



Project carried out by: Aleksander Trujillo Prokhorenko

Course: 2 DAW B

Year: 2023/2024

Project general description:

El proyecto está pensando en realizar una aplicación capaz de gestionar una biblioteca, la parte de registro de nuevos usuarios que alquilan libros, también registrar la nueva entradas de libros y autores a la biblioteca además de poder gestionar quien tiene un libro alquilado y cual.

Las tecnologías usadas en este proyecto fue el [Framework Laravel](#) del lenguaje de programación de [PHP](#), dentro de laravel usamos [Livewire](#) para hacer frontales reactivos y [Carbon](#) para el manejo de fechas.

Para la gestion de la base de datos, [phpMyAdmin](#) de [XAMP](#)

Versiones usadas:

- Laravel -> 10x
- PHP -> PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64)
- Composer -> version 2.6.6 2023-12-08 18:32:26
- XAMP -> 8.0.30 / PHP 8.0.30

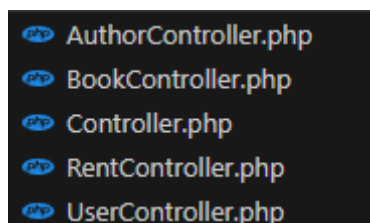
Project Structure:

Al usar laravel, el proyecto está basado en la arquitectura [MVC](#), por lo tanto trabajamos con Modelos, Controladores y Vistas.

Controladores:

ruta: App/Http/Controllers/

Dentro de esa carpeta encontraremos estos ficheros:



Están creados con la sintaxis de laravel para que los vincule directamente al modelo y poder evitar problemas de rutas.

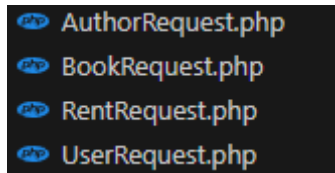
Estos controladores, controlan dos rutas exactamente, estas son las de creación y las de inserción de datos en la base de datos.

La de inserción de datos en la base de datos usa un filtro para validar los datos y evitar tener datos erróneos en nuestro sistema.

Validaciones:

ruta: App/Http/Requests/

Dentro de esa carpeta encontraremos estos ficheros:



Estos ficheros los usaremos para la validación de campos en el formulario, para evitar tener problemas en la base de datos.

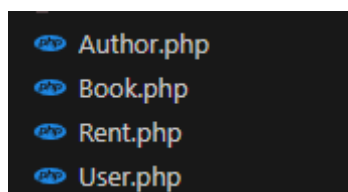
No creamos las validaciones en el controlador porque ese no es su trabajo, por lo tanto lo creamos como [CustomRequest](#), en la cual definimos los parámetros de cómo queremos que sean estos campos.

Estos ficheros tienen dos funciones, una que devuelve true, que es la que autoriza que podamos usar la validación, y la otra función devuelve como tienen que validarse estos campos.

Modelos:

ruta App/Models/

Dentro de esa carpeta encontraremos estos ficheros:



Estos son los encargados de gestionar qué datos se pueden modificar, cuales no y cuales ocultar de cara a la red, mediante las variables \$fillable, \$guarded, \$hidden.

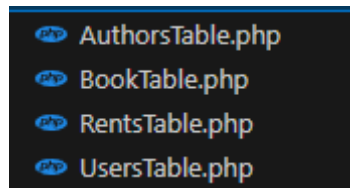
Además son los encargados de relacionar los modelos entre sí, es decir, si una tabla tiene una relación 1 a N como en el caso de libros y alquileres, son los encargados de hacerlo mediante una función hasMany y BelongsTo

LÓGICA REACTIVA

Livewire

ruta App/Livewire/

Dentro de esa carpeta encontraremos estos ficheros:



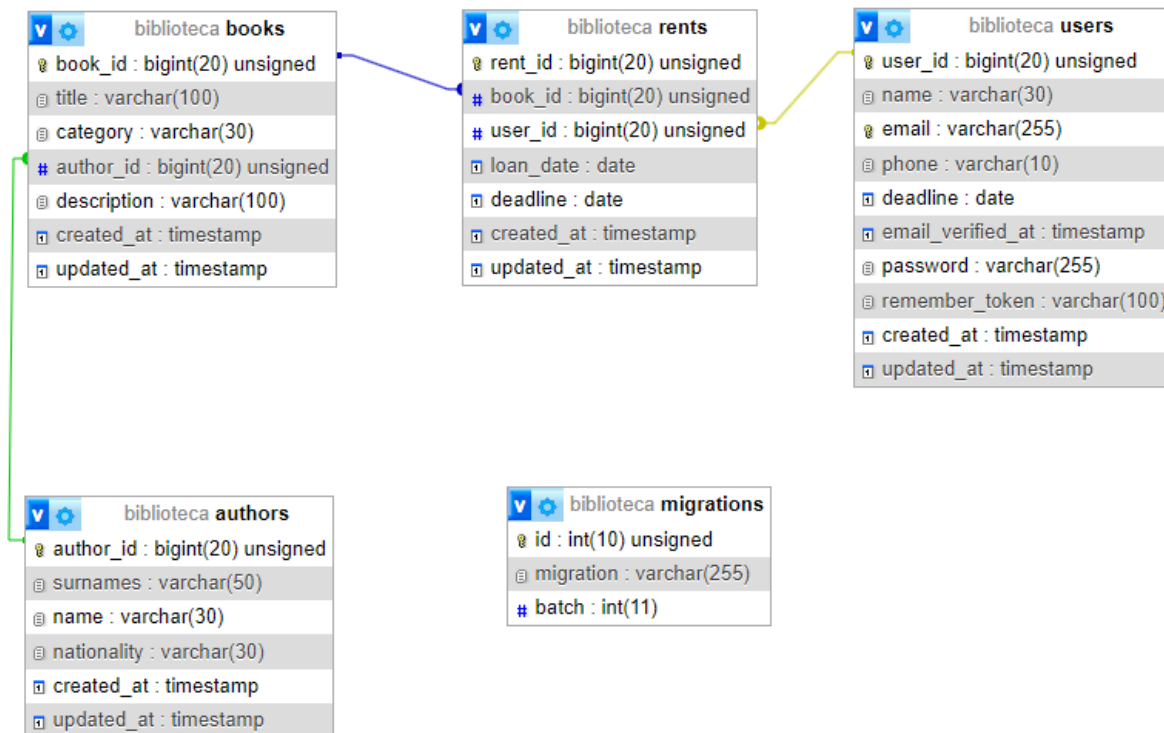
Estos ficheros son los encargados de realizar que la aplicación sea reactiva, ya que cargamos las vistas desde ellos, al igual que los procesos de edición y borrado se hacen sin necesidad de refrescar el navegador a tiempo real.

Esto es posible ya que el paquete de livewire de laravel usa peticiones asíncronas en segundo plano sin que tengamos que hacer nada, usando una sintaxis similar a Vue.

Estos ficheros están estructurados en la siguientes funciones:

- `mount()` -> Hook que se activa cuando se renderiza la página, lo usamos para inicializar todas las variables
- `render()` -> carga la vista con todos los datos de la base de datos, para su manipulación
- `edit(id)` -> obtenemos el id del campo que queremos modificar y implementamos las variables del componente con estos
- `update()` -> obtengo el objeto que queremos modificar con el id anterior, posteriormente validamos los campos y si son correctos los modificamos
- `resetInput()` -> se encarga de limpiar todos los campos después de hacer un update para evitar tener problemas.
- `rules()` -> encarga de hacer la validación de todos los datos, es una forma de hacerlo en livewire, son como los Custom Request.

DATA BASE



La aplicación cuenta con 5 tablas, pero realmente solo vamos a usar 4, que son Autores, Alquileres, Libros, Usuarios, la quinta que no usamos es la de migraciones que crea laravel por cada vez que realizamos una modificación.

RELACIONES ENTRE TABLAS:

- Libros N — 1 Autor
- Libros 1 — 1 Alquileres
- Usuarios 1 — N Alquileres

El conector que usamos entre la aplicación y laravel lo encontramos en el archivo .env que está en el directorio raíz de la aplicación, este archivo es un archivo en el que están todas las variables de entorno, entre ellas los diferentes conectores a gestores de base de datos.

En nuestro caso al usar XAMP, estamos usando [MySQL](#) por lo tanto lo en la variable de Mysql le indicamos el nombre de la base de datos, usuario y contraseña.

RUTAS:

Nuestras rutas para realizar cualquier operación sobre estas tablas están controladas por el archivo WEB.PHP, que es el encargado, y es donde indicaremos qué url queremos que tengan en el navegador, el controlador que usaremos y la función para dicha ruta que lo gestiona.

Este fichero se encuentra en:

ruta: Raiz/routes/

Para una mayor usabilidad de estas rutas todas tienen sus alias asignados con ->name, para una mayor escalabilidad o por si se realiza alguna modificación en la url de la ruta así evitamos tener que cambiarlo manualmente en todos lados.

VISTAS:

Todas las vistas de la aplicación se encuentran en:

ruta: Raiz/resources/views/

Dentro de esto nos encontramos con diferentes carpetas:

1. Carpeta layout:

Aquí encontraremos los componentes de nuestra aplicación, entre ellos se encuentra nuestro HTML padre del cual heredan todas las vistas su estructura, este se llama **app.blade.php**, en él encontramos una estructura de HTML 5 básica con diferentes secciones como el título, body marcadas con la directiva @yield, para poder usarlo en otras vistas y indicar que queremos que nuestro contenido está dentro de esa sección.

1.1 Carpeta layout>forms:

En esta carpeta encontraremos todos los formularios de inserción de datos en nuestra aplicación que extienden de **app.blade.php**, todos estos formularios tiene la directiva @csrf para aumentar la seguridad de la aplicación y evitar que nos lleguen datos desde otro sitio que no sea el nuestro, es un token de laravel formado de manera random que lo comprueba.

1.2 Carpeta layout>partials:

Aquí nos encontramos con la plantilla del sidebar de la aplicación, esta nos permite tener acceso a todas las vistas gracias a sus botones que están linkeados con las rutas.

2. Livewire:

En esta carpeta tenemos todos los componentes que usamos, estos más bien son las tablas donde mostramos los datos de la base de datos, además de editar y borrar.

3. Vistas

Tenemos las vistas de todas las tablas, en estas se extiende de **app.blade.php, sidebar, componentes livewire** y así damos estructura a nuestra aplicación de manera escalable y reutilizable.

EJECUCIÓN CORRECTA DEL PROGRAMA

Crear la base de datos con todas la tablas:

```
php artisan migrate
```

Llenar la base de datos de datos:

```
php artisan db:seed --class:AuthorSeed
```

```
php artisan db:seed --class:BookSeed
```

```
php artisan db:seed --class:UserSeed
```

```
php artisan db:seed --class:RentSeed
```

Ejecutar la aplicación

```
php artisan serve
```