

MVC Biblioteca ALV



MVC Biblioteca ALV

ÍNDICE

1.0. Ventajas del patrón MVC.....	3
2.0. Comparativa con el proyecto de la primera evaluación.....	3
3.0. Justificación del diseño.....	3
Fichero .htaccess.....	3
Carpeta view.....	4
Carpeta controller.....	4
Carpeta model.....	5
Carpeta css.....	7
Carpeta sql.....	8
Carpeta documentation.....	9
4.0. Enlace al repositorio de GitHub.....	9

MVC Biblioteca ALV

1.0. Ventajas del patrón MVC

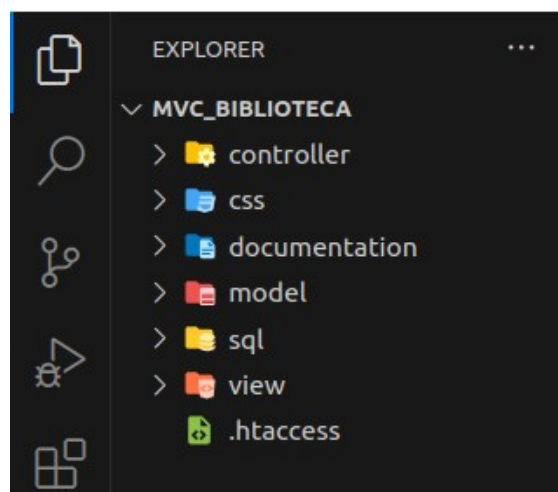
La principal ventaja del patrón MVC es la separación del código haciendo el código más legible por un programador, la reutilización del código, la resolución de problemas en el código al no estar todo mezclado y la posibilidad de división del equipo para una mayor productividad segmentando el equipo en los diferentes aspectos del patrón MVC.

2.0. Comparativa con el proyecto de la primera evaluación

La principal diferencia con el proyecto de la primera evaluación es la reutilización del código, la legibilidad del código y la facilidad de corrección de errores ya que en el proyecto de la primera evaluación es bastante difícil la reutilización del código, la legibilidad del código ya que esta mezclado la vista, el modelo y el controlador y por consecuencia de eso la corrección de errores que es bastante más complejo solucionarlos porque no sabes bien donde pueda estar que eso en el patrón MVC no pasa.

3.0. Justificación del diseño

He decidido hacer una separación bastante agresiva separando todo al máximo posible para una mayor reutilización y legibilidad del código. Lo he separado en las siguientes carpetas:



Fichero .htaccess

El fichero .htaccess es un fichero de configuración donde nos permite activar las rutas amigables entre otras cosas pero que principalmente lo he utilizado para implementar las rutas amigables para un mayor posicionamiento en las búsquedas de los usuarios por internet ya que es como trabaja los buscadores con rutas amigables. Las rutas amigables son básicamente rutas sencillas que redirecciona a rutas no tan amigables con el usuario.

MVC Biblioteca ALV

```

1 # Regla que dice si un directorio es un enlace simbólico, que lo siga.
2 Options +FollowSymLinks
3 # Regla donde le decimos a Apache que queremos sobre escribir las URLs para crear URLs amigables.
4 RewriteEngine on
5 # Regla de reescritura: RewriteRule ^url-amigable-aqui$ url-no-amigable-va-aqui.php
6 # Si no escribes nada automaticamente te dirige al controlador principal de la página
7 RewriteRule ^$ controller/principal.php
8 # Si la ruta contiene BorrarLibro va al controlador encargado de borrar el libro con el id indicado en la ruta
9 RewriteRule ^BorrarLibro/([0-9]+)$ controller/borrarLibro.php?id=$1
10 # Si la ruta contiene Libros va al controlador encargado de mostrar el formulario para agregar un nuevo libro
11 RewriteRule ^Libros$ controller/formContLibro.inc.php
12 # Si la ruta contiene GrabarLibro va al controlador encargado de guardar el libro en la base de datos
13 RewriteRule ^GrabarLibro$ controller/grabarLibro.php
14 # Ruta para el estilo de la página
15 RewriteRule ^css/style$ css/style.css

```

Carpeta view

La carpeta view contiene la vista de la página es decir lo que ven los usuarios.

Dentro de la carpeta view hay dos vistas una llamada listado.inc.php donde muestra los libros que este almacenado en la base de datos y con una opción que es borrar el libro que borra el libro de la base de datos y el otro archivo es formularioLibro.inc.php de donde hay un formulario para agregar un libro a la base de datos.

```

view > formularioLibro.inc.php > html > head > title
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width,
6 initial-scale=1.0">
7 <link rel="stylesheet" href="<?php echo $ruta?>/css/
8 style">
9 <title>Libreria ALV</title>
10 </head>
11 <body>
12 <form action="<?php echo $ruta?>/GrabarLibro"
13 enctype="multipart/form-data" method="POST">
14 <label for="titulo">Titulo</label><br>
15 <input type="text" id="titulo" name="titulo"><br><br>
16 <label for="autor">Autor</label><br>
17 <input type="text" id="autor" name="autor"><br><br>
18 <label for="descripcion">Descripción</label><br>
19 <textarea name="descripcion" cols="60" rows="6"
20 id="descripcion"></textarea><br><br>
21 <input type="submit" value="Aceptar">
22 </form>
23 </body>
24 </html>

```

```

view > listado.inc.php > html > body > h1
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width,
6 initial-scale=1.0">
7 <link rel="stylesheet" href="<?php echo $ruta?>/css/
8 style">
9 <title>Libreria ALV</title>
10 </head>
11 <body>
12 <h1>Biblioteca ALV</h1>
13 <hr>
14 <!-- Rutas con url amigables -->
15 <a id="newBook" href="<?php echo $ruta?>/Libros">Agregar nuevo
16 libro</a>
17 <hr>
18 <!-- modificar casi todo -->
19 <?php
20 foreach($data['libros'] as $libro) {
21 <?>
22 <h3><?php echo $libro->getTitulo();?></h3>
23 <p>Autor: <?php echo $libro->getAutor();?></p><br>
24 <p id="width40"><?php echo $libro->getDescripcion();?></p><br>
25 <a id="cssBorrar" href="<?php echo $ruta."/
26 BorrarLibro/".$libro->getId();?>">Borrar libro</a>
27 <?php
28 }
29 <?>
30 </body>
31 </html>

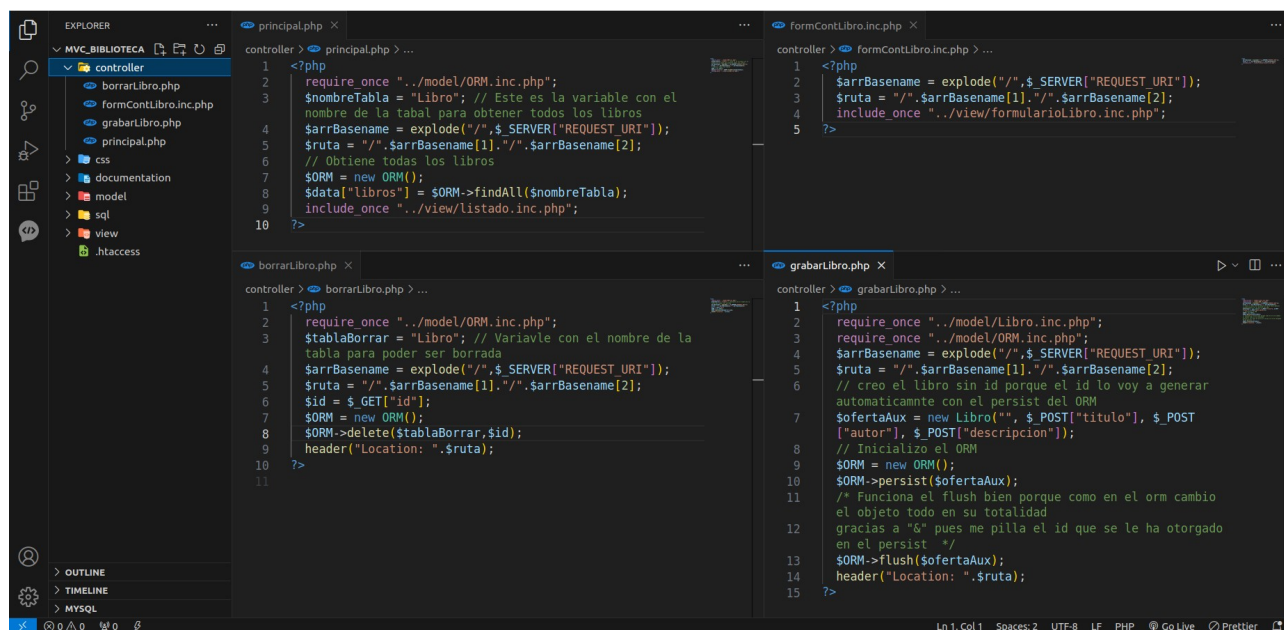
```

Carpeta controller

La carpeta controller contiene los controladores que son los encargados de hacer todo, es decir, el cerebro de la página web.

Dentro de esta carpeta hay cuatro archivos uno que es principal.php que es el encargado de mostrar todos los libros que han en la base de datos mediante la vista del archivo listado.inc.php, otro archivo que es formContLibro.inc.php que es el encargado de visualizar el formulario con la opción de grabar los datos introducidos para crear un nuevo libro, otro archivo que es borrarLibro.php que es el encargado de borrar el libro seleccionado mediante su id y por ultimo el fichero grabarLibro.php que es el encargado de guardar el libro de los datos introducidos en el formulario.

MVC Biblioteca ALV

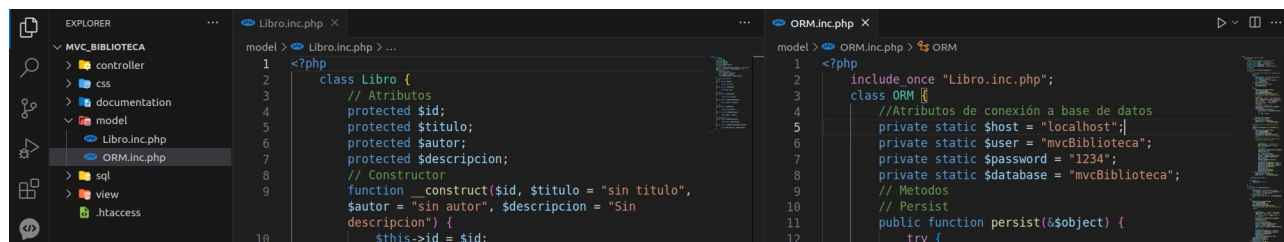


Carpeta model

La carpeta model contiene la conexiones y archivos necesarios para la conexión y consulta con la base de datos.

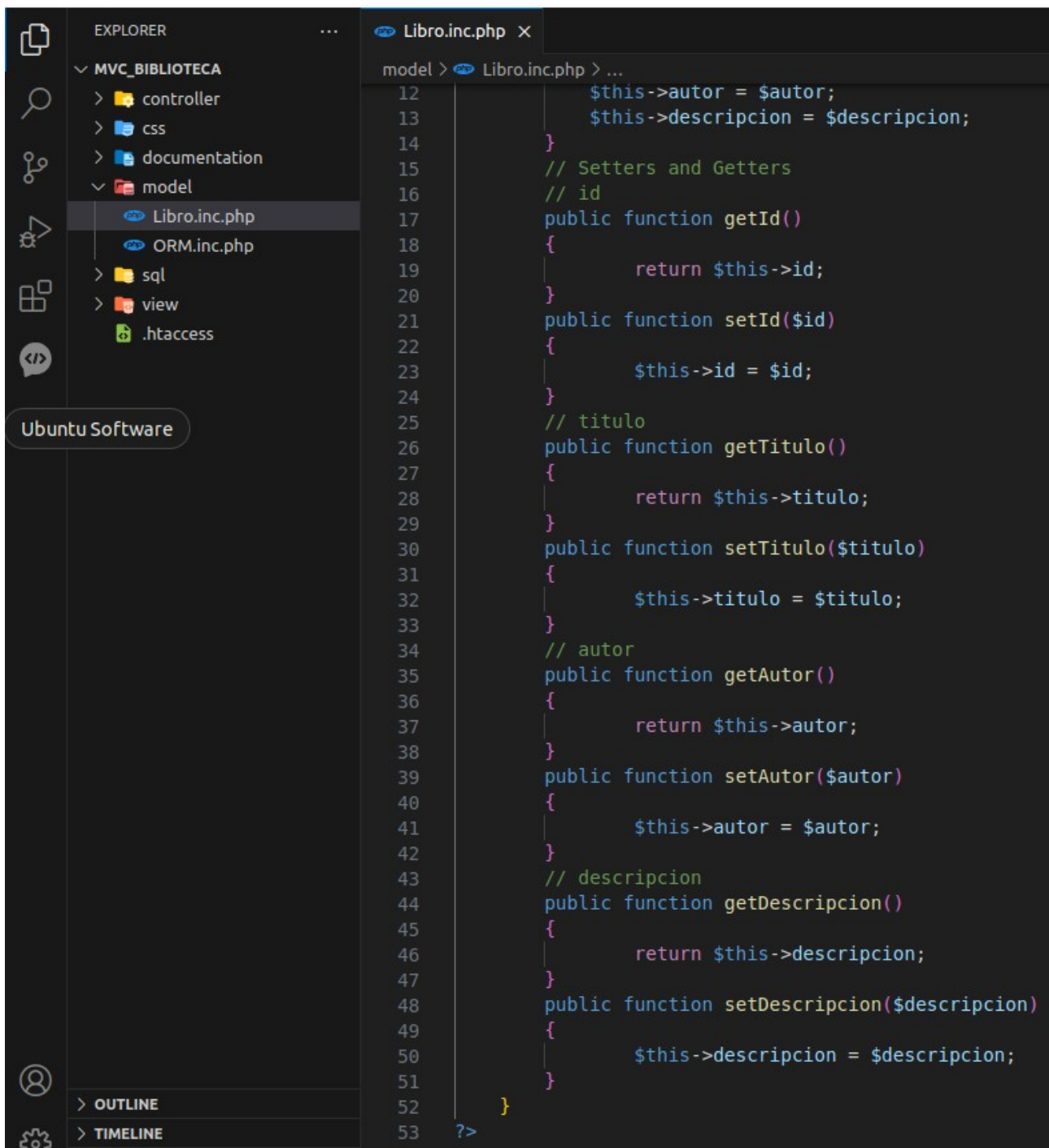
Dentro de esta carpeta hay dos ficheros uno que es Libro.inc.php que es una clase donde crearemos los objetos para almacenarlos en la base de datos y el otro fichero es ORM.inc.php que es el encargado de la conexión a la base de datos, realizar el cierre de la conexión, insertar un dato en la base de datos, actualizar un dato, eliminar un dato, mostrar todos los datos de la base de datos y mostrar un dato en específico.

En clase libro he decidido que los atributos sean protected para mayor seguridad y en la clase ORM private porque no se debería saber ni aunque se realice herencia. Porque así en ambos archivos solo puedes modificar mediante los métodos establecidos para una mayor protección e integridad de los datos.



En la clase Libro solo hay getters and setters para poder modificar los atributos mediante métodos controlados.

MVC Biblioteca ALV



The screenshot shows a code editor with the following structure:

- EXPLORER**
 - MVC_BIBLIOTECA
 - controller
 - css
 - documentation
 - model
 - Libro.inc.php (selected)
 - ORM.inc.php
 - sql
 - view
 - .htaccess

The **Libro.inc.php** file is open, showing the following PHP code:

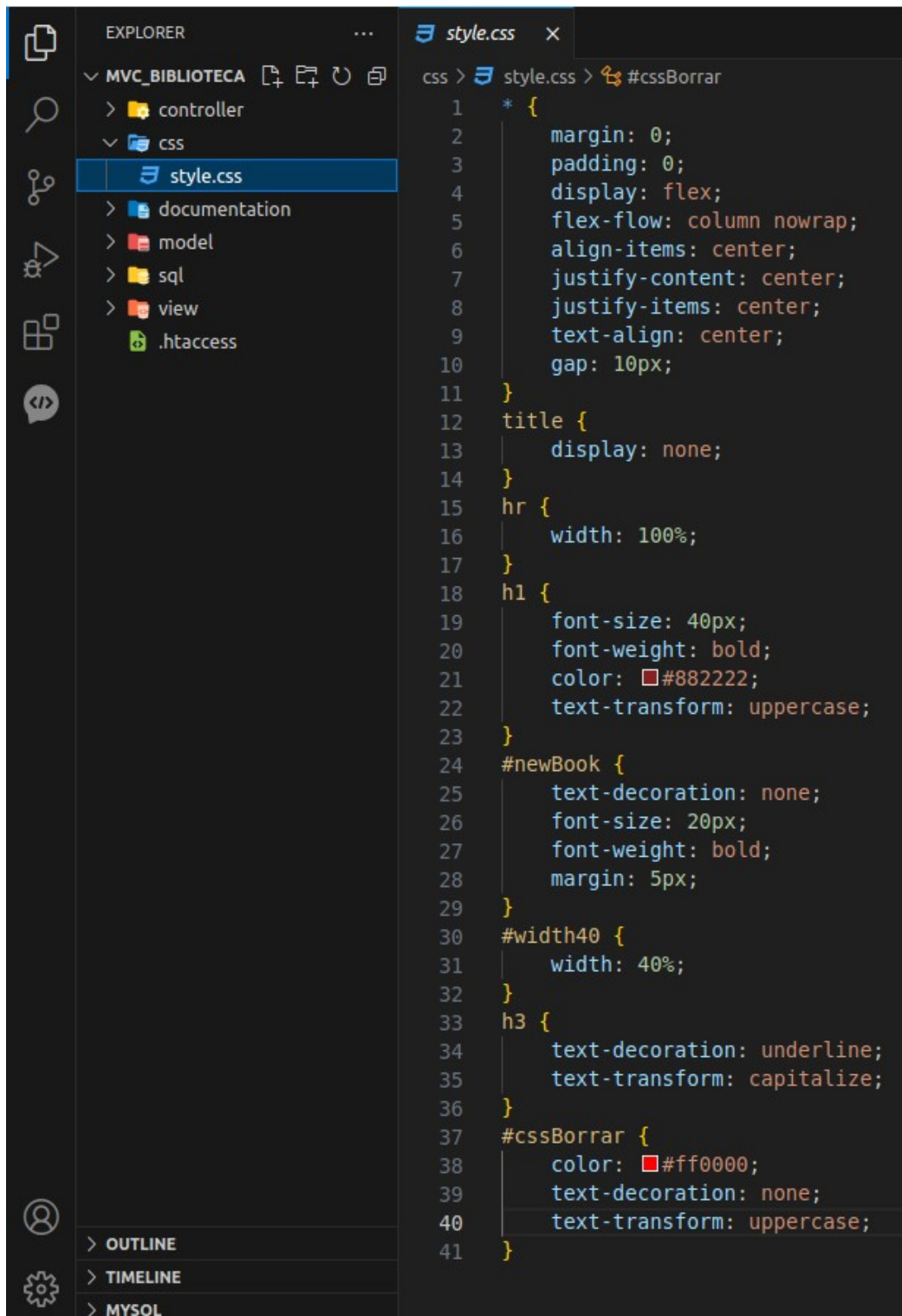
```
12     $this->autor = $autor;
13     $this->descripcion = $descripcion;
14 }
15 // Setters and Getters
16 // id
17 public function getId()
18 {
19     return $this->id;
20 }
21 public function setId($id)
22 {
23     $this->id = $id;
24 }
25 // titulo
26 public function getTitulo()
27 {
28     return $this->titulo;
29 }
30 public function setTitulo($titulo)
31 {
32     $this->titulo = $titulo;
33 }
34 // autor
35 public function getAutor()
36 {
37     return $this->autor;
38 }
39 public function setAutor($autor)
40 {
41     $this->autor = $autor;
42 }
43 // descripcion
44 public function getDescripcion()
45 {
46     return $this->descripcion;
47 }
48 public function setDescripcion($descripcion)
49 {
50     $this->descripcion = $descripcion;
51 }
52 }
53 ?>
```

Y en el ORM es un archivo bastante extenso que todos los métodos son importantes ya que son los que nos permite comunicarnos con la base de datos

MVC Biblioteca ALV

Carpeta css

La carpeta css contiene el archivo css para modificar el aspecto visual de la página web.



The screenshot shows a code editor with the Explorer view on the left and the style.css file open in the main editor. The Explorer view shows the project structure:

- MVC_BIBLIOTECA
 - controller
 - css
 - style.css
 - documentation
 - model
 - sql
 - view
 - .htaccess

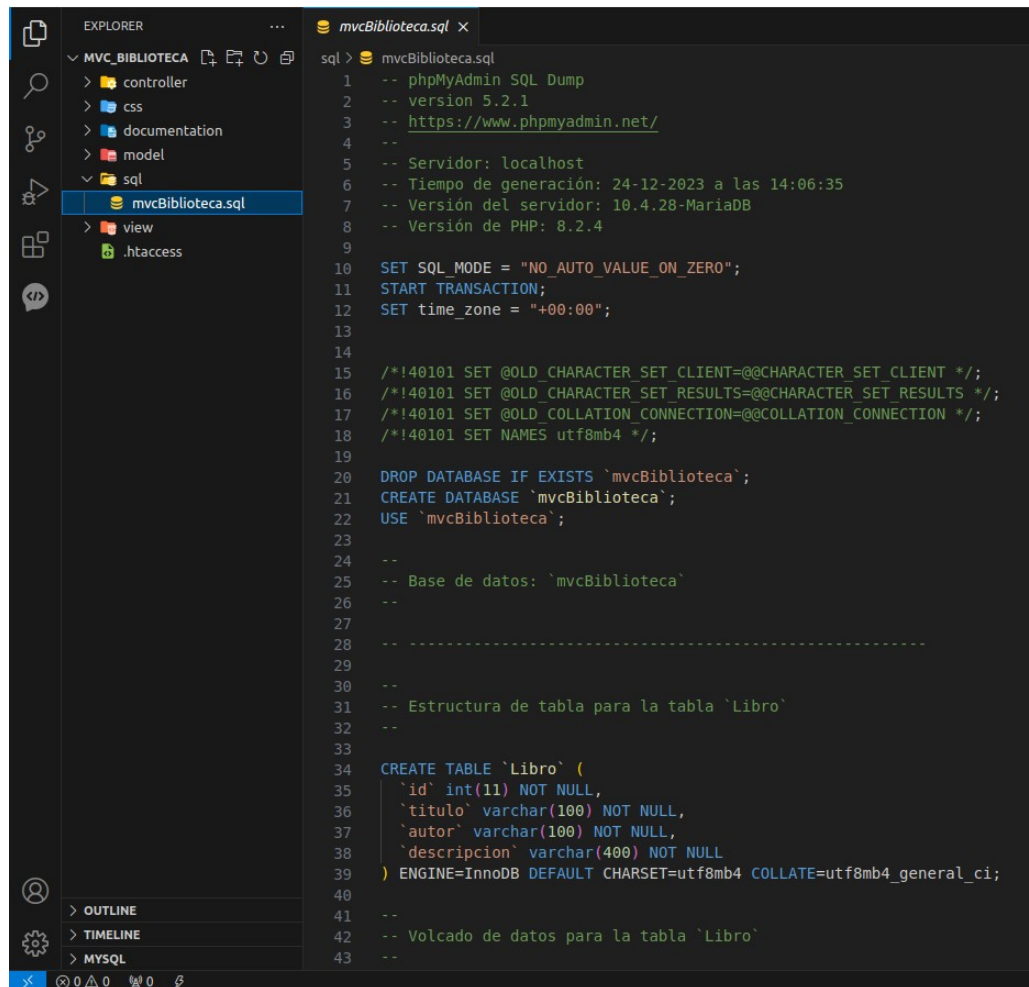
The style.css file content is as follows:

```
1 * {
2     margin: 0;
3     padding: 0;
4     display: flex;
5     flex-flow: column nowrap;
6     align-items: center;
7     justify-content: center;
8     justify-items: center;
9     text-align: center;
10    gap: 10px;
11 }
12 title {
13     display: none;
14 }
15 hr {
16     width: 100%;
17 }
18 h1 {
19     font-size: 40px;
20     font-weight: bold;
21     color: #882222;
22     text-transform: uppercase;
23 }
24 #newBook {
25     text-decoration: none;
26     font-size: 20px;
27     font-weight: bold;
28     margin: 5px;
29 }
30 #width40 {
31     width: 40%;
32 }
33 h3 {
34     text-decoration: underline;
35     text-transform: capitalize;
36 }
37 #cssBorrar {
38     color: #ff0000;
39     text-decoration: none;
40     text-transform: uppercase;
41 }
```

MVC Biblioteca ALV

Carpeta sql

La carpeta sql contiene el archivo con extensión sql para la carga inicial de datos de la base de datos, donde esta la creación de las tablas necesarias con los campos necesarios, las modificaciones de los parámetros necesarios como el auto increment en el id, la primary key y la inserción de un par de datos para una mejor visualización de la página web.



```
sql > mvcBiblioteca.sql
1  -- phpMyAdmin SQL Dump
2  -- version 5.2.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Servidor: localhost
6  -- Tiempo de generación: 24-12-2023 a las 14:06:35
7  -- Versión del servidor: 10.4.28-MariaDB
8  -- Versión de PHP: 8.2.4
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 DROP DATABASE IF EXISTS `mvcBiblioteca`;
21 CREATE DATABASE `mvcBiblioteca`;
22 USE `mvcBiblioteca`;
23
24 --
25 -- Base de datos: `mvcBiblioteca`
26 --
27
28 -- -----
29
30 --
31 -- Estructura de tabla para la tabla `Libro`
32 --
33
34 CREATE TABLE `Libro` (
35   `id` int(11) NOT NULL,
36   `titulo` varchar(100) NOT NULL,
37   `autor` varchar(100) NOT NULL,
38   `descripcion` varchar(400) NOT NULL,
39 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
40
41 --
42 -- Volcado de datos para la tabla `Libro`
43 --
```

Carpeta documentation

La carpeta documentation contiene esta documentación que estas leyendo.

4.0. Enlace al repositorio de GitHub

Enlace al repositorio de GitHub → https://github.com/Adrian-Leal-Vacas/MVC_BibliotecaALV

Realizado por: Adrián Leal Vacas