



<b>I. Introducción.....</b>	<b>2</b>
A. Presentación del proyecto.....	2
B. Objetivos del proyecto.....	2
<b>II. Análisis de requerimientos.....</b>	<b>2</b>
A. Identificación de necesidades y requerimientos.....	2
B. Identificación de público.....	3
C. Estudio de mercado y competencia.....	3
<b>III. Diseño y planificación.....</b>	<b>3</b>
A. Definición de la arquitectura del proyecto.....	4
B. Diseño de la interfaz de usuario.....	5
C. Planificación de las tareas y los recursos necesarios.....	12
1. Fase de Diseño.....	12
2. Fase de Desarrollo.....	12
3. Fase de Pruebas.....	13
<b>IV. Implementación y pruebas.....</b>	<b>13</b>
A. Desarrollo de las funcionalidades del proyecto.....	13
B. Pruebas unitarias y de integración.....	14
C. Corrección de errores y optimización del rendimiento.....	15
<b>V. Documentación.....</b>	<b>16</b>
A. Documentación técnica.....	16
1. Config.....	17
2. Controladores.....	17
3. Estilos.....	21
4. Img.....	21
5. Js.....	21
6. Modelos.....	22
7. Vistas.....	24
8. Index.php.....	25
B. Documentación de usuario.....	27
C. Manual de instalación y configuración.....	30
<b>VI. Mantenimiento y evolución.....</b>	<b>30</b>
A. Plan de mantenimiento y soporte.....	30
B. Identificación de posibles mejoras y evolución del proyecto.....	31
C. Actualizaciones y mejoras futuras.....	31
<b>VII. Conclusiones.....</b>	<b>32</b>
A. Evaluación del proyecto.....	32
B. Cumplimiento de objetivos y requisitos.....	32
<b>VIII. Bibliografía y referencias.....</b>	<b>33</b>
A. Fuentes utilizadas en el proyecto.....	33
B. Referencias y enlaces de interés.....	33

## I. Introducción.

### A. Presentación del proyecto.

Marchante de Libros es una aplicación web diseñada como servicio web de una biblioteca para reserva y préstamos de libros, todo para fomentar el hábito de lectura haciendo que sea compatible la búsqueda de estos libros con la forma actual de préstamos y reservas que vemos tanto en nuestro día a día.

El objetivo principal es aumentar la atención al cliente mediante el portal a internet que ofrece esta página web y también, para mejorar la organización de la biblioteca.

En la aplicación destacamos:

- Interfaz intuitiva para los clientes: interfaz limpia, cómoda y fácil de navegar por ella.
- Interfaz cómoda para los administradores (Trabajadores): interfaz sencilla con todo lo que necesita cualquier organizador/reponedor de la biblioteca, comodidad para añadir más género y simpleza para ser efectivo y rápido.

### B. Objetivos del proyecto.

Marchante de libros tiene como objetivo ayudar a la organización de un local tanto como para los clientes (Usuarios) y para los trabajadores (Administradores), todo esto contando con la interfaz limpia y un diseño sencillo.

Busca ofrecer una experiencia rápida y sencilla, teniendo en cuenta los problemas comunes de organización y de devolución de libros, solucionando estos con más organización y limpieza a la hora de las reservas. A su vez, busca que el cliente tenga información constante del local y pueda estar informado si se añaden más libros o si alguno no está disponible o al contrario.

## II. Análisis de requerimientos.

### A. Identificación de necesidades y requerimientos.

Gestión de Usuarios:

- Registro y autenticación (Log In) de usuarios para tener acceso personalizado a la web.
- Implementación de medidas de seguridad para proteger los datos de los usuarios y accesos no autorizados.

Gestión de Productos:

- Posibilidad de ver la reserva realizada.
- Posibilidad de ver el historial de libros reservados devueltos o prestados que tienes como usuario.

**Gestión de Reservas:**

- Posibilidad de ver la reserva, tanto en un histórico como en una página hecha para las reservas
- Poder cancelar la reserva en cualquier momento

**Gestión de Préstamos:**

- Posibilidad de ver los préstamos, tanto en un histórico como en una página hecha para los préstamos
- Poder devolver el libro en cualquier momento

**B. Identificación de público.**

Esta aplicación estará orientada a todo tipo de público, prestando atención a las juventudes entre los 16 y los 24 años, puesto que son los que más acuden a las bibliotecas, pero solo para el estudio. Esta aplicación busca acercar a las generaciones más ancianas a las nuevas tecnologías y agrupar a todos con la premisa de leer algo novedoso o que nunca han leído, como al que le recomiendan una buena serie.

**C. Estudio de mercado y competencia.**

Marchante de libros se basa en una biblioteca de una localidad relativamente grande con mucha afluencia de personas de diferentes edades, pero solo existe este establecimiento de este estilo en el pueblo, muchos de los clientes seguirán siendo habituales.

Esta aplicación está pensada para facilitar al usuario el listado de libros y también para que los trabajadores tengan una mejor organización dentro del cuerpo técnico, todo esto buscando mejorar la experiencia de ambos usuarios.

**Competencia:**

No existe ninguna competencia puesto que este negocio no se dedica a la compra y venta de objetos simplemente al préstamo de un bien común y trabaja para un servicio público

### III. Diseño y planificación.

El backend de la aplicación se desarrolla utilizando PHP, SQL y peticiones asíncronas con AJAX mientras que el frontend de la aplicación está desarrollada utilizando PHP/HTML, TailWind y JavaScript.

## A. Definición de la arquitectura del proyecto.

### Backend:

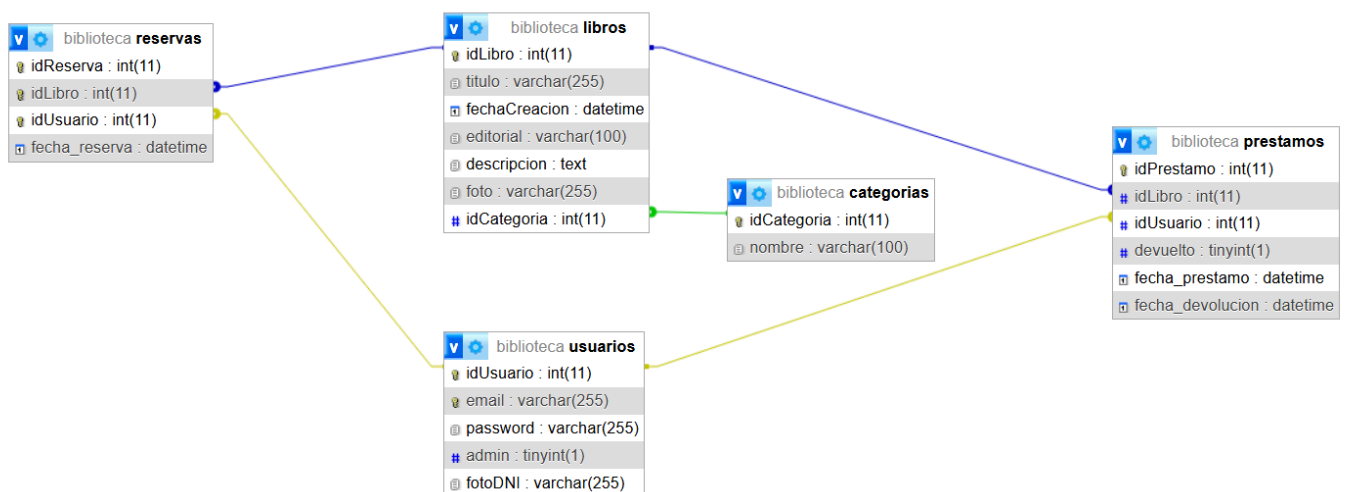
#### - Estructura general:

La estructura general del backend está realizada con PHP, como ya hemos dicho antes, pero tiene un enfoque de mapeo de diferentes acciones.

#### - Interacción con la base de datos:

La interacción con la base de datos se realiza mediante consultas SQL, las cuales se encargan de la obtención de la información y manipulación de ésta. Toda la Base de Datos está creada en PHPMyAdmin

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> <b>categorias</b>	Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> <b>libros</b>	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> <b>prestamos</b>	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> <b>reservas</b>	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> <b>usuarios</b>	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	32.0 KB
<b>5 tablas</b>	<b>Número de filas</b>	<b>25</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>176.0 KB</b>



Tablas de la Base de Datos llamada biblioteca

#### - Seguridad:

-

Esta aplicación implementa medidas de seguridad como el encriptado de las contraseñas de los usuarios y la validación de datos de entrada para prevenir ataques de inyección SQL.

También incluye una seguridad otorgada por el mapeo entre vistas de PHP que impide que se pueda navegar a través de las URLs.

## - Frontend:

El frontend está organizado en diferentes directorios dependiendo de si son las diferentes vistas de la aplicación, con aplicación de **Tailwind**, o archivos de estilos como CSS.

Este se comunica con el backend para obtener y enviar datos utilizando el método POST con formularios y peticiones, por ejemplo en las acciones del usuario, como el registro o el inicio de sesión, las cuales se realizan a través de interacciones.

## - Lenguajes utilizados:

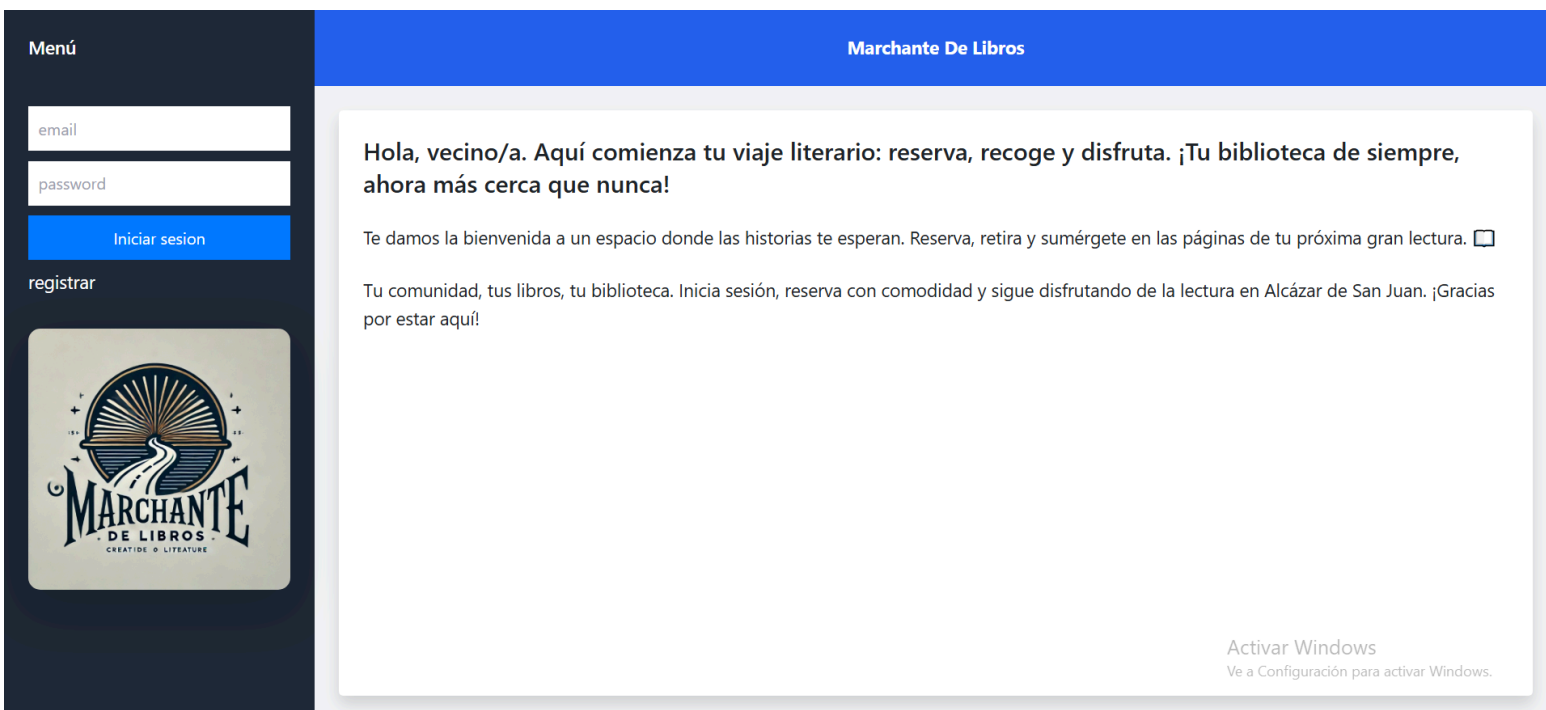
- **Html:** Con el cual se ha estructurado los elementos de la web imágenes, tablas, etc.
- **Tailwind:** El cual maneja gran parte de los estilos de la página.
- **Css :** Este solo ha sido utilizado para 4 archivos para manejar excepciones con los estilos.
- **JavaScript:** Combinado con los dos lenguajes anteriores es el encargado de proporcionar los contenidos que se actualizan dinámicamente.

## B. Diseño de la interfaz de usuario.

### Inicio:

Lo primero que podemos encontrar cuando entramos a la aplicación es un menú, que nos acompañará durante todo el proceso, con la opción de inicio de sesión. Debajo de este se encuentra el botón de registro de usuario y debajo de este el logo de la página.

A la derecha tenemos un contenedor donde se generan mensajes relacionados con la lectura. También indicamos que para avanzar se requiere del inicio sesión obligatorio.



### Página de registro:

Mantenemos a la izquierda el inicio de sesión y el logo de la página. Lo que cambia en esta página es el contenedor del centro de la página que ahora muestra un formulario para hacer el registro de un nuevo usuario.

Con campos de texto para el correo electrónico, campos de tipo password para no ver la contraseña que se quiere poner y un campo para la foto del DNI de la persona que quiera acceder.

A su vez, los botones de enviar la información y el de volver al inicio de la página.

Menú

email

password

Iniciar sesion

registrar

MARCHANTE DE LIBROS  
CREATIDE O LITRATURE

Marchante De Libros

### Registro

Correo Electrónico

Correo electrónico

Contraseña

Contraseña

Foto DNI

Seleccionar archivo Ningún archivo seleccionado

Registrar

Volver

Activar Windows  
Ve a Configuración para activar Windows.

### Página principal:

Esta es la visión de la página principal de la aplicación una vez iniciada la sesión. Como podemos observar, en el menú de la izquierda hay un saludo con el correo electrónico del usuario que haya iniciado sesión y el botón de cierre de sesión.

Cabe recalcar que, dependiendo de si el usuario es administrador o no, tendrá una vista de esta página distinta. En este caso tenemos la vista de un administrador que es la que más opciones tiene.

## Menú

Hola admin@gmail.com

Cerrar sesión



## Marchante De Libros

[Añadir Libro](#)[Ver Reservas](#)[Ver Prestamo](#)[Ver Libros](#)[Ver Historial](#)[Sobre Mí](#)

Activar Windows

Ve a Configuración para activar Windows.

Vista de la página principal desde la vista de un administrador

## Menú

Hola pablo@gmail.com

Cerrar sesión



## Marchante De Libros

[Ver Libros](#)[Ver Historial](#)[Sobre Mí](#)

Activar Windows

Ve a Configuración para activar Windows.

Vista de la página principal desde la cuenta de un usuario normal y corriente



### Añadir libro:

En esta página podemos ver la aparición de un formulario a rellenar para añadir la información de un nuevo libro sin recargar la página.

Marchante De Libros

Añadir Libro

Ver Reservas

Ver Prestamo

Ver Libros

Ver Historial

Sobre Mí

Título

Título del libro

Editorial

Editorial del libro

Foto Libro

Seleccionar archivo

Ningún archivo seleccionado

Descripción

Descripción del libro

Seleccione la categoría

Ciencia Ficción

Enviar

### Ver Reservas:

Aquí podemos ver todas las reservas actuales que hay de cada libro para el usuario que la ha solicitado, una tabla en la que aparece el libro pedido para prestar, el usuario y la fecha exacta de la reserva.

Marchante De Libros		
Añadir Libro	Ver Reservas	Ver Prestamo
Ver Libros	Ver Historial	Sobre Mí

Titulo del Libro	Email del Usuario	Fecha de la reserva
El alquimista	andres@gmail.com	2025-02-04 18:50:02
Crónica del pájaro que da cuerda al mundo	pablo@gmail.com	2025-02-04 18:50:22

Ver préstamos

Es similar a la página anterior, con la excepción de que aquí lo que vemos son los préstamos.

Marchante De Libros		
Añadir Libro	Ver Reservas	Ver Prestamo
Ver Libros	Ver Historial	Sobre Mí

Título del Libro	Email del Usuario	Fecha Préstamo
1984	pablo@gmail.com	2025-02-04 18:57:25

Ver historial:

En esta página podemos observar el histórico del usuario que esté conectado, es decir, todos los libros prestados o reservados, su estado y la fecha de devolución o de préstamo.

Marchante De Libros			
Ver Libros	Ver Historial	Sobre Mí	

Categoría	Título	Estado	Fecha de préstamo
Ciencia Ficción	1984	En propiedad	2025-02-04 18:57:25

### Ver Libros:

En esta página podemos observar todos los productos de la web que se ofrecen para su reserva y préstamo y también un buscador para los libros.

Mostraremos todo agrupado en una tabla en la que se ven el título del libro, su editorial, una foto del mismo y una breve descripción.

A la derecha tiene las opciones que, como el menú de arriba, cambian según si el usuario es administrador o usuario normal.

Marchante De Libros

Añadir Libro

Ver Reservas

Ver Prestamo

Ver Libros

Ver Historial




Sobre Mí

Busqueda por título

Escribe el libro que buscas

Buscar

Reserva de libros

Categoría	Título	Foto	Descripción	Reservar	Editar	Prestar/Devolver	Eliminar Libro
Ciencia Ficción	1984		Distopía sobre un régimen totalitario y vigilancia masiva.	Prestado	<a href="#">Editar Libros</a>	<a href="#">Devolver Libro</a>	<a href="#">Eliminar Libro</a>
Ciencia Ficción	Los juegos del hambre		En un futuro distópico, adolescentes luchan a muerte en un reality show.	Disponible	<a href="#">Editar Libros</a>	Disponible	<a href="#">Eliminar Libro</a>
							

Vista administrador de la tabla libros

**Marchante De Libros**

[Ver Libros](#)[Ver Historial](#)[Sobre Mí](#)

Busqueda por título

Escribe el libro que buscas

Buscar

**Reserva de libros**



Categoría	Título	Foto	Descripción	Reservar
Ciencia Ficción	1984		Distopía sobre un régimen totalitario y vigilancia masiva.	Prestado
Ciencia Ficción	Los juegos del hambre		En un futuro distópico, adolescentes luchan a muerte en un reality show.	<a href="#">Reservar</a>

Tabla libros desde la vista de un usuario

### Sobre mí:

A continuación mostraremos información necesaria del creador junto con su currículum

**Marchante De Libros**

[Ver Libros](#)[Ver Historial](#)[Sobre Mí](#)

**Sobre Mí**



**PABLO MARCHANTE FERNÁNDEZ**  
INFORMÁTICO

**CONTACTO**

- [pablonarfdez@gmail.com](mailto:pablonarfdez@gmail.com)
- 672 76 63 01
- [@pmarxan167](#)

**EXPERIENCIA**

Operario de producción  
Adiuvia 2020-2021, 2022-2023 y 2023-2024

Camarero  
Experiencia en bares

**ESTUDIOS**

Ingeniería informática 2º de carrera  
Universidad de Castilla la Mancha

2º DAW de la rama de informática

**SOBRE MÍ**

Me considero una persona que se organiza rápido y que no le cuesta aprender cosas nuevas. Además, tengo gran facilidad de adaptación.

Tengo por hobby la fotografía y últimamente estoy grabando videos musicales, por lo que me manejo con los programas de

Bienvenido a la sección "Sobre Mí". Este sistema de biblioteca está diseñado para facilitar la gestión de libros, reservas y préstamos de forma eficiente y accesible.

El proyecto busca ofrecer una experiencia intuitiva tanto para administradores como para usuarios, garantizando una navegación rápida y funciones personalizadas.

Está hecho por Pablo Marchante Fernández, un joven estudiante de Desarrollo de Aplicaciones Web en el I.E.S. Juan Bosco de Alcázar de San Juan.

Con este proyecto se busca fomentar la cultura lectora, que poco a poco se ha estancado en nuestra sociedad, de una forma más novedosa pero manteniendo el

## C. Planificación de las tareas y los recursos necesarios.

### 1. Fase de Diseño.

La fase de diseño es muy importante en el desarrollo de esta aplicación, puesto que establece las bases visuales y funcionales de la plataforma. También es relevante tener un diseño limpio y que invite a ser utilizado para el objetivo principal de la página.

En esta etapa, se define cómo será la experiencia del usuario y la interfaz de usuario, ha de ser intuitiva, limpia y fácil de usar. Además, se seleccionan las tecnologías a utilizar para llevar a cabo el proyecto y se diseña la estructura de la base de datos. Durante esta fase, se establecen los objetivos del proyecto y se identifican las necesidades de los usuarios.

Se ha hecho un análisis de diferentes páginas como librerías o tiendas online para ver la organización más limpia y segura de hacer esta aplicación.

La realización de esta parte del proyecto ha llevado 120 horas aproximadamente. Esto se debe a que ya tenía una organización basada en la funcionalidad de la página.

### 2. Fase de Desarrollo.

La fase de desarrollo es fundamental en la creación de cualquier proyecto, donde se codifican las funcionalidades planificadas, se integran los distintos componentes del sistema y se realizan pruebas iniciales para asegurar su correcto funcionamiento.

En esta etapa, se construyen y configuran los componentes necesarios para que la librería online funcione de manera limpia, rápida y efectiva.

Este proceso ha sido la parte más exhaustiva del proyecto, al cual he tenido que dedicar 45 días aproximadamente. En esta fase, he encontrado varios problemas que me han obligado a ir cambiando la idea inicial de este proyecto.

### 3. Fase de Pruebas.

La fase de pruebas es crucial en el desarrollo del proyecto, ya que verifica y valida el correcto funcionamiento del sistema, identificando y corrigiendo errores para asegurar que cumple con los requisitos y ofrece una experiencia de usuario satisfactoria. Todo esto gracias a una serie de pruebas para ver si todo funciona correctamente.

La implementación de esta parte del proyecto ha llevado aproximadamente 48 horas, ya que las vistas han dado bastante problemas por como estaba ejecutando la aplicación y descolocó o hacía que no se vieran algunos elementos importantes del programa.

Esto ha requerido realizar muchas pruebas, incluyendo las pruebas finales.

## IV. Implementación y pruebas.

### A. Desarrollo de las funcionalidades del proyecto.

- Usuarios
  1. Login
  2. Registro
  3. Búsqueda de libros por título
  4. Reserva de libros
  5. Préstamo de libros
  6. Ver el histórico de préstamos y reservas
- Administradores
  1. Login
  2. Autorización de préstamo de libros
  3. Búsqueda de libros por título
  4. Ver las reservas activas
  5. Ver los préstamos activos
  6. Añadir libros nuevos
  7. Editar libros ya existentes
  8. Eliminar libros existentes
  9. Devolver el libro

## B. Pruebas unitarias y de integración

<b>Rol</b>	<b>Prueba</b>	<b>Resultado</b>
Usuario	Registro de usuario.	Deberá tener todos los datos llenos.
Usuario	Registrar usuarios con correo ya existente.	Fallo, no puede registrar de nuevo a otro usuario.
Usuario	Hacer una reserva.	Cambia la tabla en la que están las opciones.
Usuario	Cancelar reserva.	Cambia la tabla en la que están las opciones.
Usuario	Botón “Ver historial”.	Enseña una tabla con el histórico de libros prestados o reservados.
Usuario/ Admin	Inicio de sesión.	Debe tener todos los datos necesarios.
Usuario/ Admin	Cierre de sesión.	Debe llevarnos a la página principal del login.
Usuario/ Admin	Inicio con datos incorrectos.	Error, nos avisará de que no hemos escrito bien algún elemento.
Usuario/ Admin	Botón “Ver libros”.	Deberá enseñarnos la tabla con todos los libros.
Usuario/ Admin	Búsqueda de libros.	Se podrá filtrar por título del libro y lo enseñará en la tabla.
Usuario/ Admin	Botón “Sobre mí”.	Llevará a la página “Sobre mí”.
Admin	Botón “Añadir libro”.	Deberá aparecer un formulario para rellenar.
Admin	Formulario “Añadir libro”.	Se debe completar todos los campos a rellenar.
Admin	Botón “Ver reservas”.	Veremos una tabla con todas las reservas actuales.
Admin	Botón “Ver Préstamos”.	Veremos una tabla con todas los préstamos actuales.
Admin	Editar libros.	Aparecerá un formulario con la información del libro actual a editar.
Admin	Formulario para editar libros.	Debe tener todos los campos a rellenar con la información pertinente.

<b>Rol</b>	<b>Prueba</b>	<b>Resultado</b>
Admin	Prestar libros	Cambiará la tabla de libros.
Admin	Devolver libro	Cambiará la tabla de libros
Admin	Eliminar libro	Cambiará la tabla de libros el libro no volverá a aparecer, enseñando la página principal

### C. Corrección de errores y optimización del rendimiento.

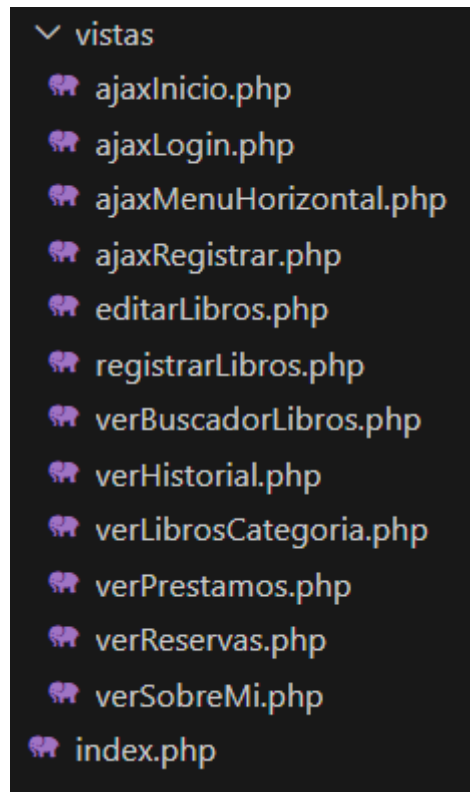
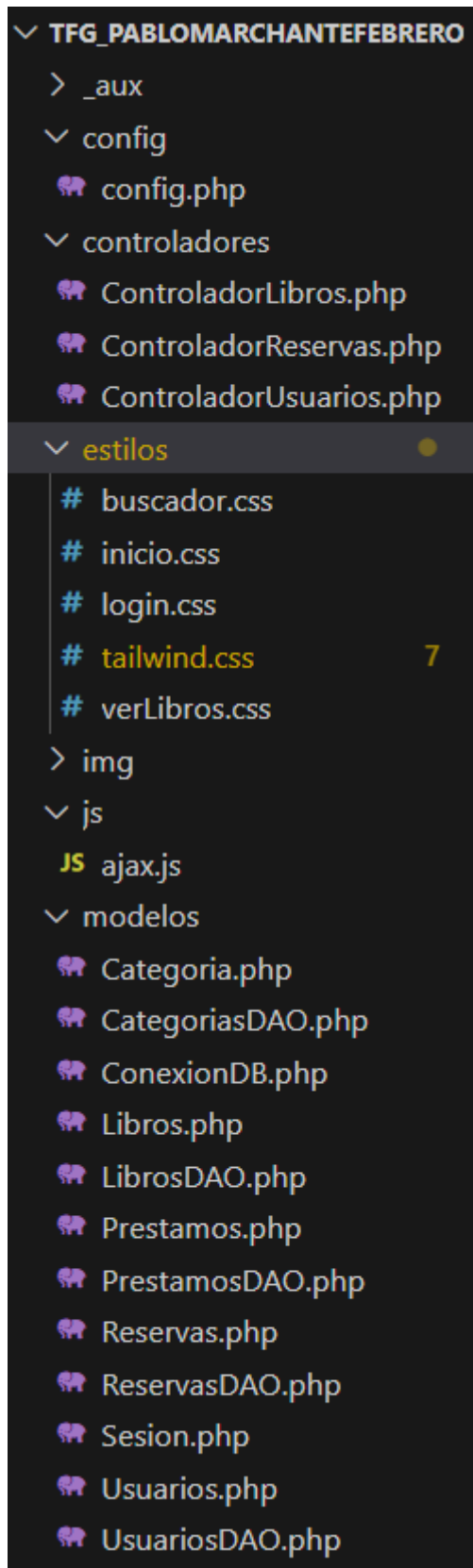
Para mejorar la eficiencia de la aplicación, se llevará a cabo una depuración del código, eliminando elementos que no aporten funcionalidades relevantes.

Asimismo, se minimizarán los redireccionamientos a otras páginas con el objetivo de optimizar la experiencia del usuario y reducir el tiempo de carga. Esto permitirá una navegación más ágil y sin interrupciones.



## V. Documentación.

### A. Documentación técnica.



Index está fuera de la carpeta vistas, es decir está en la carpeta principal del proyecto.

## 1. Config.

- config.php:

Declara constantes que almacenan la información necesaria para configurar y acceder a una base de datos MySQL.

```
config > config.php
1  <?php
2
3  define('MYSQL_USER', 'root');
4  define('MYSQL_PASS', '');
5  define('MYSQL_DB', 'biblioteca');
6  define('MYSQL_HOST', 'localhost');
```

## 2. Controladores.

- ControladorLibros.php

Este controlador en PHP gestiona las operaciones principales relacionadas con libros en el proyecto. Establece conexiones con la base de datos y utiliza el patrón DAO para acceder a esta.

También, gestiona las vistas que carga, gracias al mapeo realizado en index.php, dependiendo de la función que se ejecute.

En esta página están definidos las siguientes funciones:

- verLibros()
- buscarLibros()
- insertarLibros()
- editarLibros()
- formularioEditarLibros()
- formularioInsertarLibros()
- eliminarLibro()

Ej.:

```
public function verLibros(){  
    if($_SERVER['REQUEST_METHOD']=='GET'){  
        //Creamos la conexión utilizando la clase que hemos creado  
        $conexionDB = new ConexionDB(MYSQL_USER,MYSQL_PASS,MYSQL_HOST,MYSQL_DB);  
        $conn = $conexionDB->getConexion();  
  
        $librosDAO = new LibrosDAO($conn);  
        $libros = $librosDAO->getLibrosCategoria();  
  
        $reservasDAO = new ReservasDAO($conn);  
  
        $prestamosDAO = new PrestamosDAO($conn);  
  
        require 'vistas/verBuscadorLibros.php';  
    }  
}  
  
public function buscarLibros(){  
    if($_SERVER['REQUEST_METHOD']=='POST'){  
        //Creamos la conexión utilizando la clase que hemos creado  
        $conexionDB = new ConexionDB(MYSQL_USER,MYSQL_PASS,MYSQL_HOST,MYSQL_DB);  
        $conn = $conexionDB->getConexion();  
  
        //limpiamos los datos que vienen del usuario  
        $busqueda = htmlspecialchars($_POST['busqueda']);  
  
        $librosDAO = new LibrosDAO($conn);  
        $libros = $librosDAO->buscarLibros($busqueda);  
  
        $reservasDAO = new ReservasDAO($conn);  
  
        $prestamosDAO = new PrestamosDAO($conn);  
  
        require 'vistas/verLibrosCategoria.php';  
    }  
}
```

- ControladorReservas.php

Este controlador maneja las operaciones relacionadas con reservas y préstamos de libros en el proyecto. Todo esto sumado al control de vistas ya mencionado en el anterior controlador.

En esta página están definidos las siguientes funciones:

- a. realizarReserva()
- b. cancelarReserva()
- c. realizarPrestamo()
- d. devolverLibro()
- e. verHistorial()
- f. verReservas()
- g. verPrestamos()

Ej.:

```
public function devolverLibro(){
    if($_SERVER['REQUEST_METHOD']=='GET'){
        //Creamos la conexión utilizando la clase que hemos creado
        $conexionDB = new ConexionDB(MYSQL_USER,MYSQL_PASS,MYSQL_HOST,MYSQL_DB);
        $conn = $conexionDB->getConexion();

        $idPrestamo = $_GET["idPrestamo"];

        $prestamosDAO = new PrestamosDAO($conn);
        $prestamo = $prestamosDAO->devolverPrestamo($idPrestamo);

        $reservasDAO = new ReservasDAO($conn);

        $librosDAO = new LibrosDAO($conn);
        $libros = $librosDAO->getLibrosCategoria();

        require 'vistas/verBuscadorLibros.php';
    }
}

public function verHistorial() {
    if ($_SERVER['REQUEST_METHOD'] == 'GET') {
        // Creamos la conexión utilizando la clase que hemos creado
        $conexionDB = new ConexionDB(MYSQL_USER, MYSQL_PASS, MYSQL_HOST, MYSQL_DB);
        $conn = $conexionDB->getConexion();

        $prestamosDAO = new PrestamosDAO($conn);
        $libros = $prestamosDAO->enseniarHistorial();

        require 'vistas/verHistorial.php';
    }
}
```

- ControladorUsuarios.php

Este controlador gestiona el registro, inicio de sesión, cierre de sesión y vistas informativas del sistema.

En esta página están definidos las siguientes funciones:

- registrar()
- login()
- logout()
- inicio()
- pintarMenuHorizontal()
- verSobreMi()

Ej.:

```
public function login(){

    if($_SERVER['REQUEST_METHOD']=='POST'){
        //Creamos la conexión utilizando la clase que hemos creado
        $conexionDB = new ConexionDB(MYSQL_USER,MYSQL_PASS,MYSQL_HOST,MYSQL_DB);
        $conn = $conexionDB->getConexion();

        //limpiamos los datos que vienen del usuario
        $email = htmlspecialchars($_POST['email']);
        $password = htmlspecialchars($_POST['password']);

        //Validamos el usuario
        $usuariosDAO = new UsuariosDAO($conn);
        if($usuario = $usuariosDAO->getByEmail($email)){
            if(password_verify($password, $usuario->getPassword()))
            {
                //email y password correctos. Iniciamos sesión
                Sesion::iniciarSesion($usuario);
                Sesion::isAdminSesion($usuario->getAdmin());

                //Creamos la cookie para que nos recuerde 1 semana
                setcookie('sid',$usuario->getSid(),time()+24*60*60,'/');

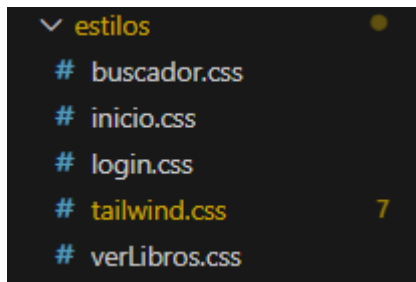
                require 'vistas/ajaxLogin.php';
                die();
                header('location: index.php');
            }
        }

        //email o password incorrectos, redirigir a index.php
        guardarMensaje("Email o password incorrectos");
        header('location: index.php');
    }
}

public function logout(){
    Sesion::cerrarSesion();
    setcookie('sid','',0,'/');
    header('location: index.php');
}
```

### 3. Estilos.

En esta carpeta están recogidos los estilos en .css de toda la aplicación web.



El archivo tailwind.css es el archivo que maneja todas las clases de tailwind.

### 4. Img.

En este directorio encontramos almacenadas todas las imágenes de usuarios y libros de la página web junto con el logo y archivos .jpeg de apoyo.

### 5. Js.

Contiene un archivo que contiene funciones en JavaScript y utiliza jQuery para realizar peticiones AJAX en un entorno web.

Actualiza el contenido del contenedor seleccionado con la respuesta del servidor.

Ej.:

```
function procesarFormularioBusqueda(){  
  
    $("#formBusqueda").submit(function (event) {  
        event.preventDefault();  
        var formData = $(this).serialize();  
        $.ajax({  
            type: "POST",  
            url: "index.php?accion=buscarLibros",  
            data: formData,  
            success: function (response) {  
  
                $( "#contenedorLibros" ).html( response );  
  
            }  
        });  
    });  
}  
  
function verLibros(){  
    $.ajax({  
        type: "GET",  
        url: "index.php?accion=verLibros",  
        success: function (response) {  
  
            $( "#contenedor" ).html( response );  
  
        }  
    });  
}
```

## 6. Modelos.

Aquí hay dos clases de archivos: los archivos en los que se definen las clases que va a utilizar la aplicación y los DAOs, en estos se definen los métodos que utilizaré en los controladores.

También encontramos, en el interior de esta carpeta, el archivo de ConexionDB.php, que se encarga de establecer y gestionar la conexión con la base de datos y Sesion.php que gestiona las sesiones de usuario en la aplicación web.

Ej.:

```
class Categoria{
    private $idCategoria;
    private $nombre;

    /**
     * Get the value of Categoria
     */
    public function getIdCategoria()
    {
        return $this->idCategoria;
    }

    /**
     * Set the value of Categoria
     */
    public function setIdCategoria($idCategoria): self
    {
        $this->idCategoria = $idCategoria;

        return $this;
    }

    /**
     * Get the value of nombre
     */
    public function getNombre()
    {
        return $this->nombre;
    }

    /**
     * Set the value of nombre
     */
    public function setNombre($nombre): self
    {
        $this->nombre = $nombre;

        return $this;
    }
}
```

- Clases.
  - a. Categoria.php
  - b. Libros.php
  - c. Prestamos.php
  - d. Reservas.php
  - e. Usuarios.php

- DAOs.
  - a. CategoriaDAO.php
  - b. LibrosDAO.php
  - c. PrestamosDAO.php
  - d. ReservasDAO.php
  - e. UsuariosDAO.php

Ej.:

```
function insertPrestamo($idLibro, $idUsuario): int|bool{
    if(!$stmt = $this->conn->prepare("INSERT INTO prestamos (idLibro, idUsuario, devuelto) VALUES (?,?,0)")){
        die("Error al preparar la consulta insert: " . $this->conn->error );
    }

    $stmt->bind_param('ii',$idLibro, $idUsuario);
    if($stmt->execute()){
        return $stmt->insert_id;
    }
    else{
        return false;
    }
}

function devolverPrestamo($idPrestamo): int|bool{
    if(!$stmt = $this->conn->prepare("UPDATE prestamos SET devuelto = 1 WHERE idPrestamo = ?")){
        die("Error al preparar la consulta insert: " . $this->conn->error );
    }

    $stmt->bind_param('i', $idPrestamo);
    if($stmt->execute()){
        return true;
    }
    else{
        return false;
    }
}
```



- Conexión.
  - a. ConexionDB.php
  - b. Sesion.php

```
<?php
class ConexionDB{
    private $conn;

    function __construct($user, $password, $host, $database)
    {
        $this->conn = new mysqli($host,$user,$password,$database);
        if($this->conn->connect_error){
            die('Error al conectar con MySQL');
        }
    }

    function getConexion(){
        return $this->conn;
    }
}
```

```
<?php
class Sesion{
    static public function getUsuario():Usuario|false{
        if(isset($_SESSION['usuario'])){
            return unserialize($_SESSION['usuario']);
        }else{
            return false;
        }
    }

    static public function iniciarSesion($usuario){
        $_SESSION['usuario'] = serialize($usuario);
    }

    static public function IsAdminSesion($admin){
        $_SESSION['admin'] = $admin;
    }

    static public function cerrarSesion(){
        unset($_SESSION['usuario']);
        unset($_SESSION['admin']);
    }

    static public function existeSesion(){
        if(isset($_SESSION['usuario'])){
            return true;
        }else{
            return false;
        }
    }
}
/**
```

## 7. Vistas.

Aquí están los archivos con las vistas del programa, es decir, lo que verá e interactúa el usuario.

Son los siguientes archivos:

- a. ajaxInicio.php
- b. ajaxLogin.php
- c. ajaxMenuHorizontal.php
- d. ajaxRegistrar.php
- e. editarLibros.php
- f. registrarLibros.php
- g. verBuscadorLibros.php
- h. verHistorial.php
- i. verLibrosCategoria.php
- j. verPrestamos.php
- k. verReservas.php
- l. verSobreMi.php

## 8. Index.php.

En index.php se define el array \$mapa que hace de mapeo entre rutas en el que cada clave dentro del array representa una ruta o acción, y su valor es otro array con información relevante.

Podemos destacar dentro del mapeo la siguiente información:

- ❖ 'controlador'

-El nombre del controlador que maneja la ruta.

- ❖ 'metodo'

-El método que hay dentro del controlador que debe ejecutar.

- ❖ 'privada'

-Un indicador booleano que indica si la ruta requiere autenticación o algún permiso especial.

```
<?php
```

```
require_once 'config/config.php';
require_once 'modelos/ConexionDB.php';
require_once 'modelos/Usuarios.php';
require_once 'modelos/UsuariosDAO.php';
require_once 'controladores/ControladorUsuarios.php';
require_once 'modelos/Sesion.php';
require_once '_aux/funciones.php';
require_once 'controladores/ControladorReservas.php';
require_once 'controladores/ControladorLibros.php';
require_once 'modelos/Reservas.php';
require_once 'modelos/ReservasDAO.php';
require_once 'modelos/Libros.php';
require_once 'modelos/LibrosDAO.php';
require_once 'modelos/Categoria.php';
require_once 'modelos/Prestamos.php';
require_once 'modelos/PrestamosDAO.php';
require_once 'modelos/CategoriasDAO.php';

//Uso de variables de sesión
session_start();

//Mapa de enrutamiento
$mapa = array(
    'inicio'=>array('controlador'=>'ControladorUsuarios',
        'metodo'=>'inicio',
        'privada'=>false),
    'login'=>array('controlador'=>'ControladorUsuarios',
        'metodo'=>'login',
        'privada'=>false),
    'logout'=>array('controlador'=>'ControladorUsuarios',
        'metodo'=>'logout',
        'privada'=>true),
    'registrar'=>array('controlador'=>'ControladorUsuarios',
        'metodo'=>'registrar',
        'privada'=>false),
    'inicioReserva'=>array('controlador'=>'ControladorReservas',
        'metodo'=>'inicio',
        'privada'=>true),
    'verLibros'=>array('controlador'=>'ControladorLibros',
        'metodo'=>'verLibros',
        'privada'=>true),
    'realizarReserva'=>array('controlador'=>'ControladorReservas',
        'metodo'=>'realizarReserva',
        'privada'=>true).
```

Se realiza un análisis de la ruta, verificando si la URL contiene un parámetro de acción. Si el parámetro está presente, se comprueba si coincide con alguna acción definida en el mapa. En caso de no encontrar coincidencias, se devuelve un error 404 mostrando el mensaje "Página no encontrada". Si no se especifica ningún parámetro de acción, se asigna la acción por defecto como "login".

Si el usuario ya ha iniciado sesión y existe la cookie, se inicia sesión con los datos del usuario:

```
//Parseo de la ruta
if(isset($_GET['accion'])){ //Compruebo si me han pasado una acción concreta, sino pongo la acción por defecto inicio
    if(isset($_mapa[$_GET['accion']])){ //Compruebo si la acción existe en el mapa, sino muestro error 404
        $accion = $_GET['accion'];
    }
    else{
        //La acción no existe
        header('Status: 404 Not found');
        echo 'Página no encontrada';
        die();
    }
}
else{
    $accion='inicio'; //Acción por defecto
}
```

```
//Si existe la cookie y no ha iniciado sesión, le iniciamos sesión de forma automática
//if( !isset($_SESSION['email']) && isset($_COOKIE['sid'])){
if( !Session::existeSesion() && isset($_COOKIE['sid'])){
    //Conectamos con la BD
    $conexionDB = new ConexionDB(MYSQL_USER,MYSQL_PASS,MYSQL_HOST,MYSQL_DB);
    $conn = $conexionDB->getConexion();

    //Nos conectamos para obtener el id y la foto del usuario
    $usuariosDAO = new UsuariosDAO($conn);
    if($usuario = $usuariosDAO->getBySid($_COOKIE['sid'])){
        //$_SESSION['email']=$usuario->getEmail();
        //$_SESSION['id']=$usuario->getId();
        //$_SESSION['foto']=$usuario->getFoto();
        Session::iniciarSesion($usuario);
    }
}
```

Luego se verifica el acceso a las acciones privadas. Si la acción actual está marcada como privada en el mapa, se comprueba si el usuario tiene una sesión activa. En caso contrario, se redirige al usuario a la página de inicio.

```
//Si la acción es privada compruebo que ha iniciado sesión, sino, lo echamos a index
// if(!isset($_SESSION['email']) && $mapa[$accion]['privada']){
if(!Sesion::existeSesion() && $mapa[$accion]['privada']){
    header('location: index.php');
    guardarError("Debes iniciar sesión para acceder a $accion");
    die();
}
```

Finalmente, se determina el nombre del controlador y el método asociados a la acción actual. Se crea una instancia de dicho controlador y se invoca el método correspondiente.

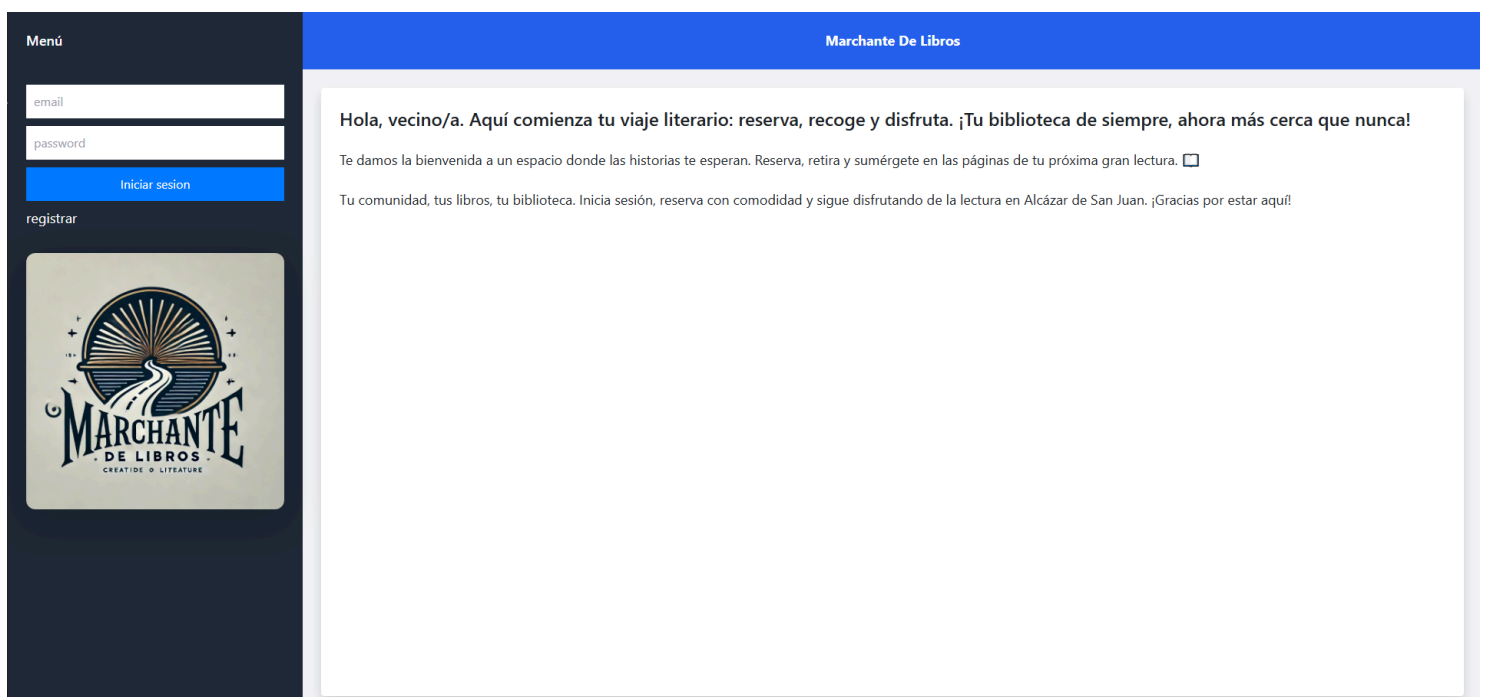
```
//$acción ya tiene la acción a ejecutar, cogemos el controlador y metodo a ejecutar del mapa
$controlador = $mapa[$accion]['controlador'];
$metodo = $mapa[$accion]['metodo'];

//Ejecutamos el método de la clase controlador
$objeto = new $controlador();
$objeto->$metodo();
```

## B. Documentación de usuario.

### Página de inicio:

La primera página que veremos al acceder a la aplicación. Contiene el Login, un botón para registrarse, el logotipo de la página y un contenedor con mensajes de ayuda.



**Registro:**

Solo lo encontraremos en la página de inicio, para asegurar que el usuario inicie sesión antes de poder realizar cualquier otra acción.

Cuando se pulse el botón, aparecerá un formulario de registro en el contenedor donde estaban los mensajes mentados anteriormente, manteniendo la posibilidad de iniciar sesión en el menú de la izquierda.

The image shows a web interface with a dark blue sidebar on the left and a main content area on the right. The sidebar, titled 'Menú', contains input fields for 'email' and 'password', a blue button labeled 'Iniciar sesion', and a link labeled 'registrar'. Below the sidebar is a large image of the 'Marchante De Libros' logo, which features a stylized sunburst and the text 'MARCHANTE DE LIBROS CREATIDE O LITEATURE'. The main content area has a blue header bar. Below it, a white box titled 'Registro' contains the registration form. The form has three sections: 'Correo Electrónico' with a text input field, 'Contraseña' with a text input field, and 'Foto DNI' with a file selection button labeled 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. At the bottom of the form is a large blue button labeled 'Registrar' and a smaller blue link labeled 'Volver'.

**Menú de opciones:**

Una vez iniciada la sesión como usuario, el contenedor cambiará para mostrarnos estas 3 opciones.

**Marchante De Libros**[Ver Libros](#)[Ver Historial](#)[Sobre Mí](#)

Ver libros:

Al pulsar el contenedor volverá a cambiar sin actualizar la página, mostrando una tabla con el listado de todos los libros junto con un buscador por títulos.




En esta tabla, el usuario, tendrá la opción de reservar un libro o cancelar dicha reserva, también podrá ver si el libro está prestado o no.

Busqueda por titulo

Escribe el libro que buscas

Buscar

Reserva de libros

Categoría	Título	Foto	Descripción	Reservar
Ciencia Ficción	1984		Distopía sobre un régimen totalitario y vigilancia masiva.	Prestado
Ciencia Ficción	Los juegos del hambre		En un futuro distópico, adolescentes luchan a muerte en un reality show.	<a href="#">Reservar</a>
				

Ver historial:

Cargará la vista del histórico de reservas y préstamos de la sesión del mismo usuario, todo esto en otra tabla con la información pertinente para el mismo.

Categoría	Título	Estado	Fecha de préstamo
Ciencia Ficción	1984	En propiedad	2025-02-04 18:57:25

Sobre mí:

Cuando pulsemos en este botón aparecerá una página nueva en el contenedor con información del creador de la página junto con su currículum.

### C. Manual de instalación y configuración.

Para poder trabajar con la web habrá que configurar localmente algunas cosas:

1. Instalación de XAMPP.
  - ❖ Instalamos XAMPP en el ordenador.
  - ❖ Añadimos nuestra carpeta con el proyecto a C:\xampp\htdocs\.
2. Configuración BDD.
  - ❖ Iniciamos, en XAMPP, los servicios de Apache y de MySQL.
  - ❖ Accedemos a PHPMyAdmin poniendo en la URL lo siguiente: <http://localhost/phpmyadmin/>
  - ❖ Importamos a PHPMyAdmin el archivo de la base de datos que está adjunta al archivo del proyecto.

## VI. Mantenimiento y evolución.

### A. Plan de mantenimiento y soporte.

- Mantenimiento correctivo, como actualizaciones para la corrección de errores, dar solución a problemas relacionados con el servidor o frontend y verificación o restauración de posibles archivos corruptos. Todo esto, aplicado con una frecuencia de bajo demanda o, según los incidentes, de forma continua.
- Mantenimiento preventivo, que se aplica con la actualización de componentes del sistema, limpieza y optimización de la base de datos y con pruebas periódicas de funcionalidad y usabilidad, ejercido con una frecuencia mensual.
- Un mantenimiento evolutivo, todo esto pensado en incorporar nuevas tecnologías a la aplicación para mejorar la interfaz o la usabilidad de la misma, a su vez, incrementando la satisfacción del usuario. Aplicado con una frecuencia pactada en la estrategia.

#### B. Identificación de posibles mejoras y evolución del proyecto.

- Añadir un área de atención al cliente, con la intención de mejorar la experiencia del usuario y poder corregir la aplicación según surjan problemas en la vista del cliente.
- Integración de notificaciones, tanto como para la vista del usuario como para el administrador. Con esta integración, las interacciones con la página pueden ser más rápidas y hacer que el uso de la página sea más rápido y cómodo.
- Agregar funciones para el usuario, como por ejemplo la de editar el perfil o la opción para que pueda donar libros a la biblioteca, puesto que este servicio se presenta gratuito como préstamo de libros. Aunque podemos incorporar, también, una subpágina de compra/venta de libros como complemento de la aplicación.

#### C. Actualizaciones y mejoras futuras.

- Integración de nuevas funcionalidades, como las nombradas en el apartado anterior.
- Adaptaciones del código y la aplicación a nuevas tecnologías, puesto que esta aplicación debe estar actualizada en todo momento.
- Añadir un refuerzo a la seguridad del programa, haciendo así que la privacidad de los usuarios esté bien tratada y ayudando a prevenir próximas vulnerabilidades.
- Ajustar el código del programa y la infraestructura del mismo para manejar mejor el tráfico de información entre el programa y la base de datos.



## VII. Conclusiones.

### A. Evaluación del proyecto.

Con el proyecto de 'Marchante de libros' se busca la integración, nuevamente, de la lectura en la sociedad actual, integrando la reserva y préstamo de libros con las nuevas tecnologías en un poblado con disponibilidad de una sola biblioteca.

La base de datos no ha sido un problema, puesto que, tiene un esquema muy similar a cualquier BDD que tenga relación con tiendas o en el que exista una adquisición de objetos. La mayor complejidad del proyecto es a la hora de distribuir cada una de las opciones que tienen los usuarios, que aunque haya dado problemas a la hora del desarrollo, el trabajo final tiene un resultado muy satisfactorio.

### B. Cumplimiento de objetivos y requisitos.

En cuanto a los requisitos que han sido impuestos para este proyecto, puedo asegurar que esta aplicación cumple con todos y cada uno de ellos a muy buen nivel. Por lo que, también puedo afirmar que los objetivos los cumple puesto que he conseguido mejorar con la programación en PHP, en la mezcla de PHP/HTML con la que siempre he tenido dudas y en el uso de los DAOs junto con el javascript.

## VIII. Bibliografía y referencias.

### A. Fuentes utilizadas en el proyecto.

El hecho de que trabaje con Tailwind ha hecho posible que la fuente para el texto utilizada dentro de la aplicación sea la *sans-serif genérica*.

En cuanto a los colores utilizados podemos distinguir:

- Blanco (#FFFFFF)
- Medianoche (#1f2937)
- Azul (#2563eb)
- Gris (#f3f4f6)

### B. Referencias y enlaces de interés.

- Página oficial de PHP: <https://www.php.net/manual/es/intro-what-is.php>
- Github: <https://github.com/>
- Página de Tailwind: <https://tailwindui.com/components#product-application-ui>
- Página de ayuda de CSS: <https://www.w3schools.com/css/>
- Compilador de HTML: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_editor](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_editor)
- Página de ayuda de Bootstrap: <https://getbootstrap.com/>
- Lenguaje JS: <https://lenguajejs.com/>