

Backend (Laravel)

Requisitos:

- PHP 8.2
- Composer

Configurar:

Para poder usar el Swagger es necesario modificar el siguiente archivo:

"Backend\app\Http\Controllers\ClientController.php"

Modificarla url con la IP del equipo.



```
/**
 * @OA\Info(
 *     title="Laravel + Vue ",
 *     version="0.2",
 * ),
 * @OA\Server(
 *     url="http://localhost:8000"
 * ),
 */
```

Y ejecutar el siguiente comando para aplicar los cambios en swagger:

"php artisan l5-swagger:generate"

Swagger:

Para poder usar el swagger solo necesitamos ejecutar:

- Php artisan serve → arranca el servicio de laravel.

Y usamos la url → [http:// \[localhost o IP\]:8000/doc](http://localhost:8000/doc)

Frontend (VUE)

Requisitos:

- Npm
- Vue 3.2

Configurar:

1. Npm i ➔ Instala las dependencias necesarias para el frontend.
2. Modificar el archivo "src/storage/index.js" para cambiar la variable url_api con la url del backend:

```
import { createStore } from 'vuex'

export default createStore({
  state: {
    url_api: "http://localhost:8000"
  },
})
```

3. Npm run build ➔ Compila la app dentro de la carpeta dist.

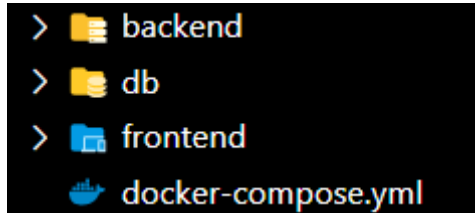
Para poder hacer una prueba fuera del Docker no es necesario hacer el build, con usar "npm run serve" ya se podrá usar.

Docker

El Docker está compuesto por el backend (Laravel), frontend (apache2), base de datos (Mysql) y phpmyadmin (opcional) para poder acceder a la base de datos.

Este Docker-compose está preparado para usar Jenkins y que el propio Jenkins sea quien se encargue de manejar y actualizar el backend y frontend.

La estructura es la siguiente:



Al ejecutar “docker-compose up -d” todos los servicios se descargarán y se pondrán en marcha.

* Debido a mi falta de conocimiento en java y por lo tanto en Spring Boot, he decidido usar Laravel 10.