



**IES Luis Vélez de Guevara**  
**Dpto. de Informática**

## **SISTEMA DE GESTIÓN DE BIBLIOTECA EN LÍNEA CON PHP Y MYSQL**



**Manuel Moreno Casado**

**Horas de Libre Configuración**

**María de los Ángeles Rider Jiménez**

**2º ASIR I.E.S LUIS VÉLEZ DE GUÉVARA**

**CURSO: 2024/2025**

## Sumario

1. Introducción.....	3
2. Requisitos del proyecto.....	3
3. Estructura de la Base de Datos.....	4
4. Funcionalidades del Proyecto.....	9
5. Detalles técnicos.....	10
6. Conclusión.....	16

## 1. Introducción.

Este proyecto consistirá en una pagina web realizada con PHP que tiene como respaldo una base de datos, de gestión de biblioteca en línea, diseñada para facilitar la administración de libros , usuarios, préstamos y categorías. Está orientada a mejorar la experiencia tanto de los usuarios como de los administradores mediante una interfaz sencilla y eficaz. A través de esta plataforma, los administradores pueden gestionar libros, usuarios, préstamos, etc., mientras que los usuarios pueden buscar, pedir prestados y devolver libros de manera fácil y rápida.

Los distintos **objetivos** que perseguimos en esta pagina web son los siguientes:

- **Administración de Libros:** Los administradores pueden agregar, modificar y eliminar libros.
- **Gestión de Préstamos:** Los usuarios pueden pedir prestados libros y devolverlos en el plazo establecido.
- **Gestión de Usuarios:** Los administradores tienen la capacidad de ver los detalles de los usuarios, incluyendo sus préstamos actuales.
- **Seguridad y Autenticación:** Implementación de un sistema de login seguro para administradores y usuarios, con contraseñas cifradas.

## 2. Requisitos del proyecto.

Para realizar este proyecto vamos a utilizar las siguientes tecnologías:

- **Frontend:** HTML, CSS, JavaScript utilizando Bootstrap
- **Backend:** PHP
- **Base de datos:** MYSQL que nos va a permitir almacenar distinta información de libros, usuarios, préstamos, etc.
- **Seguridad:** En cuanto a la seguridad se han cifrado las contraseñas mediante password\_hash () y posteriormente en el login hemos utilizado password\_verify para realizar la comparación entre las contraseñas planas y las encriptadas.

En cuanto al entorno de desarrollo, para realizar el proyecto hemos utilizado

- Apache como **Servidor Web.**
- MySQL como **Base de Datos.**
- **PHP, HTML, SQL, CSS, JAVASCRIPT** como **Lenguajes**

### 3. Estructura de la Base de Datos

#### 3.1. Tablas principales

En cuanto a la estructura de la base de datos podemos decir que está organizada en varias tablas que nos van a permitir gestionar distintos aspectos del sistema.

Las tablas principales son las siguientes:

- **Usuarios:** Almacena la información de los usuarios, incluyendo su nombre, correo, tipo de usuario (admin o usuario normal), teléfono, dirección y contraseña encriptada.
- **Libros:** Contiene los detalles de los libros disponibles en la biblioteca, como título, autor, categoría, editorial y año de publicación.
- **Autores:** Información sobre los autores de los libros.
- **Editoriales:** Información sobre las editoriales que publican los libros.
- **Categorías:** Categorías de los libros para facilitar su búsqueda.
- **Prestamos:** Registra los préstamos realizados por los usuarios, con la fecha de préstamo y devolución.

#### 3.2. Relaciones entre las tablas

Las relaciones que podemos destacar entre las tablas anteriormente mencionadas, son las siguientes.

- Los libros están relacionados con los autores, editoriales y categorías mediante claves foráneas.
- Los préstamos están relacionados tanto con los libros como los usuarios.

#### 3.3. Creación de la base de datos con PHPMyAdmin

Para nuestro proyecto hemos utilizado como servidor web Apache , utilizando para ello el XAMPP que nos va a permitir gestionar bases de datos a través de MySQL. Esto nos va a permitir trabajar con trabajar con las tecnologías anteriormente mencionadas, lo que a su vez nos proporciona un entorno versátil.

Para la implantación de la base de datos, iniciamos el **XAMPP** y activamos tanto Apache como MySQL. Tal y como observamos en la siguiente figura:

## Sistema de gestión de biblioteca en línea con PHP y MYSQL Moreno Casado, Manuel

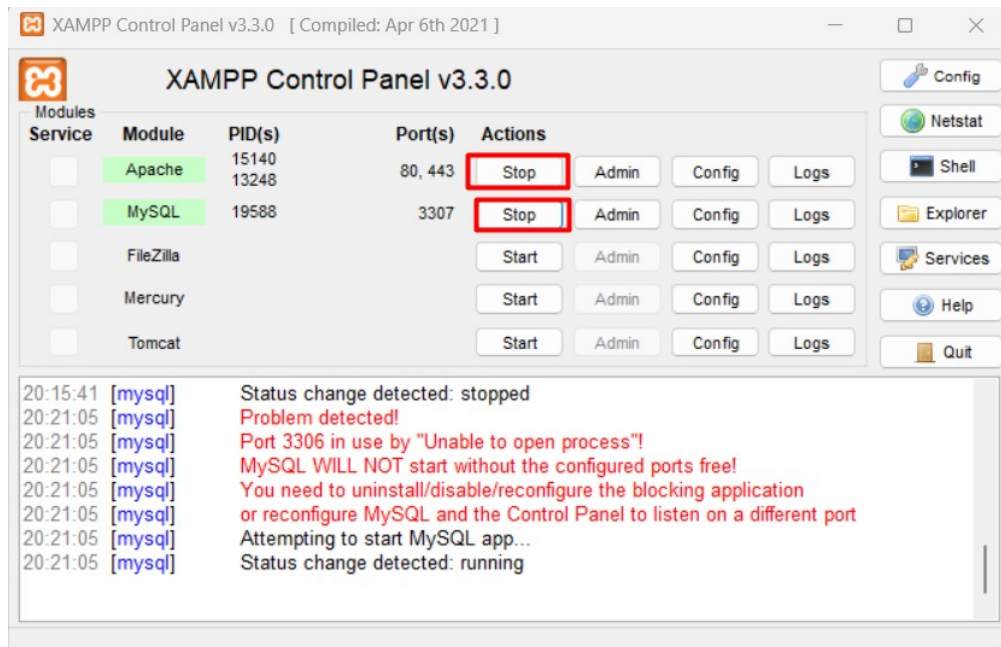


Figura 1

Una vez realizado lo anterior, nos dirigimos al navegador y escribimos localhost y posteriormente, **phpMyAdmin**.



### Welcome to XAMPP for Windows 8.2.4

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Figura 2

A la hora de crear la base de datos con sus correspondientes tablas, podemos realizarlo de dos maneras:

- **Insertar directamente el código MySQL** que hemos podido crear en MySQL Workbench. Para ello nos dirigimos al apartado de **SQL**. Tal y como podemos ver en la siguiente figura:

## Sistema de gestión de biblioteca en línea con PHP y MYSQL Moreno Casado, Manuel

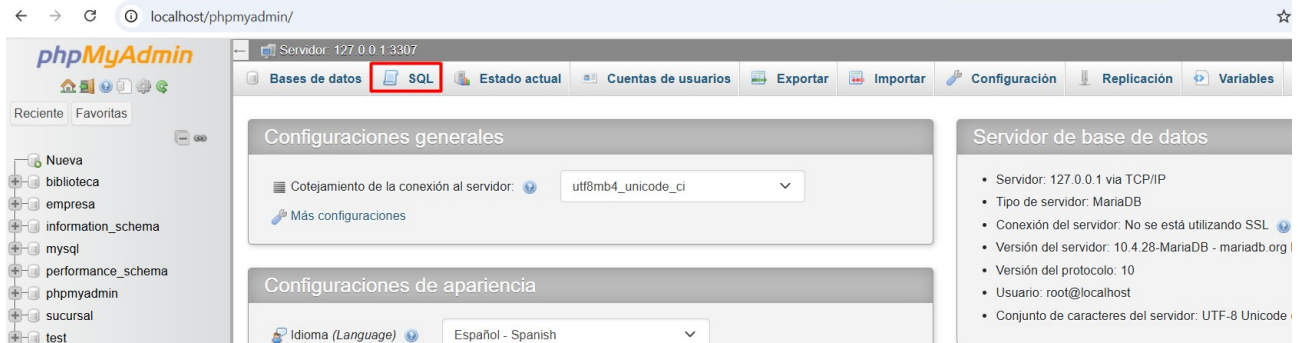


Figura 3

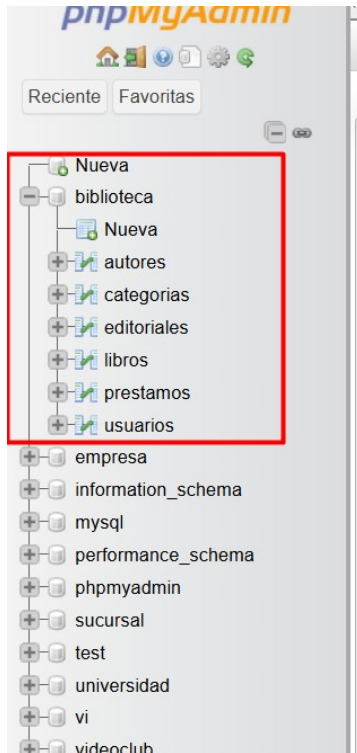
Seguidamente, nos saldrá el siguiente cuadro, en el que podremos introducir ya el código MySQL. Tal y como podemos observar a continuación. Para que la base de datos se cree debemos hacer clic en “Continuar”.



Figura 4

Finalmente, como podemos observar en el panel izquierdo la base de datos se habrá creado con sus correspondientes tablas y los datos que hayamos insertado.

## Sistema de gestión de biblioteca en línea con PHP y MYSQL Moreno Casado, Manuel



- A través de la **interfaz gráfica** que nos proporciona el propio phpMyAdmin. Para ello seguimos los siguientes pasos:

Clicamos en “**Nueva**” y posteriormente colocamos el nombre que deseemos, finalmente clicaremos en “**Crear**”.



Figura 5

Seguidamente, nos aparecerá la siguiente ventana en la que podremos crear las tablas que deseemos.

## Sistema de gestión de biblioteca en línea con PHP y MYSQL Moreno Casado, Manuel

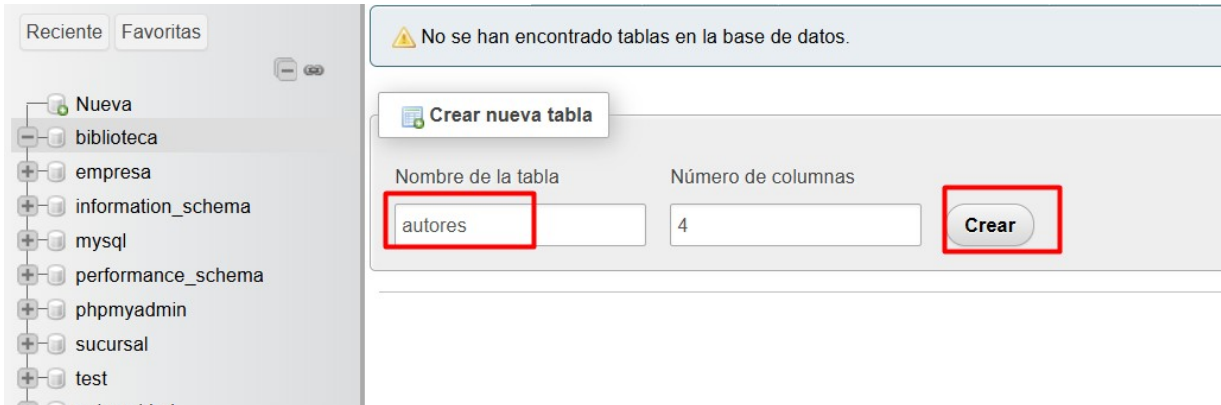
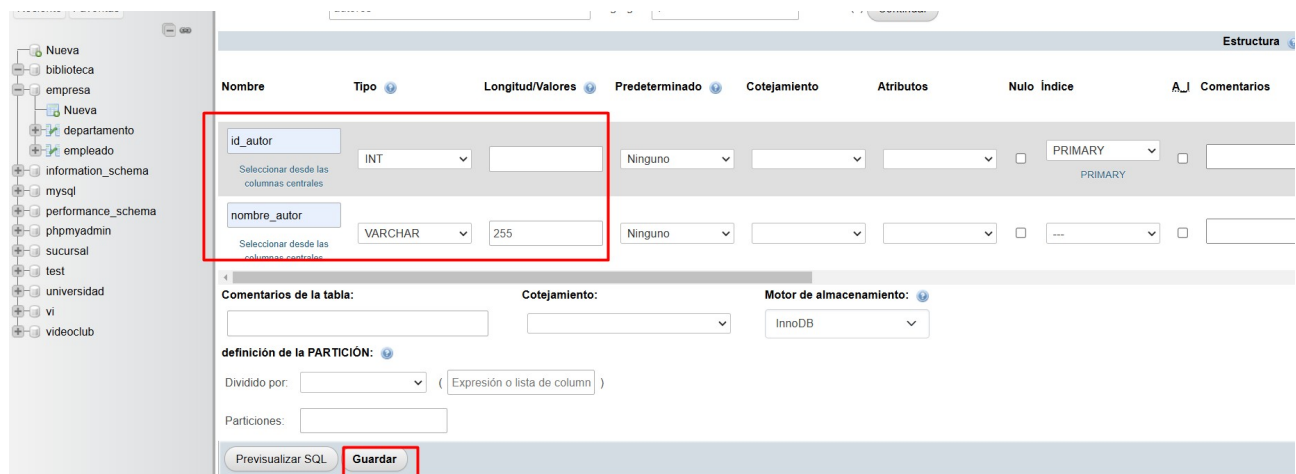
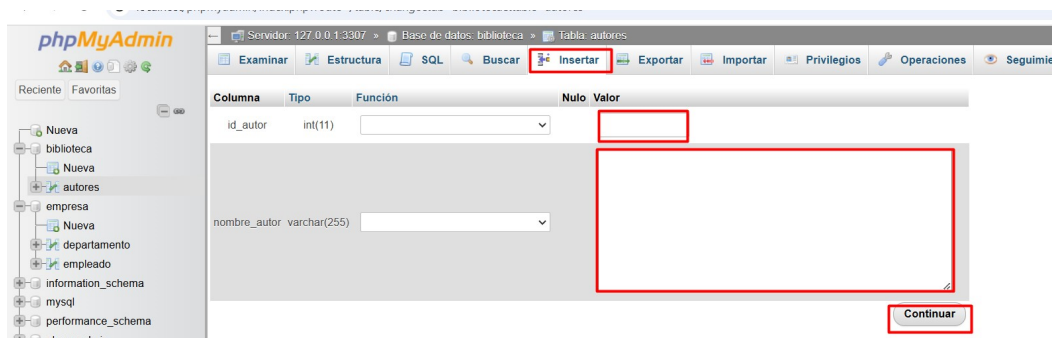


Figura 6

Una vez clicado en “**Crear**” podremos añadir los campos que deseemos a la tabla.



Finalmente, para insertar los datos en los distintos campos, nos dirigimos a la parte superior, más concretamente, realizaremos clic en “**Insertar**”. Aquí colocaremos los datos a los campos.





En nuestro caso la hemos exportado ya que previamente la hemos creado en MySQL Workbench (para visualizar la estructura y los datos nos podemos dirigir a los archivos de la pagina web).

## 4. Funcionalidades del Proyecto

En este punto vamos a desglosar las distintas funcionalidades que va a tener nuestra pagina web, haciendo una gran diferencia entre los usuarios y los administradores.

### 4.1. Funcionalidades para los usuarios

Las distintas funcionalidades para los usuarios son:

- **Iniciar sesión y registro:** Los usuarios pueden crear una cuenta y acceder a su perfil.
- **Visualización de libros:** Los usuarios pueden consultar una lista de libros disponibles, con detalles como título, autor y año de publicación.
- **Préstamos de libros:** Los usuarios pueden pedir prestados libros y devolverlos en un plazo determinado.
- **Historial de préstamos:** Los usuarios pueden ver los libros que han pedido prestados y su estado (prestado o devuelto).

### 4.2. Funcionalidades para los Administradores:

Las distintas funcionalidades para los administradores son:

- **Gestión de libros:** Los administradores pueden añadir, modificar o eliminar libros de la base de datos.
- **Gestión de usuarios:** Los administradores pueden visualizar los detalles de los usuarios y sus préstamos activos.
- **Gestión de préstamos:** Los administradores pueden ver quién tiene qué libros prestados y cuándo deben devolverlos.
- **Generación de contraseñas:** Los administradores pueden generar contraseñas seguras para los usuarios, las cuales se almacenan de manera encriptada en la base de datos.

En conclusión, en nuestro proyecto nos vamos a encontrar una multitud de funcionalidades básicas tanto para el usuario como para el administrador.

## 5. Detalles técnicos.

Nuestra pagina web estará compuesta por los siguientes archivos:

**Index.php:** Nos muestra la pagina de inicio, con una barra de navegación, incluyendo dos botones uno para iniciar sesión y otro para registrarse.

```
85 <!-- Barra de navegación -->
86 <nav class="navbar navbar-expand-lg navbar-dark">
87   <div class="container-fluid">
88     <a class="navbar-brand" href="#">Biblioteca</a>
89   </div>
90 </nav>
91
92 <!-- Contenedor principal -->
93 <div class="container">
94   <div class="title">Bienvenido a la Biblioteca</div>
95
96   <!-- Mensaje de bienvenida -->
97   <p class="text-center" style="font-size: 1.1rem; color: #fff;">
98     ¡Nos alegra que estés aquí! Puedes iniciar sesión o registrarte para comenzar a gestionar tus libros.
99   </p>
100
101   <!-- Botones para iniciar sesión o registrarse -->
102   <a href="login.php" class="btn btn-primary">Iniciar sesión</a>
103   <a href="registro.php" class="btn btn-success">Registrarse</a>
104 </div>
105
106 </body>
107 </html>
108
```

Ilustración 1

**Conexión.php:** Establece la conexión con la base de datos.

Se definen los siguientes datos necesarios para conectarse a MySQL:

- **Servidor:** localhost : 3307 (indica que MySQL está corriendo en el puerto 3307).
- **Usuario:** root (usuario predeterminado en XAMPP).
- **Contraseña:** "" (vacía por defecto en XAMPP).
- **Base de datos:** "biblioteca" (el nombre de la base de datos a la que queremos conectarnos).

Para establecer la conexión se usa la función `mysqli_connect()` para conectarse a MySQL con los datos definidos antes.

Si la conexión se establece correctamente, `$conexion` guardará la conexión activa.

```
1  <?php
2  // Variables de conexión
3  $servidor = "localhost:3307"; // Nombre del servidor (por defecto localhost)
4  $usuario = "root";           // Usuario (por defecto root en XAMPP)
5  $contrasena = "";            // Contraseña (por defecto vacío en XAMPP)
6  $nombre_base_datos = "biblioteca"; // Nombre de la base de datos
7
8
9  // Establecer la conexión
10 $conexion = mysqli_connect($servidor, $usuario, $contrasena, $nombre_base_datos);
11
12 // Verificar si la conexión fue exitosa
13 if (!$conexion) {
14     die("Error de conexión al servidor MySQL: " . mysqli_connect_error());
15 }
16
17 // Verificar si la base de datos se ha seleccionado correctamente
18 if (!mysqli_select_db($conexion, $nombre_base_datos)) {
19     die("Error al seleccionar la base de datos: " . mysqli_error($conexion));
20 }
```

Ilustración 2

**Login.php:** Muestra el formulario de inicio de sesión para los usuarios.

### 1. Iniciar sesión y conectar a la base de datos

- Se inicia la sesión con `session_start()`; para poder usar `$_SESSION`.
- Se incluye el archivo `conexion.php`, que contiene la conexión a la base de datos.

### 2. Verificar si el formulario fue enviado

Se usa `$_SERVER["REQUEST_METHOD"] == "POST"` para comprobar si se enviaron datos. Se capturan el correo (usuario), la contraseña (password) y si se seleccionó "Recordar sesión" (recordar).

### 3. Consultar el usuario en la base de datos

Se prepara una consulta segura (`mysqli_prepare`) para buscar al usuario por su correo. Se ejecuta y obtiene el resultado. Si hay un usuario con ese correo, se extrae su ID, nombre, tipo y contraseña encriptada.

### 4. Verificar la contraseña

Se usa `password_verify($password, $hash_password)` para comparar la ingresada con la almacenada.

Si la contraseña es correcta:

Se guardan el nombre, tipo de usuario e ID en `$_SESSION`.

Si el usuario marcó "Recordar sesión", se guarda una cookie con su correo (dura 30 días).

Si no, se elimina la cookie si existía.

Se redirige a **admin.php** (si es administrador) o **usuario.php** (si es usuario normal).

```
1  <?php
2  session_start();
3  include('conexion.php'); // Conexión a la base de datos
4
5  if ($_SERVER["REQUEST_METHOD"] == "POST") {
6      $usuario = $_POST['usuario']; // El correo electrónico
7      $password = $_POST['password']; // La contraseña en texto plano
8      $recordar = isset($_POST['recordar']); // Verificamos si el usuario seleccionó "Recordar sesión"
9
10     // Ejecutar consulta de manera segura con SQL preparado
11     $query = "SELECT id_usuario, nombre, tipo_usuario, password FROM Usuarios WHERE correo = ?";
12     $stmt = mysqli_prepare($conexion, $query);
13     mysqli_stmt_bind_param($stmt, "s", $usuario);
14     mysqli_stmt_execute($stmt);
15     $resultado = mysqli_stmt_get_result($stmt);
16
17     if ($resultado->num_rows == 1) {
18         $fila = mysqli_fetch_assoc($resultado);
19         $hash_password = $fila['password']; // Contraseña encriptada de la BD
20
21         // Verificar la contraseña encriptada con password_verify
22         if (password_verify($password, $hash_password)) {
23             $_SESSION['usuario'] = $fila['nombre']; // Guardar el nombre del usuario en sesión
24             $_SESSION['tipo_usuario'] = $fila['tipo_usuario']; // Guardar el tipo de usuario en sesión
25             $_SESSION['id_usuario'] = $fila['id_usuario']; // Guardar el ID del usuario en sesión
26         }
```

*Ilustración 3*

**Logout.php:** Este archivo nos va a permitir cerrar la sesión del usuario y redirigir a **login.php**.

session\_start(); → Inicia la sesión.

session\_unset(); → Borra todas las variables de sesión.

session\_destroy(); → Elimina la sesión completamente.

header("Location: login.php"); → Redirige al usuario al login.

exit(); → Finaliza la ejecución del script.

```
logout.php
1  <?php
2  session_start();
3  session_unset(); // Eliminar todas las variables de sesión
4  session_destroy(); // Destruir la sesión
5  header("Location: login.php"); // Redirigir al login
6  exit();
7  ?>
8
```

Ilustración 4

**Registro.php:** Este archivo nos va a permitir registrar usuarios en la base de datos con validación y encriptación de contraseña.

- Conexión a la BD (include('conexion.php')).
- Captura los datos del formulario (\$\_POST).
- Verifica si el correo ya existe en la BD.
- Si no existe, encripta la contraseña (password\_hash()) y guarda el usuario.
- Mensajes de éxito o error según el resultado.
- Cierra la conexión (mysqli\_close(\$conexion)).

```
registro.php > ...
<?php
session_start();
include('conexion.php'); // Conexión a la base de datos

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtener los datos del formulario
    $nombre = $_POST['nombre'];
    $correo = $_POST['correo'];
    $telefono = $_POST['telefono'];
    $direccion = $_POST['direccion'];
    $password = $_POST['password'];

    // Verificar que el correo no esté registrado
    $sql_verificar = "SELECT * FROM Usuarios WHERE correo = '$correo'";
    $resultado_verificar = mysqli_query($conexion, $sql_verificar);

    if (mysqli_num_rows($resultado_verificar) > 0) {
        $error = "Este correo electrónico ya está registrado.";
    } else {
        // Encriptar la contraseña
        $password_hash = password_hash($password, PASSWORD_DEFAULT);

        // Insertar el nuevo usuario en la base de datos
        $sql_insertar = "INSERT INTO Usuarios (nombre, correo, telefono, direccion, password, tipo_usuario)
        VALUES ('$nombre', '$correo', '$telefono', '$direccion', '$password_hash', 'usuario')";

        if (mysqli_query($conexion, $sql_insertar)) {
            $mensaje = "Registro exitoso. Ahora puedes iniciar sesión.";
        } else {
            $error = "Error al registrar el usuario. Intenta nuevamente.";
        }
    }
}

// Cerrar conexión
mysqli_close($conexion);
?>
```

Ilustración 5

**Admin.php:** Este archivo nos va a permitir gestionar libros en una biblioteca.

- **Autenticación:** Verifica si el usuario es admin antes de mostrar la página. Si no lo es, redirige a login.

- **Consulta de Libros:** Obtiene todos los libros de la base de datos y muestra su título, autor, ISBN y año.

- **Interfaz:** Muestra una tabla con los libros y botones para editar y eliminar. También incluye un botón para añadir un nuevo libro.

- **Mensajes:** Muestra alertas de éxito o error si hay mensajes en la sesión.

```
1  <?php
2  session_start();
3  include('conexion.php');
4
5  // Verificar que el usuario esté autenticado como admin
6  if (!isset($_SESSION['usuario']) || $_SESSION['tipo_usuario'] != 'admin') {
7      header("Location: login.php");
8      exit();
9  }
10
11 $usuario = $_SESSION['usuario'];
12
13 // Obtener todos los libros
14 $sql_libros = "SELECT * FROM Libros";
15 $resultado_libros = mysqli_query($conexion, $sql_libros);
16 ?>
17
```

Ilustración 6

```
17  <?php
18  if (isset($_SESSION['mensaje'])) {
19      $clase_alerta = strpos($_SESSION['mensaje'], 'Error') !== false ? 'alert-danger' : 'alert-success';
20      echo ' <div class="alert ' . $clase_alerta . ' text-center" role="alert"> ' . $_SESSION['mensaje'] . ' </div>';
21      unset($_SESSION['mensaje']);
22  }
23  ?>
24
25  <a href="introducirLibro.php" class="btn btn-success mb-4">Añadir Nuevo Libro</a>
26
27  <table>
28      <thead>
29          <tr>
30              <th>Título</th>
31              <th>Autor</th>
32              <th>ISBN</th>
33              <th>Año</th>
34              <th>Acciones</th>
35          </tr>
36      </thead>
37      <tbody>
38          <?php while ($libro = mysqli_fetch_assoc($resultado_libros)) { ?>
39              <tr>
40                  <td><?php echo $libro['titulo']; ?></td>
41                  <td>
42                      <?php
43                      $sql_autor = "SELECT nombre_autor FROM Autores WHERE id_autor = " . $libro['id_autor'];
44                      $resultado_autor = mysqli_query($conexion, $sql_autor);
45                      $fila_autor = mysqli_fetch_assoc($resultado_autor);
46                      echo $fila_autor['nombre_autor'];
47                      ?>
48                  </td>
49              </tr>
50          </tbody>
51      </table>
52
```

Ilustración 7

Dentro de este archivo tenemos **modificarLibro.php**, **introducirLibro.php**, **eliminarLibro.php** y **verPrestamos.php**

Por otro lado, tenemos **usuario.php** que es donde los usuarios de una biblioteca pueden ver los libros disponibles para alquilar.

El código está formado por:

### 1. Autenticación del Usuario

- Inicia sesión con `session_start()`.
- Verifica si el usuario está autenticado (`$_SESSION['usuario']`).
- Si no está autenticado, lo redirige a `login.php`.
- Guarda el nombre del usuario en `$usuario`.
- Si el usuario no es administrador, guarda su ID en `$id_usuario` (para filtrar libros que ya ha alquilado).

### 2. Consulta a la Base de Datos

Se obtiene una lista de libros disponibles con la información de:

- **Título**
- **ISBN**
- **Año de publicación**
- **Autor**
- **Categoría**
- **Editorial**
- Se excluyen libros que el usuario ya tiene prestados (estado = 'Prestado' en la tabla Prestamos).

Dentro de `usuario.php` tenemos **librosAlquilados**, **AlquilarLibro.php** y **devolverLibro.php**

```
<?php
session_start();
include('conexion.php');

// Verificar que el usuario esté autenticado
if (!isset($_SESSION['usuario'])) {
    header("Location: login.php");
    exit();
}

$usuario = $_SESSION['usuario'];
$id_usuario = $_SESSION['tipo_usuario'] == 'admin' ? null : $_SESSION['id_usuario'];

// Obtener los libros disponibles con información de autor, categoría y editorial
$sql_libros_disponibles = "
    SELECT l.id_libro, l.titulo, l.isbn, l.año_publicacion, a.nombre_autor, c.nombre AS categoria, e.nombre AS editorial
    FROM Libros l
    JOIN Autores a ON l.id_autor = a.id_autor
    JOIN Categorías c ON l.id_categoria = c.id_categoria
    JOIN Editoriales e ON l.id_editorial = e.id_editorial
    WHERE l.id_libro NOT IN (
        SELECT id_libro FROM Prestamos
        WHERE id_usuario = '$id_usuario' AND estado = 'Prestado'
    )
";

$resultado_libros_disponibles = mysqli_query($conexion, $sql_libros_disponibles);
?>

<!doctype html>
<html lang="es">
```

*Ilustración 8*

## 6. Conclusión

Este proyecto ofrece una forma fácil y segura de administrar una biblioteca en línea. Su diseño simple y sus funciones bien organizadas permiten que tanto los administradores como los usuarios puedan gestionar y pedir prestados libros sin complicaciones. Además, utiliza medidas de seguridad como la encriptación de contraseñas y consultas protegidas para evitar problemas y garantizar que la información esté segura.

## 7. Posibles mejoras

A nuestro proyecto se le pueden añadir muchas mejoras, entre las que creo mas necesarias:

- **Integración de búsqueda avanzada:** Agregar filtros de búsqueda por autor, categoría, editorial, etc.
- **Sistema de reservas:** Permitir que los usuarios reserven libros que están prestados.