

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



DOCENTE:

XENIA IVETTE PEÑATE GODOY

MATERIA:

TÉCNICAS DE PROGRAMACIÓN EN INTERNET

TEMA:

PRACTICA 3 - UNIDAD 3

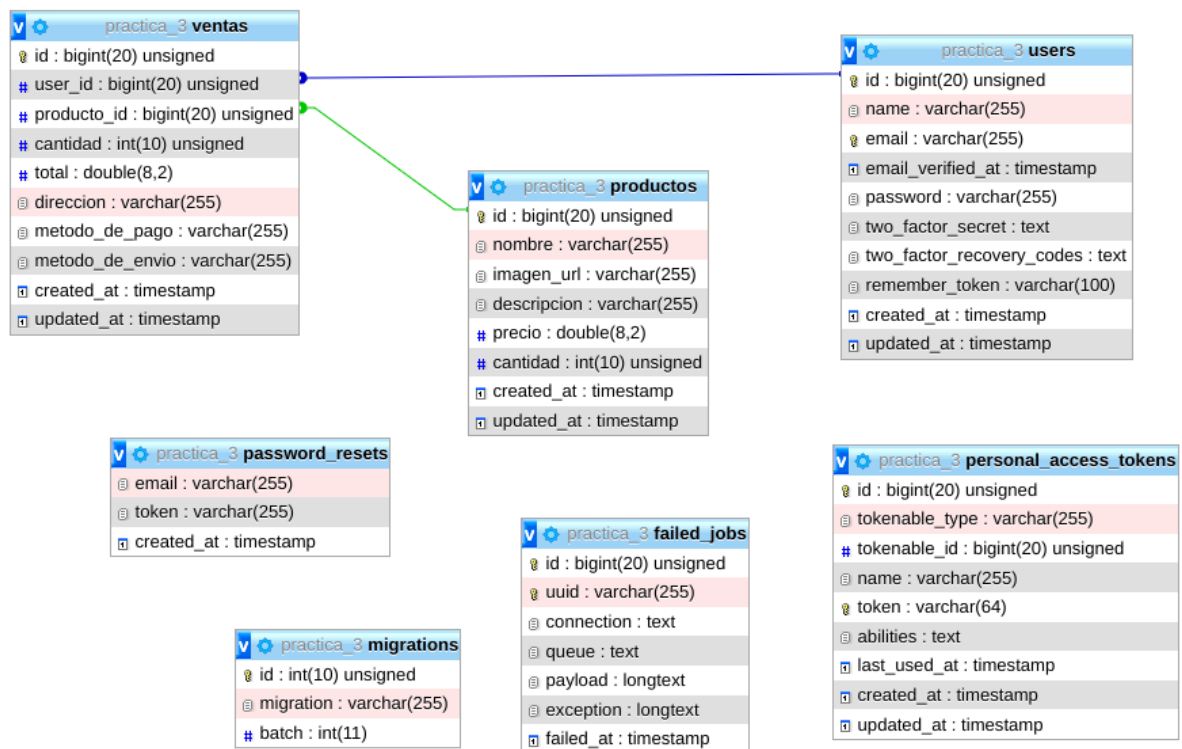
INTEGRANTES:

FERNANDO AMILCAR GALICIA CÁCERES - GC18076

NOÉ FRANCISCO MARTÍNEZ RUÍZ - MR18116

Para el desarrollo de la práctica 3 usamos dos frameworks. Laravel que es un framework de php se utilizó para la parte del backend y se utilizó Vuejs de javascript para el frontend, para visualizar mejor la funcionalidad de la aplicación, pero como en Laravel se expone API REST como veremos más adelante, se puede utilizar programas como Postman para consumir esas apis y así ver la funcionalidad sin necesidad de vistas del frontend.

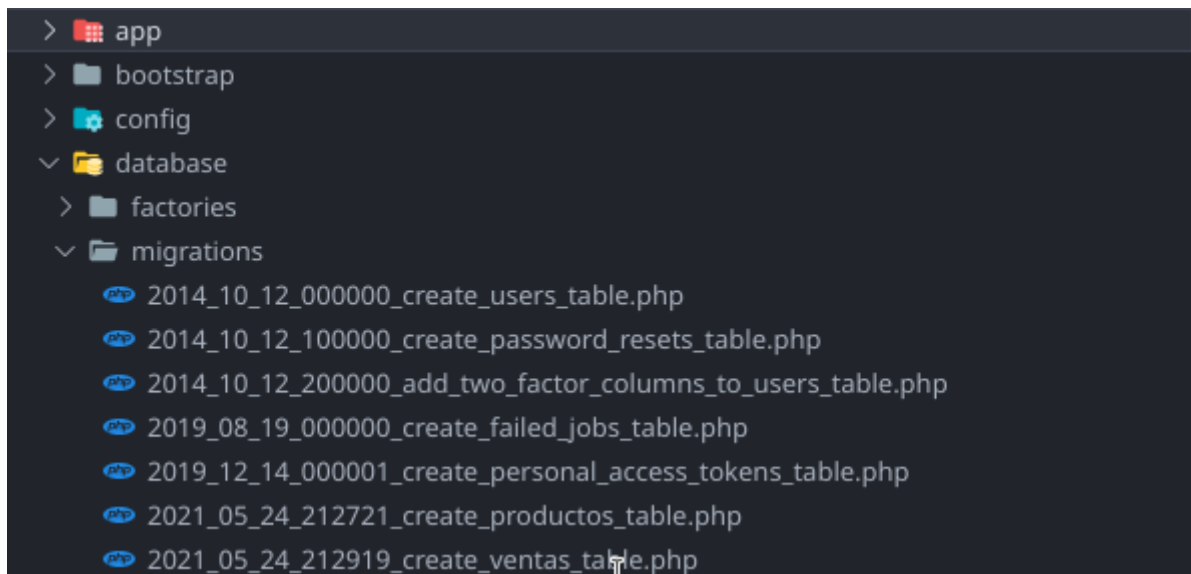
Para la práctica se utiliza un base de datos relacional en mysql que se muestra a continuación:



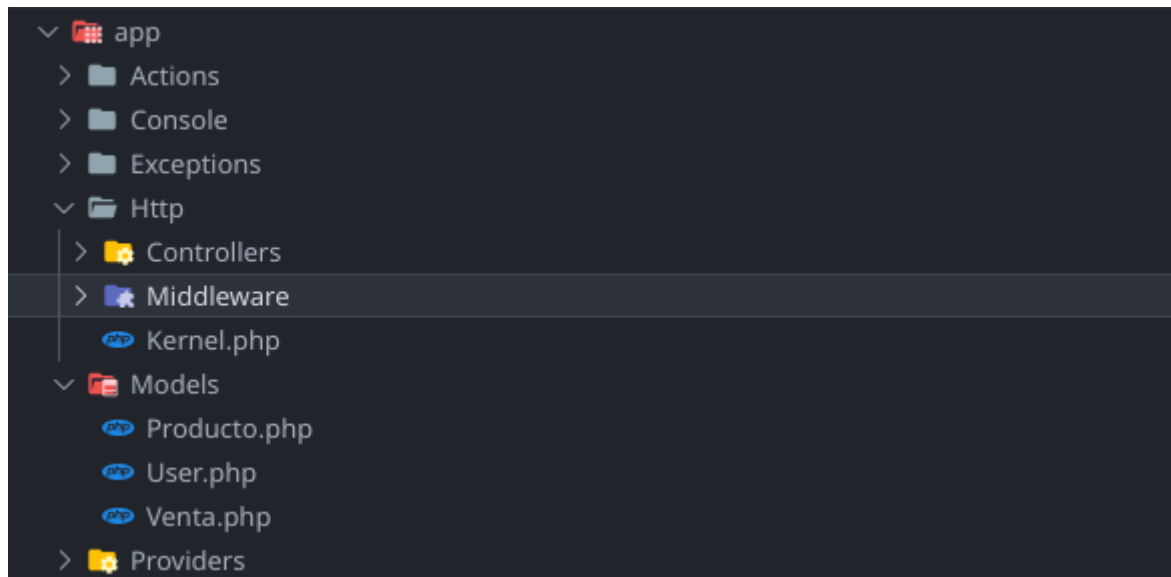
Para agilizar el proceso de la creación de la base de datos se utilizan migraciones incluidas en el framework de laravel que nos permita definir los campos de las tablas en php y con el comando **php artisan migrate** las crea en mysql.

Las migraciones se encuentran en la ruta **database/migrations** dentro del proyecto.

Es importante destacar que Laravel, por defecto agrega algunas tablas como password_resets o la migrations que se encarga directamente de cada migración, es decir cada nueva tabla o nuevo cambio en una tabla y que se sus registros se encarga laravel. Entonces para esta aplicación creamos y relacionamos la migración de productos y la de ventas, además de la migración de usuarios o users que crea laravel pero que cumple con los campos requeridos (usuario y contraseña).



Una vez realizada la migración se procede a crear los modelos que son la representación de las tablas de la base de datos, en este caso en php. Nos sirve para que el controlador pueda obtener los registros de la base de datos. Se hicieron los modelos **Productos.php** que hace representación de la tabla productos, **User.php** para saber los registros y si un usuario está registrado o no y **Venta.php** para la tabla ventas. Estos se encuentran en **app/Models** en el proyecto.

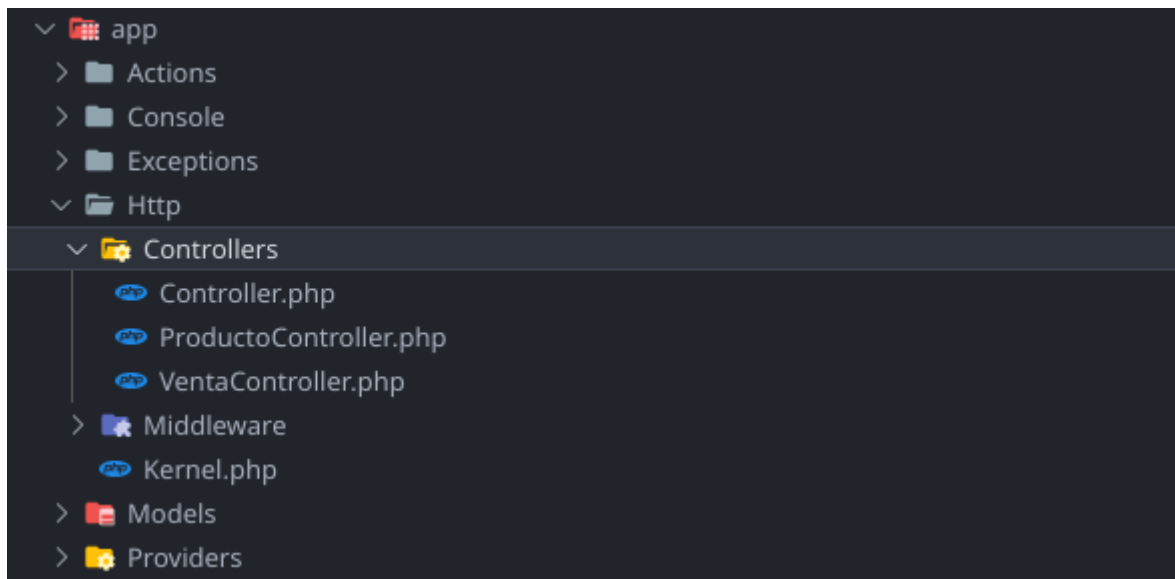


Para obtener la sesión del usuario se utilizó la api del paquete de Sanctum de laravel. Lo que hace es verificar si está autenticado y autorizado el usuario, haciendo uso de un middleware propio de sanctum.

También se utilizó el paquete Fortify de laravel, que proporciona las ruta Auth del usuario, es decir la ruta de login o de inicio sesión, la ruta de register o registro, la ruta de logout o de

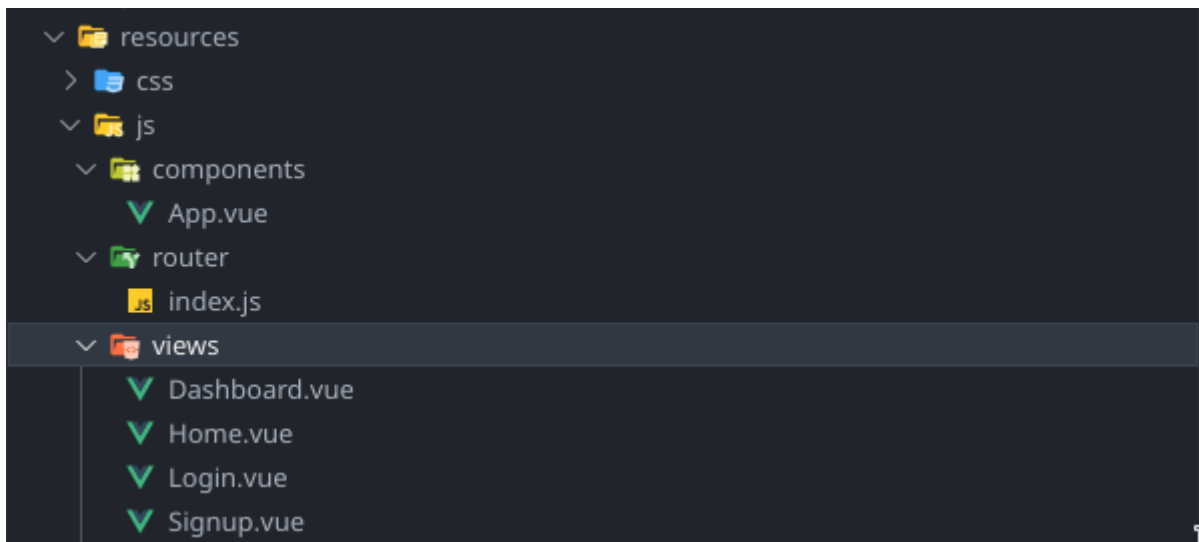
cierre de sesión, entre otras, pero que para esta aplicación solo son necesarias las mencionadas y que se pueden ver mediante el comando **php artisan route:list**.

Para realizar la API que expusiera los recursos de la base de datos a la interfaz de usuario se utilizaron los controladores de **ProductoController.php** que se encarga de proveernos de los productos que se encuentran registrados en la base de datos y de los cuales podrá elegir el usuario para vender y **VentaController.php** que se encarga de procesar los datos enviados desde el frontend que corresponden a una venta por parte del usuario, estos controladores se encuentran en **app/Http/Controllers** dentro del proyecto.

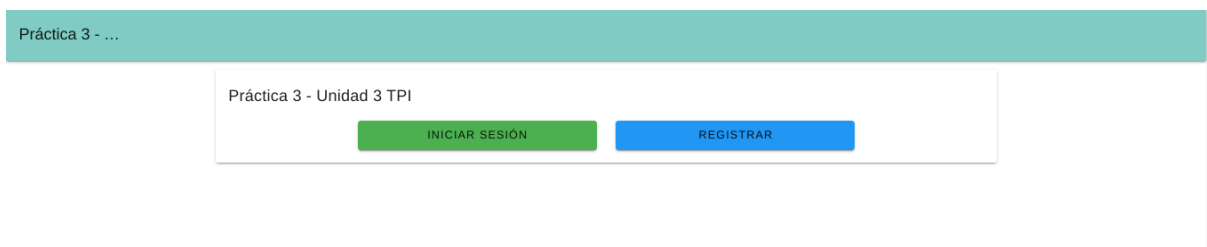


Para el frontend se crearon las vistas Home.vue que contiene la página de inicio para redirigir a iniciar sesión o registrar. Login.vue tiene un formulario para iniciar sesión con un usuario ya registrado, hace uso de la ruta api login que provee Fortify. Signup.vue sirve para registrar un nuevo usuario haciendo uso de la ruta api register de Fortify. Ambas vistas después de realizar su respectiva función recuperan el usuario por medio de la ruta api con el middleware auth de Sanctum.

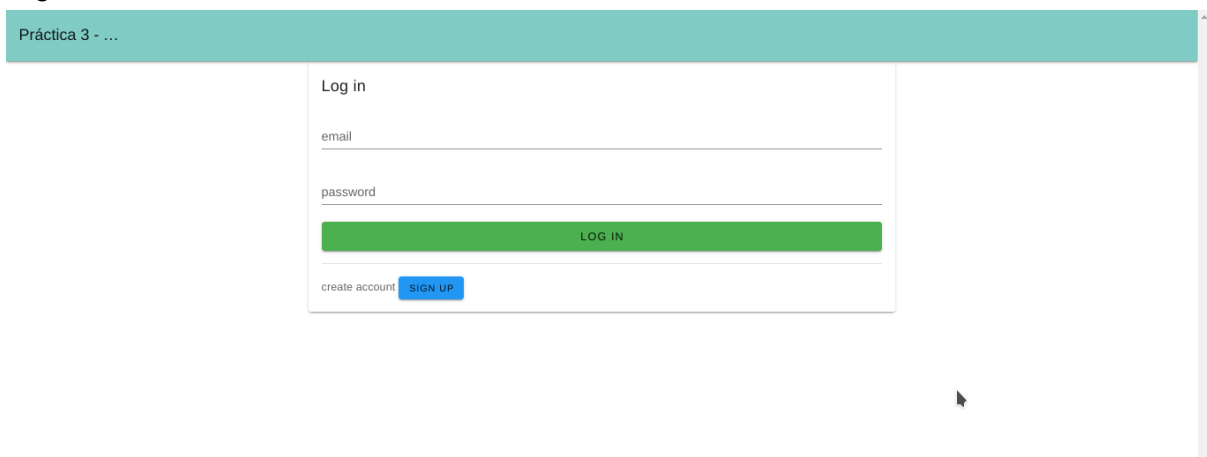
La vista Dashboard.vue, tiene un formulario con los campos necesarios para realizar la venta consumiendo las apis necesarias de producto y venta, y cuando se realiza una venta, se persiste en la base de datos y así queda guardada la información de dicha venta. Estos archivos se encuentran en resources/js/views dentro del proyecto.



Home.vue



Login.vue



Singup.vue

Práctica 3 - ...

Sign up

Name

email

password

Confirm password

SIGN UP

do you have account?

LOG IN

Ejemplo de una venta Dashboard.vue

Práctica 3 - ...

LOGOUT

Registrar Venta

Productos

Coca cola

Cantidad

3

Método de Pago

Efectivo

Método de Envío

Pedidos Ya!

Dirección

Universidad de El Salvador

Total

\$ 1.5

CANCELAR

GUARDAR