Documento mini proyecto 1

Cristian David Bernal Colonia 1667405

John Edward Córdoba Rodríguez 1667468

Jeimy Andrea Villegas Torres 1667607

Universidad del Valle.

Tecnología en Sistemas

Aplicaciones en la Web y Redes Inalámbricas

Tuluá – Valle del Cauca.

2019

Lenguaje:

- Php versión 7.2.20

Motor base de datos:

- Mysql versión 5.0.12

Framework usado:

- Laravel 5.8 y 4.0

Componentes:

- Composer version 1.8.6

Paquetes:

- **Acacha adminlte:** plantilla usada para el login y la estructura básica de la aplicación. Además para crear CRUD's de forma elegante.
- **Flash :** paquete de composer que sirve para configurar los mensajes que se mostraran aplicación.
- **Chosen:** Chosen es un complemento de jQuery que hace que las cajas de selección largas y difíciles de manejar sean mucho más fáciles de usar.
- **Trumbowyg:** para dar estilo y agregar funciones a los text area.

Editor:

- Sublime Text 3

Intérprete y gestionador:

- Xampp

Gestión de proyecto y control de versiones:

- Git bash
- Github

Iniciando

Para empezar se crea el proyecto laravel por medio del comando `composer create-project --prefer-dist laravel/laravel blog` Y este nos generará la carpeta con los diferentes componentes separados en sus respectivas carpetas, tanto el front como el back.

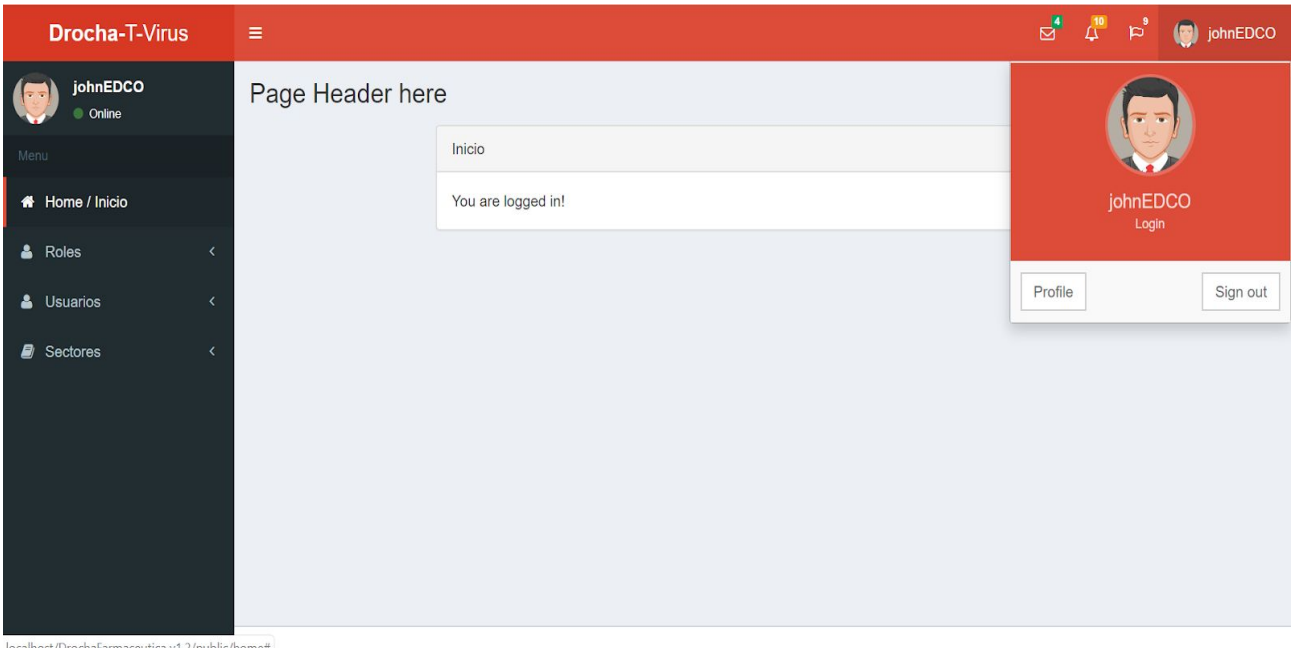
Al abrir el proyecto en la carpeta *Resources* aparece todo lo que tiene que ver con el front end, pues ahí es donde se crean las vistas y el diseño de la aplicación .

También se puede encontrar que en la ubicación *App\Http* se puede trabajar en el back end y es donde se crean los modelos y controladores, los cuales tienen como objetivo recibir los datos capturados por el front y hacer uso de ellos, ya sea para guardar, eliminar, editar, hacer búsquedas, según lo solicitado por el usuario.

En la carpeta *Routes* se encuentra las rutas de la aplicación.

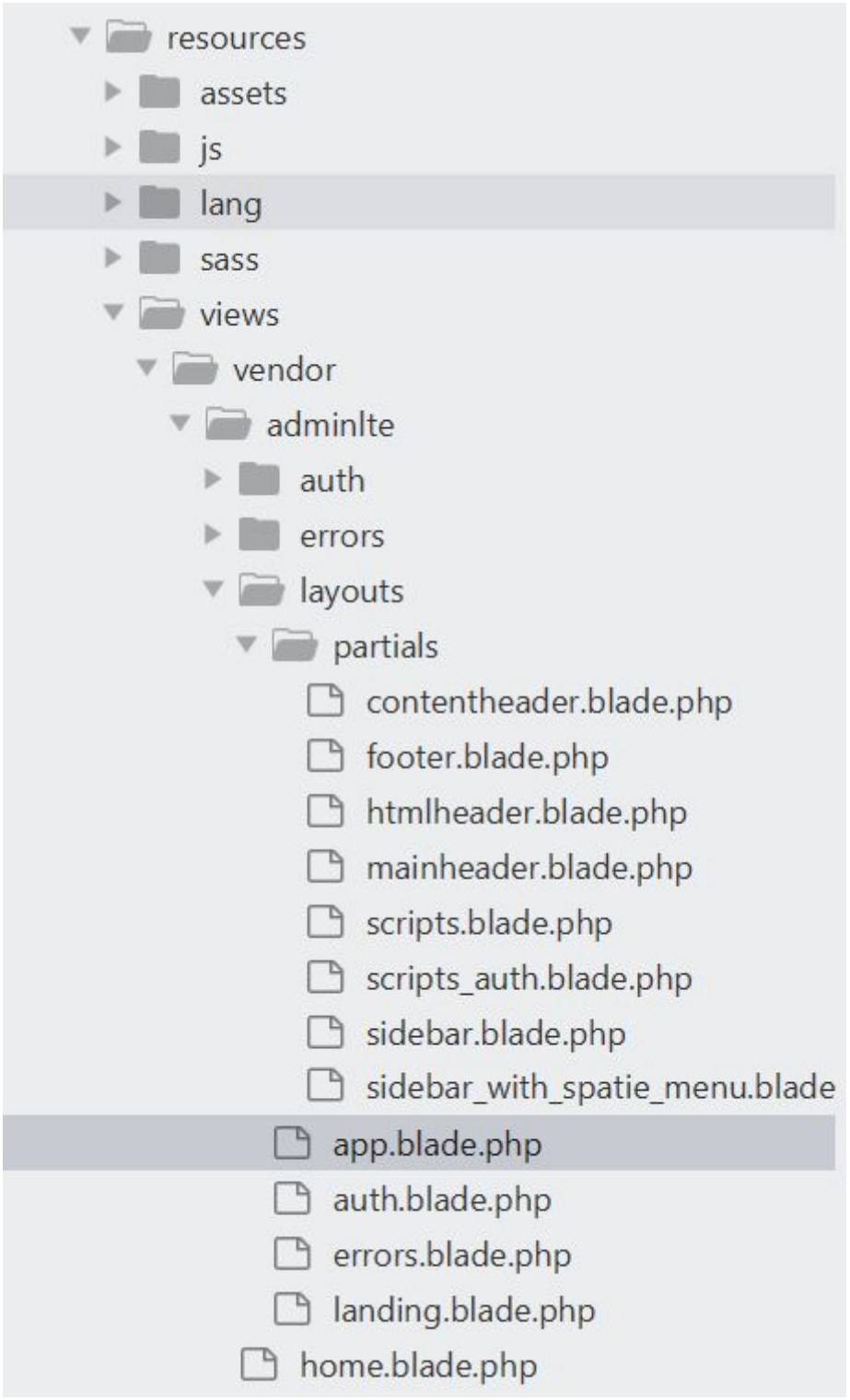
En *Database\migrations* se encuentran las migraciones que se harán a la base de datos.

Este proyecto se crea a partir de la plantilla *acacha/admintle* que nos ofrece una estructura básica y elegante, en la cual está la barra superior y lateral izquierda, el footer, menu de cerrar sesión etc.



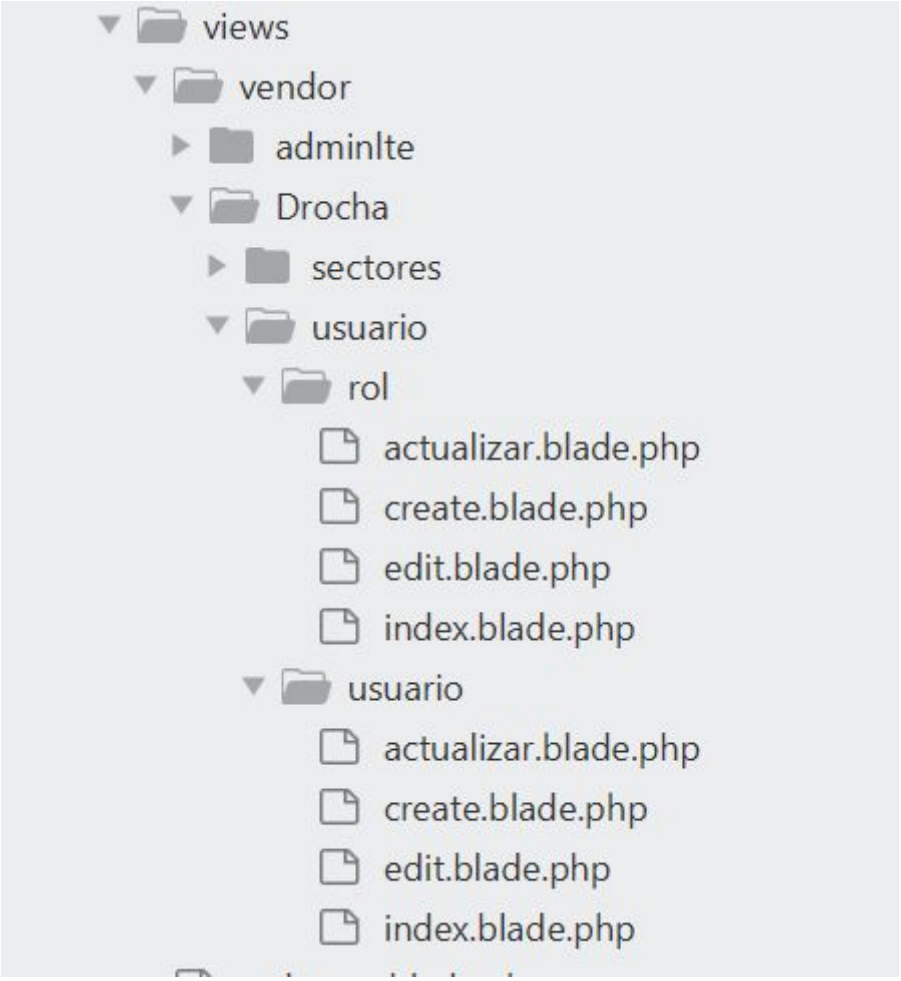
Las vistas se crean en archivos *.blade.php* que es donde se escribe todo el código html, y todo lo que tenga que ver con la parte visual de la aplicación.

Ahora bien *acacha/adminlte* nos deja los archivos *.blade.php* que serían el cuerpo, la cabecera, los menús, etc de la aplicación.

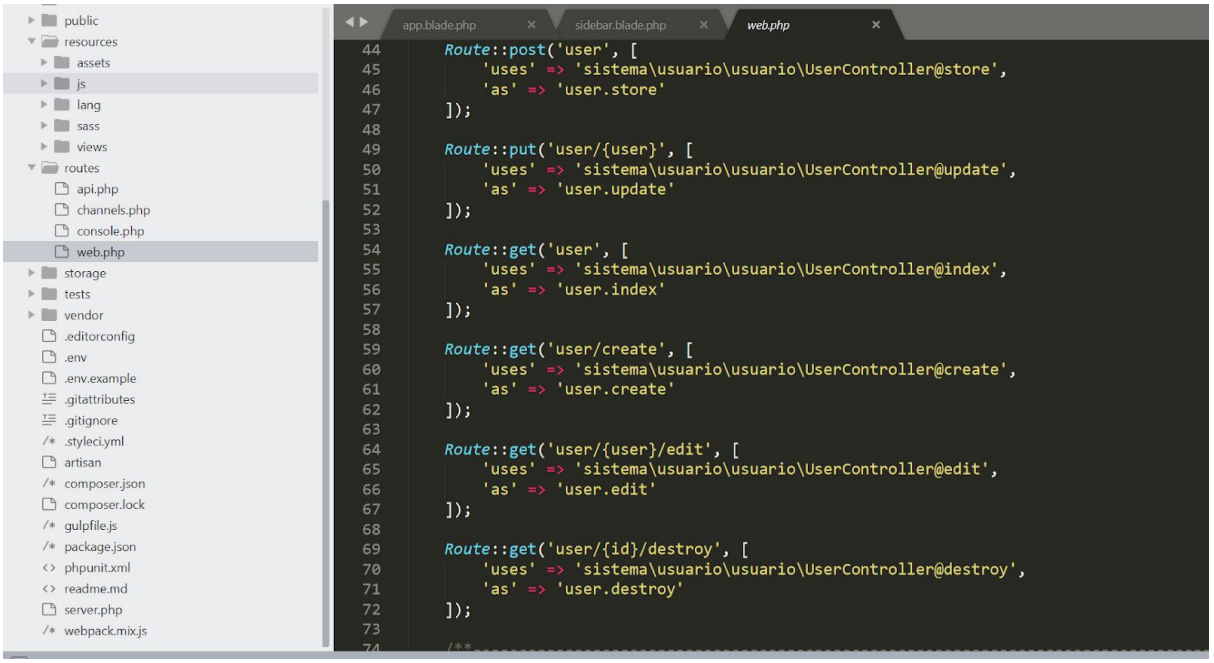


Cada uno de los archivos contiene los diferentes partes que conforman el total de la estructura básica de la aplicación, como por ejemplo en el *app.blade.php* es donde está todo el cuerpo y contenido. También tenemos el *sidebar.blade.php* que es el que contiene la vista de el menú lateral izquierdo donde aparece las opciones que se le da al usuario.

Ahora aquí tenemos las vistas de la parte crear, editar, buscar, eliminar, usuario o rol, cada uno con su codigo html y su css (bootstrap)



Ahora bien una vez que las vistas están creadas, por sí solas no nos dirigen hacia una y otra vista, por lo tanto se necesitan configurar las rutas, que son las que permiten movernos entre vistas. En este proyecto se crean un grupo de rutas de la siguiente manera.



Cada una de las rutas apunta un controlador y un método que retorna la vista indicada y las acciones que se le hayan indicado a el método en el que accedió.

'uses' -> se define la ruta donde se encuentra el controlador y seguido al método que quiere acceder.

'as' -> es el nombre de la ruta que le damos para acceder a la vista.

Por ejemplo en la última ruta *'user/{id}/destroy*' tiene apuntado al controlador *'UserController@destroy'* y seguido al método destroy. Esto me permite que yo pueda ir a esta ruta con solo poner el *'user.destroy'*. Cabe aclarar que el controlador es el que realiza las acciones (back end) y retorna la vista o redirecciona a una en la que ya se encuentre, dependiendo del caso.

Método	Acción
GET	Recupera el recurso desde el servidor
POST	Crea un recurso en el servidor
PUT	Actualiza el recurso en el servidor
DELETE	Elimina el recurso del servidor