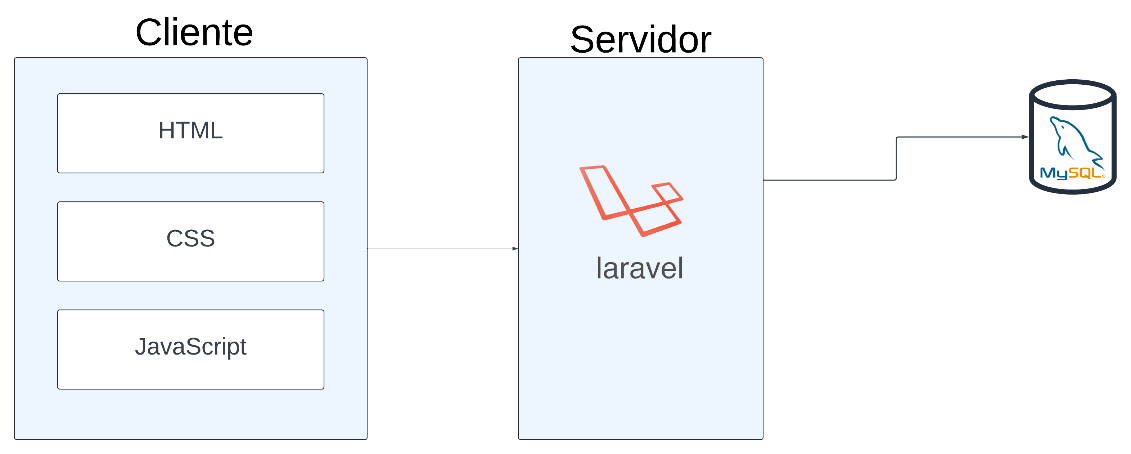
**Especificaciones técnicas**

1. **Introducción.**

La aplicación es una plataforma web para digitalizar su inventario de libros. Donde podrá gestionar un catálogo mediante una interfaz web para interactuar con la API.

1. **Arquitectura del sistema.**

Diagrama de Arquitectura: La conexión entre el frontend, backend y base de datos.



* 1. Cliente, hará las vistas para mostrar la información y la interacción con el usuario, en este caso, mostrará una página con el catálogo de los libros y tendrá otra página para la gestión de libros.
  2. Servidor, gestiona toda la lógica de creación, edición, actualización y borrado de libros.
  3. MySQL, almacena la información de los libros.

1. **Entorno de desarrollo**

**Requisitos previos**:

* 1. **Framework**: Laravel 10.48.20

<https://laravel.com/docs/10.x/installation>

Framework de php

* 1. **PHP**: Versión 8.1.10

<https://www.php.net/downloads.php>

Entorno para el lenguaje de programación

* 1. **Laragon**: 6.0

<https://laragon.org/download/>

Servidor para alojar el proyecto

* 1. Visual Studio Code

<https://code.visualstudio.com/download>

Editor de texto

* 1. Extensión de Visual Studio Code (Live Serve)

Está extensión nos permite correr el frontend.

* 1. Composer: version 2.4.1

Manejador de paquetes para php

* 1. Node: v22.11.0

https://nodejs.org/es/download

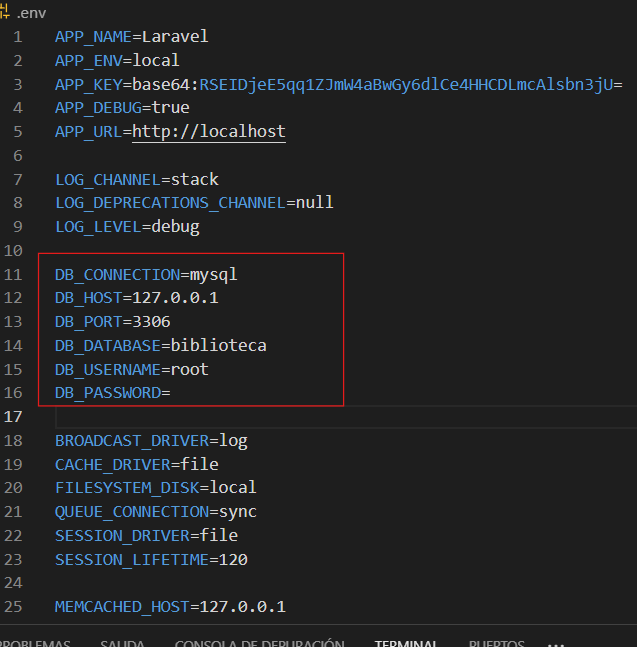
1. **Configuración del proyecto**

**Backend**

* 1. Después de tener instalado laragon, tendremos que clonar el proyecto “biblioteca-backend” en la carpeta ..\Laragon\laragon\www para que el proyecto pueda correr.

La ruta a esa carpeta se puede encontrar dando click en “**Root”**:



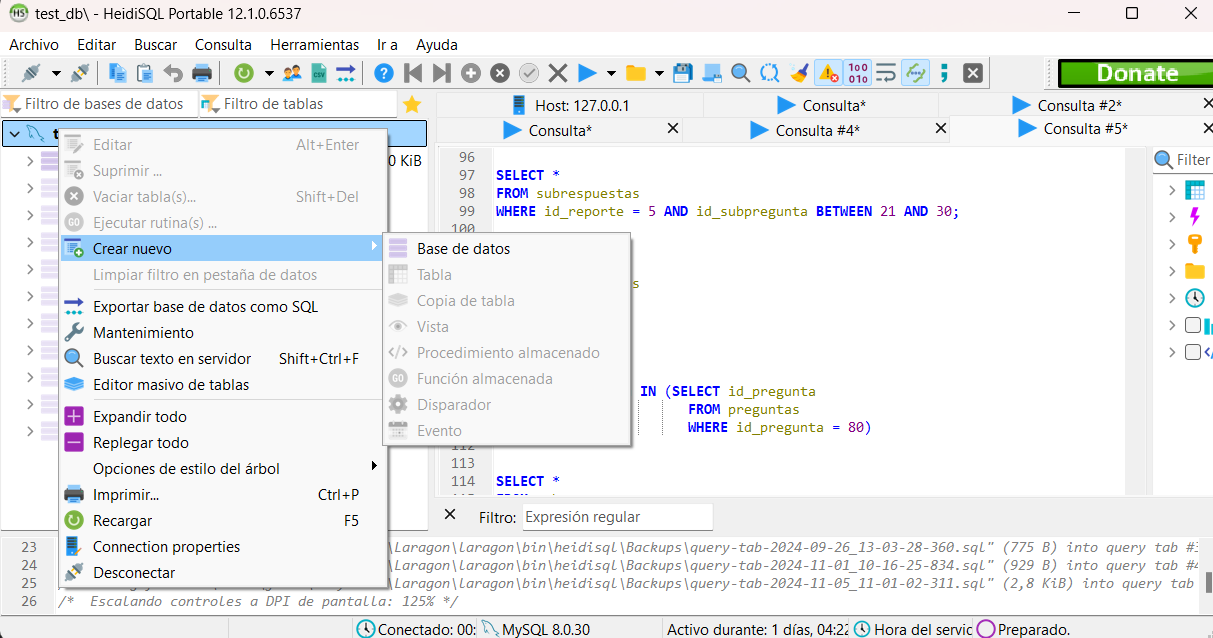
* 1. Después damos click en “**Terminal**” y ejecutamos los siguientes comandos:
     + Composer install
     + Npm install
  2. Después damos click en “Iniciar Todo”, en este momento el proyecto ya deberá estar corriendo.
  3. Abrimos el proyecto y configuramos el archivo .env  
     

Modificamos los datos a la base de datos.

**Base de datos**

* 1. Damos en el botón “**Base de datos**”
  2. Podemos creamos una conexión e ingresamos:

Creamos una nueva base de datos llamada biblioteca sin necesidad de ejecutar sentencias SQL.

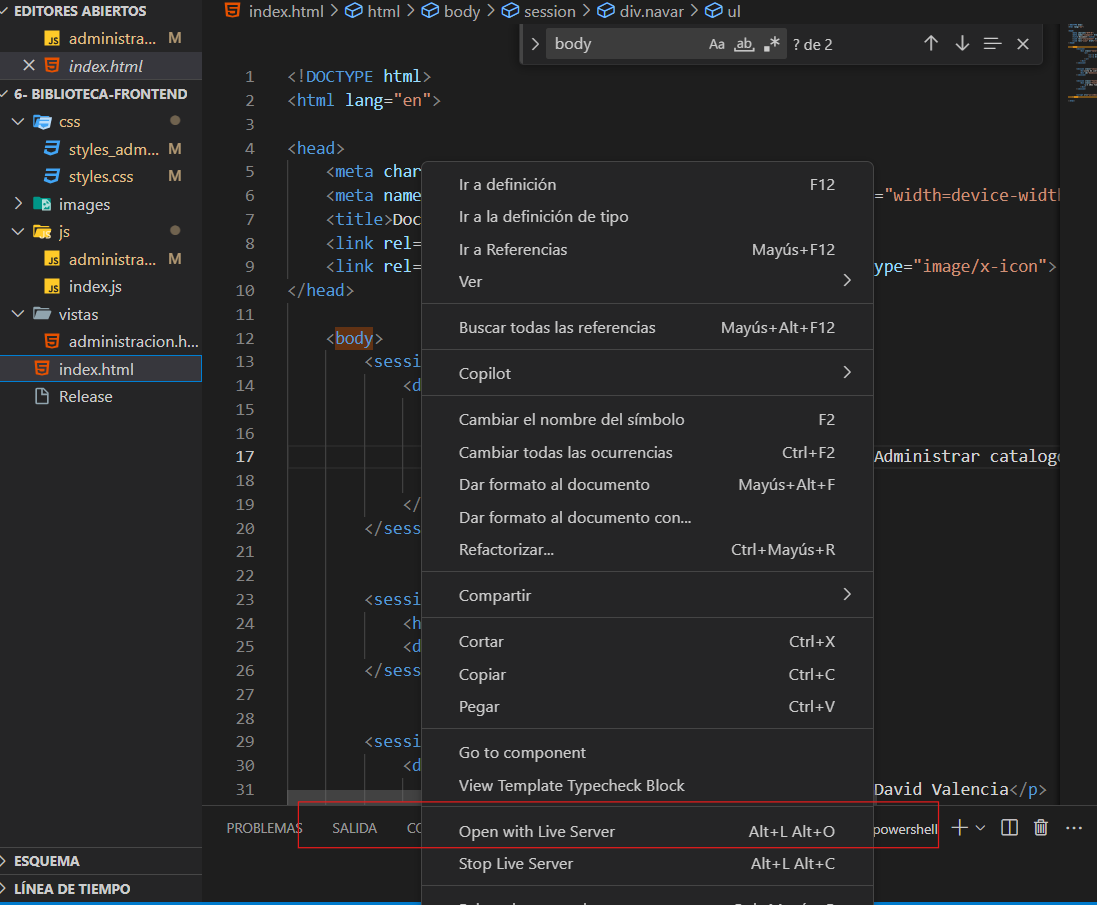


**Ejecutar las migraciones**

* 1. Nuevamente damos click en “**terminal**” y ejecutamos el siguiente comando:

**php artisan migrate:fresh**  
Esto creará las tablas en base de datos y en caso de estar sucia la base de datos por algún motivo, se hará la respectiva actualización.

**Frontend**

* 1. Podemos clonar el proyecto de github Biblioteca-frontend en cualquier parte del equipo.
  2. Abrimos el proyecto en Visual Studio Code, nos ubicamos el archivo HTML y damos Click derecho y ejecutamos:  
     

1. **Estructura de la API**
   1. **Rutas**
      * Route::apiResource('book', BookController::class);

Controla las peticiones para las operaciones de CRUD de los libros.

* 1. **Controladores**
     + **BookController**

Contiene todos los métodos que controlan las operaciones del CRUD

1. Index.
2. Store.
3. Update.
4. Destroy.

* **BookRequest**

Contiene las reglas para verificar el request que está enviando el cliente

'title' => 'required|max:200',

            'author' => 'required|max:120',

            'date\_publication' => ['required', 'date'],

            'gender' => ['required', 'max:100'],

            'category' => ['required', 'max:100']

Contiene la respuesta para el cliente en caso de error.

* **BookResource**

return [

'id\_book'=>$this->id\_book,

'nombre'=>$this->title,

'autor'=>$this->author,

'fecha de publicacion'=>$this->date\_publication,

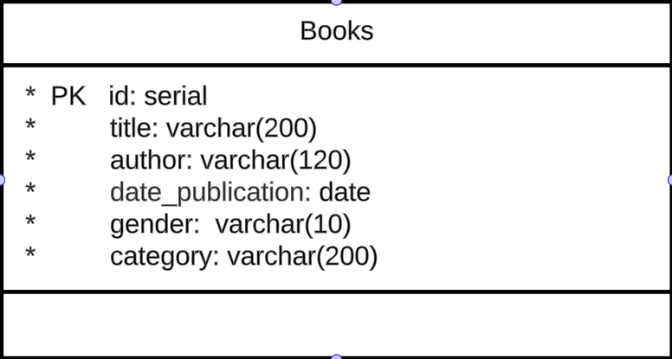
'categoria'=>$this->category,

'genero'=>$this->gender,

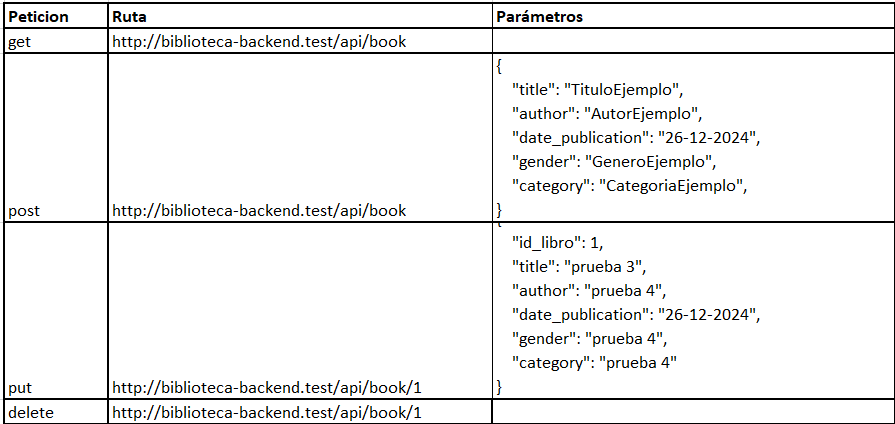
'libreria'=>$this->id\_library,

];

1. **Arquitectura de base de datos.**

****

1. **Documentación de la APIRest.**

****