un enlace para cerrar sesión.



4.5.View MisLibros

En esta vista se mostrará una tabla paginada con los libros adquiridos por el usuario de la sesión, se muestra el ISBN y el título del libro, además de la fecha de inicio del préstamo y la fecha de fin de préstamo. El estado funciona igual que en la vista cuerpo.

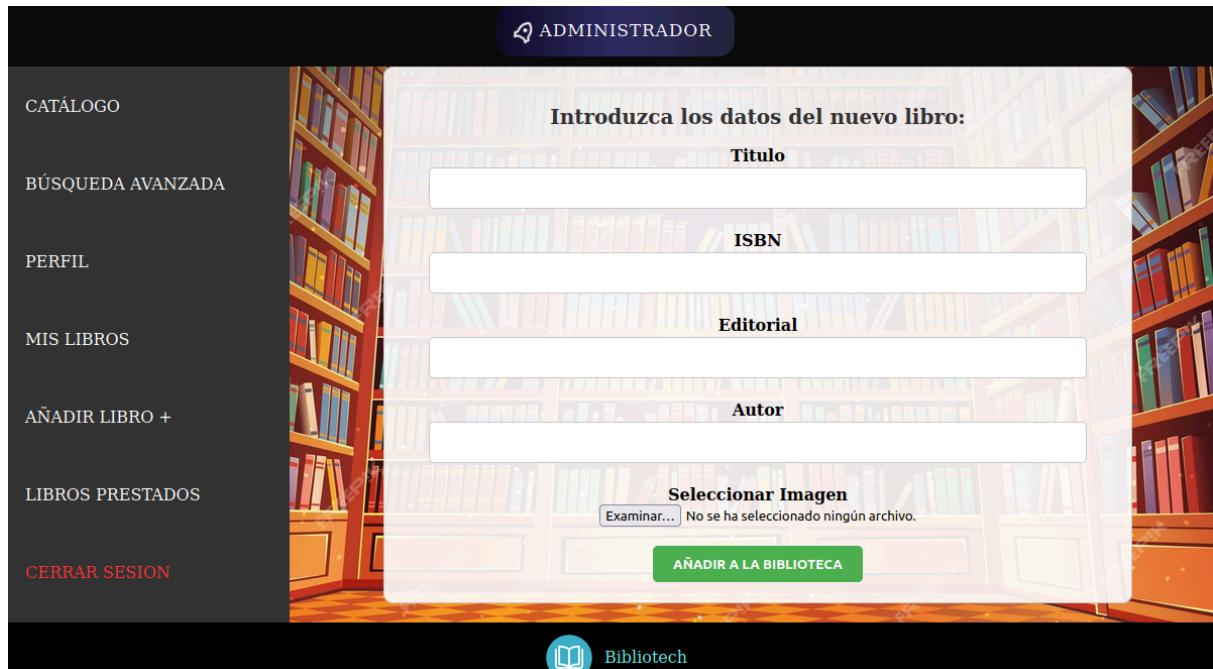
CATÁLOGO	ISBN	Título	Fecha Inicio	Fecha Fin	Estado
	978-0142407332	The Catcher in the Rye	2023-11-23	2023-12-23	Devolver

Página 1

4.6. View AñadirLibro

En esta vista el usuario administrador podrá introducir nuevos libros a la biblioteca, siempre y cuando el ISBN sea válido y no coincida con ninguno existente.

El funcionamiento de la imagen insertada es necesario explicarlo. La imagen antes de introducirla en el formulario de esta vista debe de estar en la carpetas img del proyecto porque no he encontrado código que me permita reubicar la imagen a mi carpeta directamente.



Si el formato del ISBN no es válido.



Si el ISBN ya existe.



4.7. View LibrosPrestados

En esta vista se mostrarán todos los libros prestados actualmente por cualquier usuario con sus datos y las fechas de préstamo establecidas.

	ISBN	Id Usuario	Username	Fecha Inicio	Fecha Fin
	978-0061120084	2	joan	2023-11-23	2023-12-23
	978-0142407332	4	admin	2023-11-23	2023-12-23

4.8. View InfoLibro

Esta vista mostrará la información del libro que seleccionemos en el catálogo y si el usuario es administrador permitirá eliminar ese ejemplar, por el contrario no mostrará esta opción.

Además para no tener que volver a la tabla para adquirir el libro, si el libro no está adquirido te aparecerá la opción de adquirir y si libro está prestado por el usuario de la sesión te aparecerá devolver.

4.9. View 403 Forbidden

Si eres un usuario estándar y pretendes entrar a rutas de administrador mediante la URL , no te lo permitirá y te redirigirá a esta pantalla de Permiso Denegado.



5. Conclusión

El desarrollo del proyecto de la biblioteca en PHP utilizando MySQLi ha implicado la creación y gestión de múltiples formularios distribuidos a lo largo de la página web. Estos formularios permiten la interacción del usuario con el sistema, permitiendo operaciones como la búsqueda de libros, el registro de usuarios,etc.

El uso de sesiones en PHP ha servido para identificar y autenticar a los usuarios que acceden al sistema. Gracias a estas sesiones, se determina el tipo de usuario que ha iniciado sesión, permitiendo así la visualización y ejecución de funciones específicas según los permisos asociados a cada perfil .

Además, se han empleado consultas SQL a la base de datos para recopilar información necesaria para la ejecución de diversas funcionalidades del sistema. Estas consultas se han utilizado para recuperar datos de libros, usuarios y préstamos.