

Tarea Unidad 5: Diseño de una Aplicación para la Gestión de una biblioteca

El objetivo es desarrollar una aplicación web que permita la **gestión de una biblioteca** digital de libros electrónicos. Para ello, trabajaremos con la **tabla libros**, la cual formará parte de una nueva base de datos. Podemos reutilizar el usuario administrador de ejercicios anteriores con los permisos requeridos en la base de datos.

La estructura de la tabla libros y los scripts necesarios para la creación de la base de datos serán facilitados a través de un archivo `tabla_libros.sql`.

En la tabla libros guardaremos la siguiente información para cada libro:

- Título del libro
- Autor
- Género (por ejemplo, Novela, Ciencia Ficción, Ensayo, etc.)
- Fecha de publicación
- ISBN
- Tamaño en Kb
- Número de páginas (opcional)

Se utilizará el programa **login.php** para acceder a la aplicación de biblioteca, tendrá un formulario que nos pide un usuario y contraseña y accede a la **tabla de usuarios** donde valida que sea un usuario autorizado. La contraseña se almacenará utilizando la codificación SHA256. Si el usuario es correcto se crea la **sesión** para ese usuario y podrá ejecutar la aplicación de biblioteca.

Nota: Como no se implementa un sistema para cargar usuarios, lo haremos directamente sobre la bbdd, podéis utilizar esta herramienta para generar la clave con SHA256 <https://emn178.github.io/online-tools/sha256.html>.

Si en el acceso al **index.php** se detecta que no existen datos en la tabla de libros se mostrará un botón de instalación, **instalación.php**, donde utilizando Faker se hará la inserción automática de 10 libros.

Una vez que existan libros, o el caso que ya existiesen, se cargará el listado de la biblioteca, **libros.php**. Desde el listado de libros tendremos un botón para insertar nuevos libros a través de **fcrear.php**.

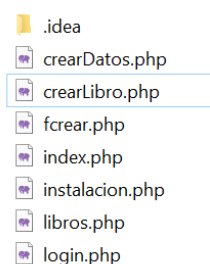
Todas las páginas utilizarán el sistema de plantillas **Blade**.

Dependencias obligatorias para instalar con Composer:

- philo/laravel-blade como motor de plantillas.
- fakerphp/faker para generar datos aleatorios de prueba.
- autoload con optimizador.

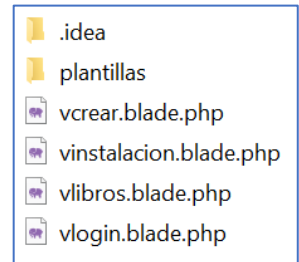
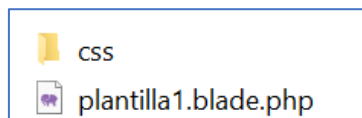
Estructura sugerida del proyecto:

1. cache: Carpeta necesaria para el funcionamiento de Blade, con permisos "777" en Linux.
2. public: Aquí se almacenan las páginas que se muestran en el navegador.
 - crearDatos.php: Página que genera datos de ejemplo para los libros.
 - crearLibros.php: Página "action" del formulario para crear un nuevo libro, con validaciones para evitar duplicados y campos vacíos.



- fcrear.php: Formulario para crear el libro. Usará la vista vcrear.blade.php.
 - login.php: página de login
 - index.php: Página de inicio que verifica si la tabla libros tiene datos y, de no ser así, redirige a instalacion.php para generar datos de ejemplo.
 - instalacion.php: Carga la vista vinstalacion.php que contiene un botón para generar datos de ejemplo.
 - libros.php: Muestra una lista de los libros en una tabla.
3. sql: Contendrá el archivo .sql.
 4. src: Contiene las clases para gestionar la base de datos:
 - Conexion.php: Establece la conexión con la base de datos.
 - Libro.php: Gestiona la tabla libros, con los métodos para devolver los libros, insertar, etc.
 5. vendor: Generada por Composer.
 6. views: Contiene todas las vistas y plantillas utilizadas.

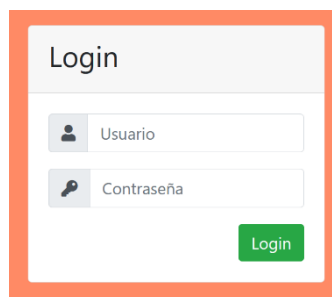
Dentro de la carpeta plantilla tenemos la plantilla de nivel superior que todas las demás incluyen:



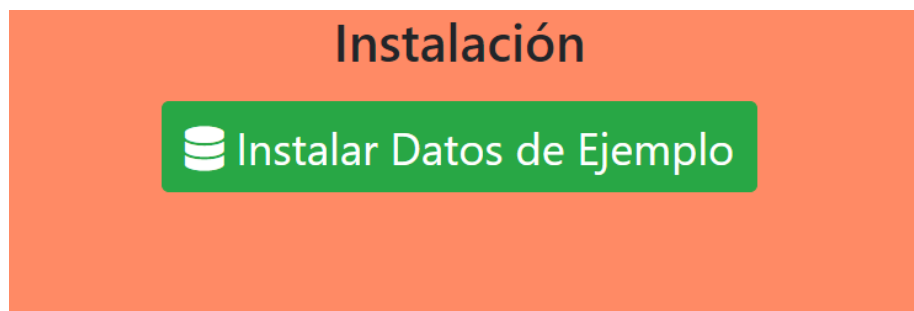
El archivo composer.json estará presente con las dependencias del proyecto.

Pantallas:

Login.php



Instalación.php



Listado de Libros						
+ Nuevo Libro						
Título	Autor	Género	Fecha de Publicación	ISBN	Número de Páginas	Tamaño del Libro
Ipsa magnam facilis.	Furman Luetngen	Ciencia Ficción	2021-08-05	9794720371687	197	2.5MB
Est quo quod.	Loyal Aufderhar	Ciencia Ficción	2007-07-23	9793078976292	420	2.5MB

Crear Libro

Título

Autor

Género

Fecha de Publicación

ISBN

Número de Páginas

Tamaño del Libro

Crear

Limpiar

Volver

Calificación

- Crea la estructura correcta del proyecto y se instalan correctamente las dependencias con Composer. 1.5 Puntos.
- Se crean correctamente, con los métodos necesarios, las clases "Conexion" y "Libros" y se hace uso de "namespaces" y herencia. 1.5 Puntos.
- Crea correctamente los datos de ejemplo con "phpfaker/faker". 1 Punto.
- Las páginas de la carpeta "public" cargan las vistas apropiadamente y les pasan los parámetros necesarios. 1 Punto.
- El formulario para crear un libro funciona correctamente, controlando posibles errores. 1 Punto.
- Las vistas hacen uso de una plantilla común. 0.5 Puntos.
- Se realiza el login codificando la contraseña con SHA256, se gestiona la sesión para entrar en la página de inicio. 1.5 punto.
- La página de inicio comprueba si existen datos o no en la tabla "libros" y hace las redirecciones adecuadas. 0.5 Puntos.
- Se hace uso correcto de las sesiones para mostrar mensajes. 0.5 Puntos.
- El diseño es adecuado y el código está comentado. 1 Punto