Documentación

**1. Introducción**

El presente trabajo práctico tuvo como objetivo desarrollar un sistema para administrar cursos/aulas y estudiantes en una institución educativa.

El sistema permite:

* Registrar cursos con sus datos básicos.
* Registrar estudiantes con su información de contacto.
* Gestionar inscripciones de estudiantes a distintos cursos.

Para llevarlo a cabo se utilizó una arquitectura full stack con:

* Frontend: React.
* Backend: Node.js con Express.
* Base de datos: MySQL.

**2. Tecnologías utilizadas:**

React: desarrollo de la interfaz gráfica.

Node.js + Express: creación de la API REST, con sus controllers, routers e index.js para manejar la conexión al servidor.

MySQL: almacenamiento de datos mediante tablas relacionales.

GitHub: repositorio compartido para control de versiones y colaboración.

Desarrollo del mismo:

1. Incializamos en visual con un git init para permitirnos subir el proyecto a github.
2. Creamos la Carpeta Frontend e Inicializamos con sus respectivos comandos:

Npm create vitelatest

Npm i

1. Creamos la carpeta Backend e inicializamos con sus respectivos comandos:

* Npm init –y
* Npm i express
* Npm i nodemon: logra que no caiga el servidor o conexión en caso de algún error y se levanta automáticamente.
* Npm i mysql2
* npm install cors

1. Creamos en Backend la carpeta controllers, config DB (dentro del archivo dataBase), middleware,routes.
2. Creamos en Backend el archivo index.js.
3. Ingresamos en la carpeta controllers, creamos tres archivos correspondientes a las tablas. Estudiante.js,cursos.js e inscripción.js.
4. Creación de la Base de Datos en MySQL. Se creó una base de datos con tres tablas principales:

CREATE DATABASE trabajo\_practico1;

USE practico;

CREATE TABLE cursos (

id\_curso INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

descripcion VARCHAR(500)

);

CREATE TABLE estudiantes (

id\_estudiante INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE inscripciones (

id\_inscripcion INT AUTO\_INCREMENT PRIMARY KEY,

fecha\_inscripcion DATE NOT NULL,

id\_estudiante INT,

id\_curso INT,

FOREIGN KEY (id\_estudiante) REFERENCES estudiantes(id\_estudiante),

FOREIGN KEY (id\_curso) REFERENCES cursos(id\_curso)

);

SELECT \* FROM cursos;

INSERT INTO cursos (id\_curso, nombre, descripcion) VALUES (1, "Priscila", "4to");

SELECT \* FROM estudiantes;

INSERT INTO estudiantes (id\_estudiante,nombre,email) VALUES (01, "Priscila", "priiolariaga23@gmail.com");

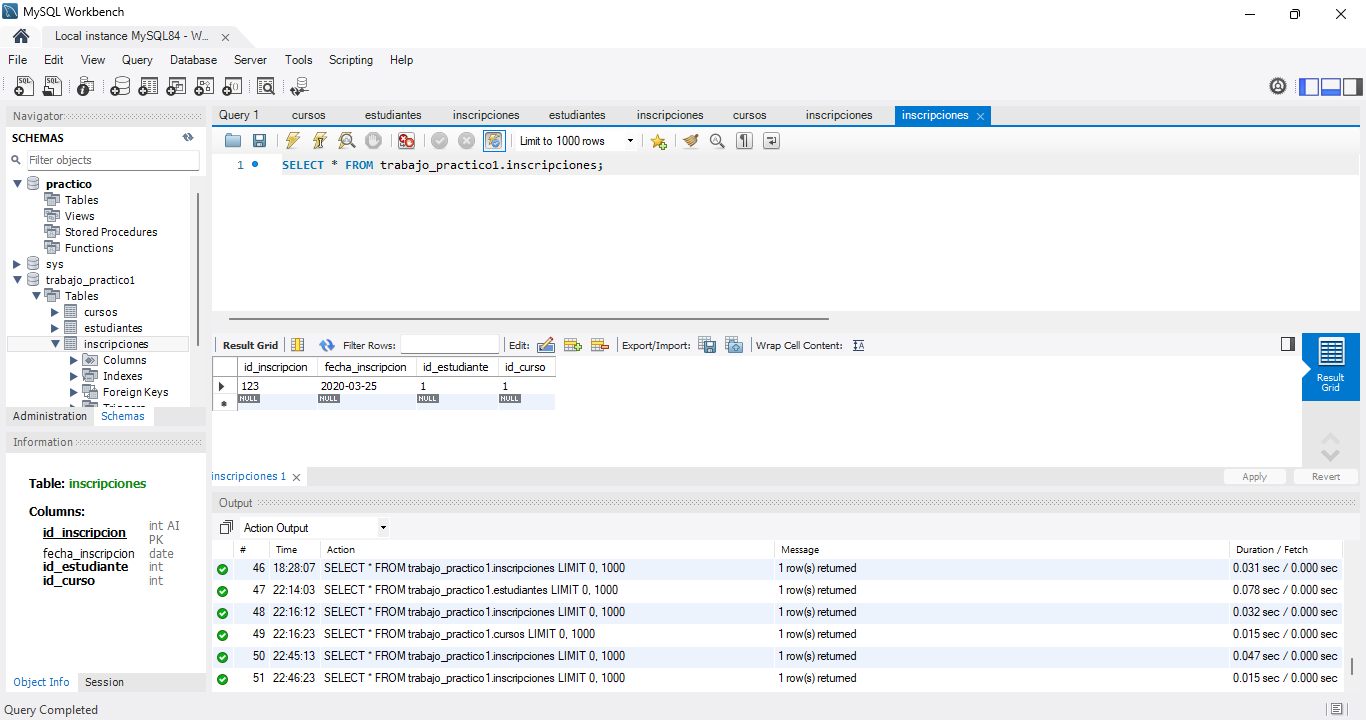
ALTER TABLE inscripciones

MODIFY fecha\_inscripcion DATE NOT NULL;

SELECT \* FROM inscripciones;

INSERT INTO inscripciones (id\_inscripcion,fecha\_inscripcion,id\_estudiante,id\_curso) VALUES (123,"20-03-25",1,01);

SHOW TABLES;

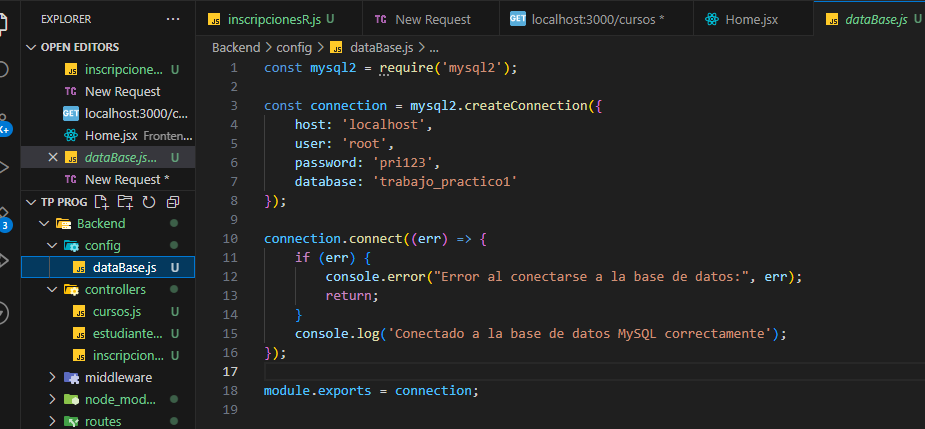


1. Regresamos a la terminal de Visual Studio Code e Instalamos lo siguiente:

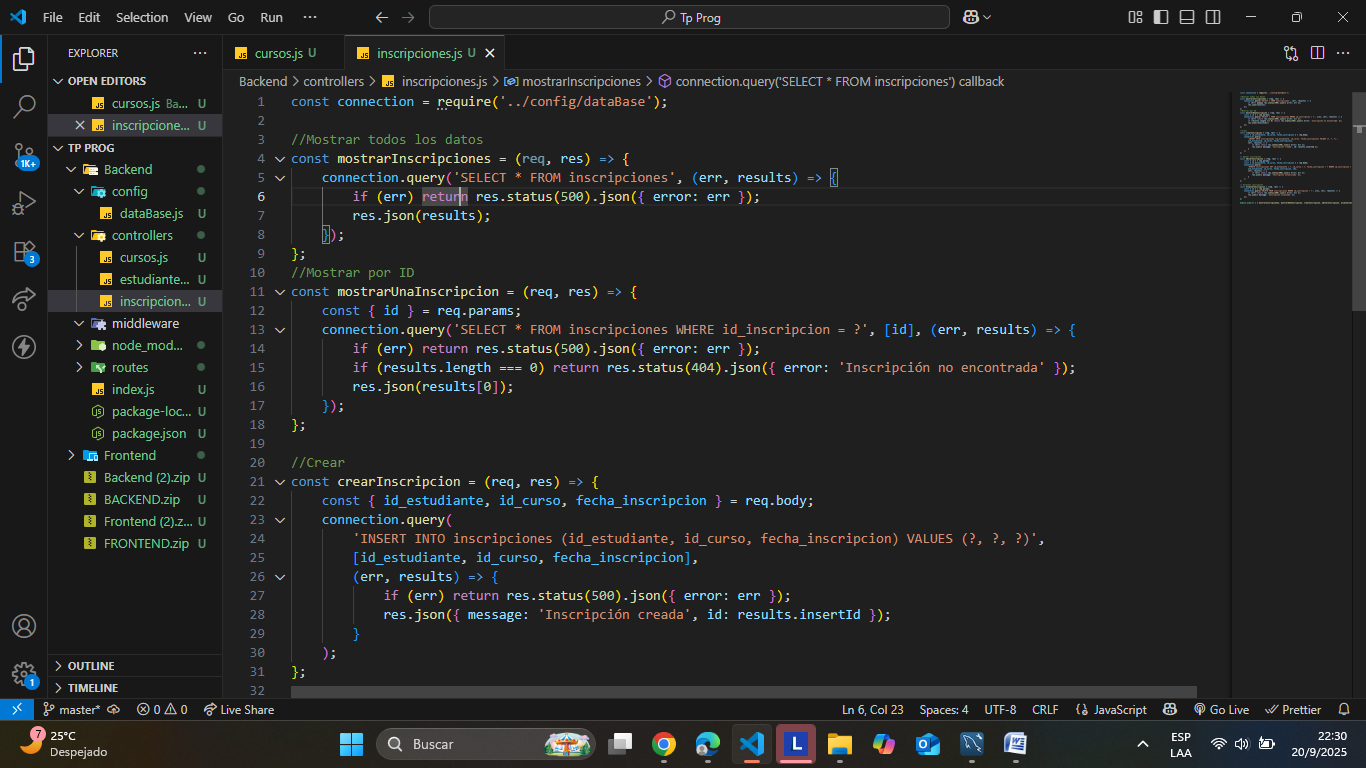
npm i react-router-dom: Sirve para **manejar las rutas** dentro de una aplicación React.

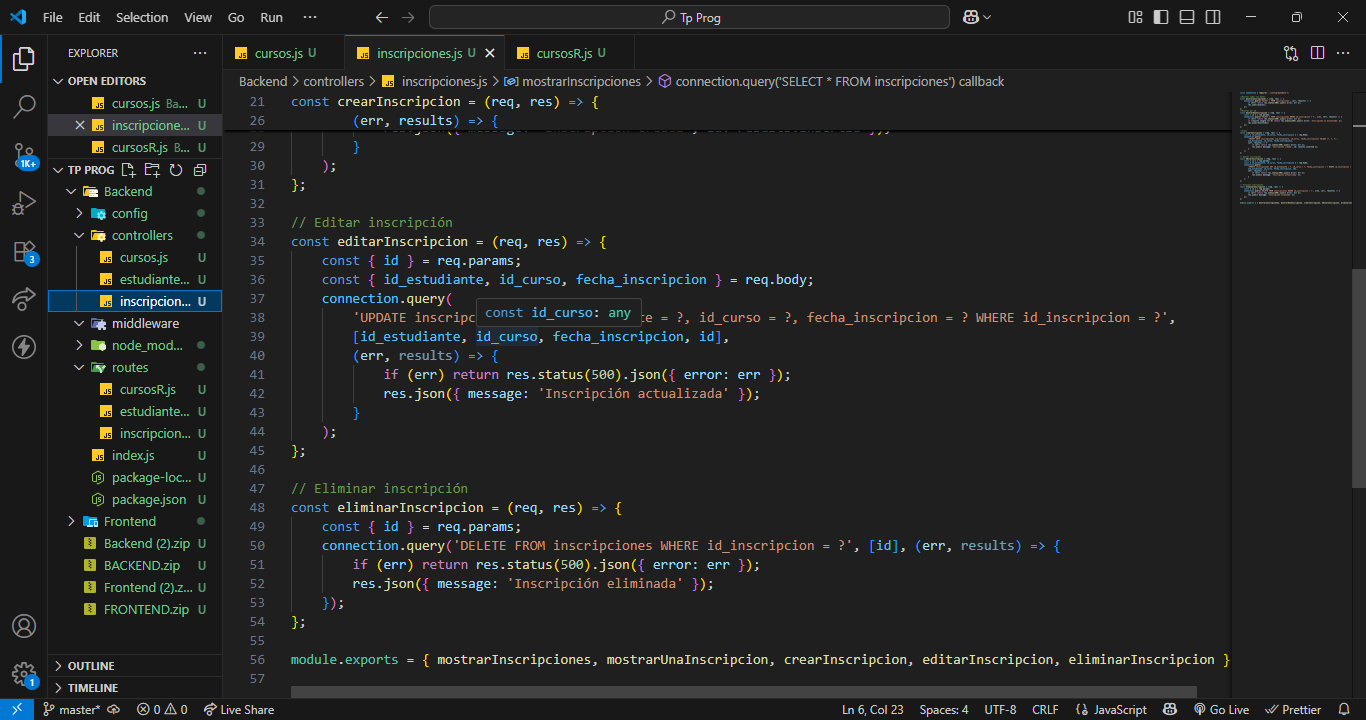
npm i axios: Sirve para **hacer peticiones HTTP al** backend o a APIs externas.

1. Archivo de la carpeta Config “dataBase”

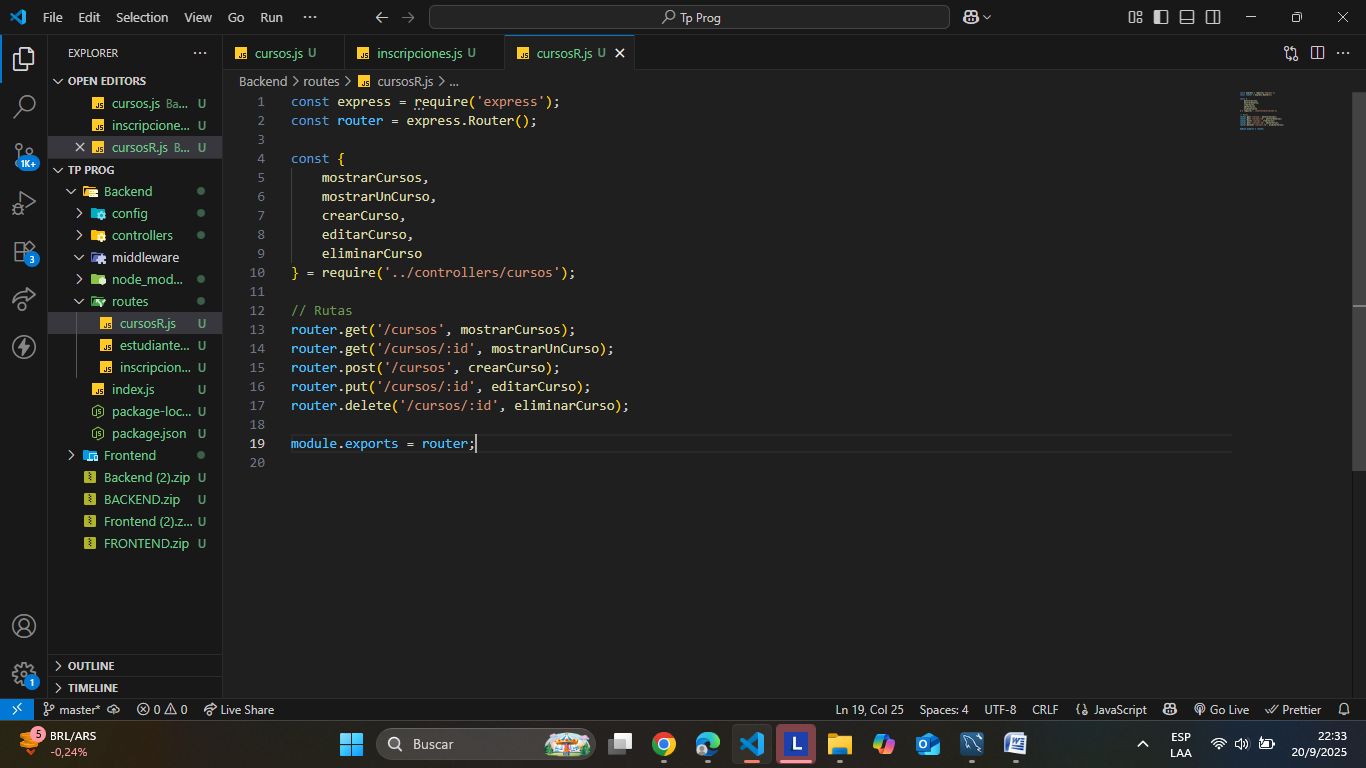


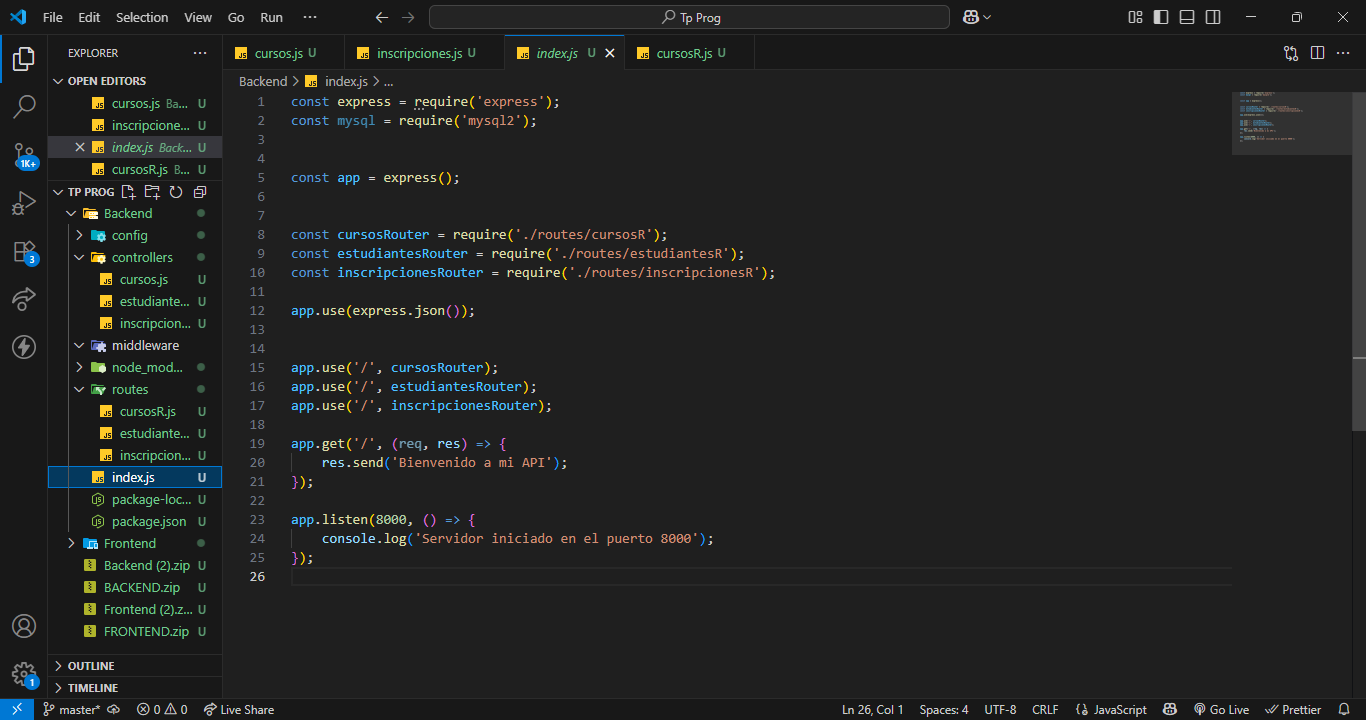
1. Continuamos con los Controllers, aplicando el mismo procedimiento en los tres archivos: Estudiantes, cursos e inscripciones.



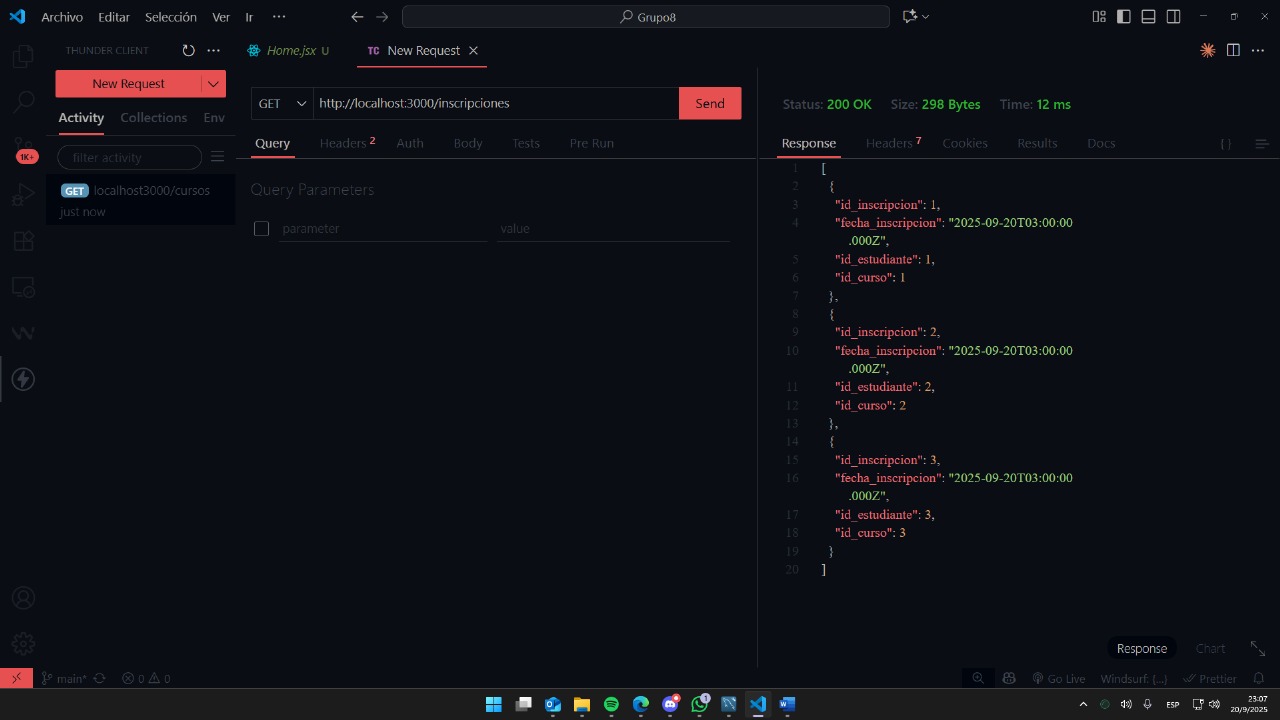


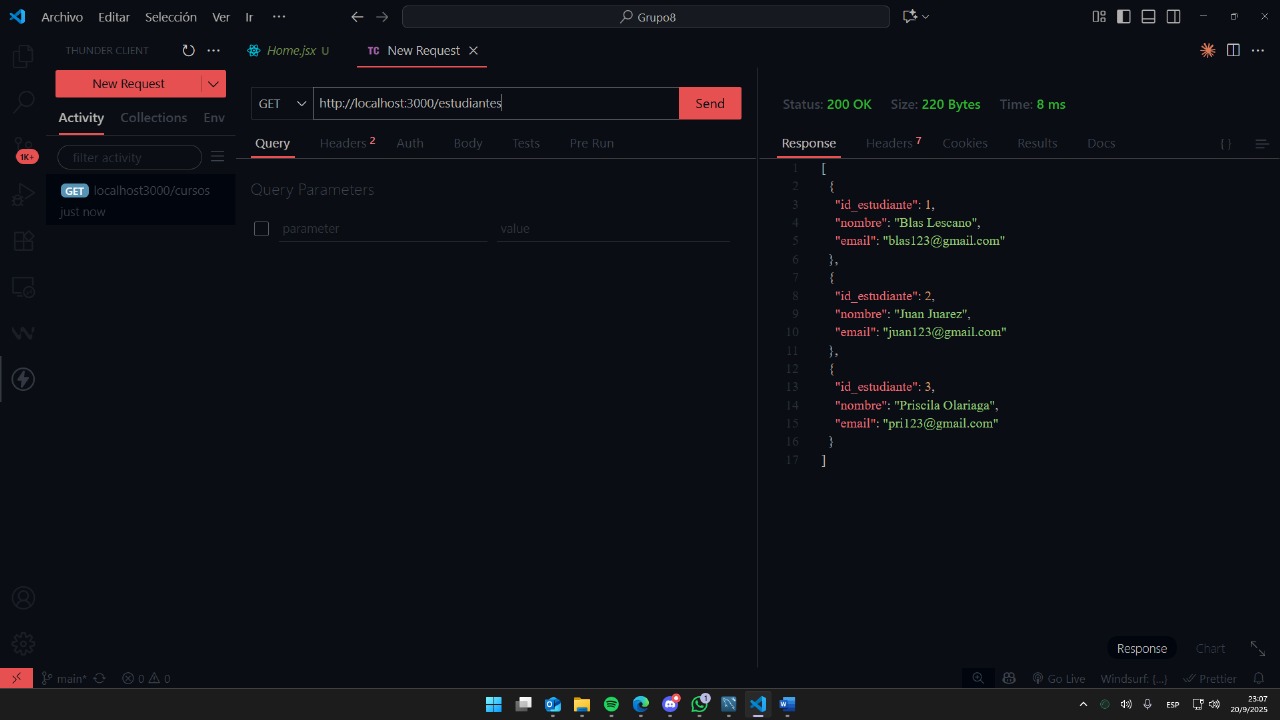
1. Luego, continuamos con los archivos de Routes:

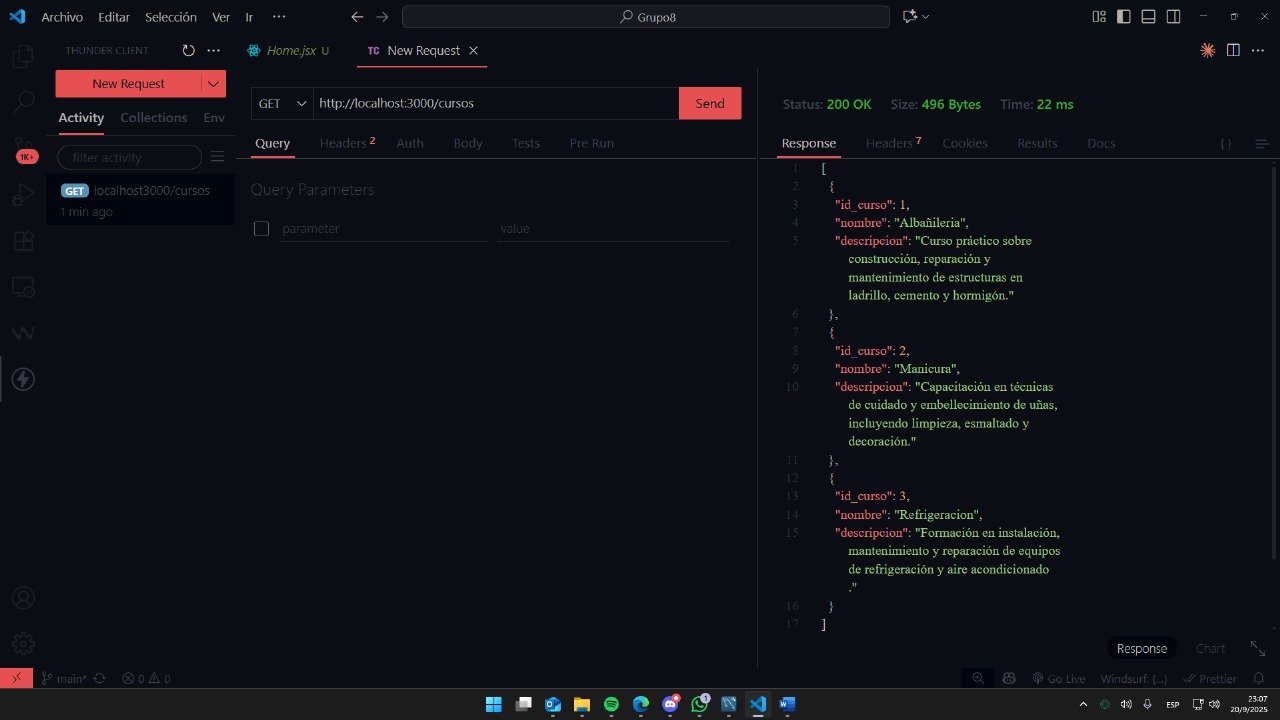




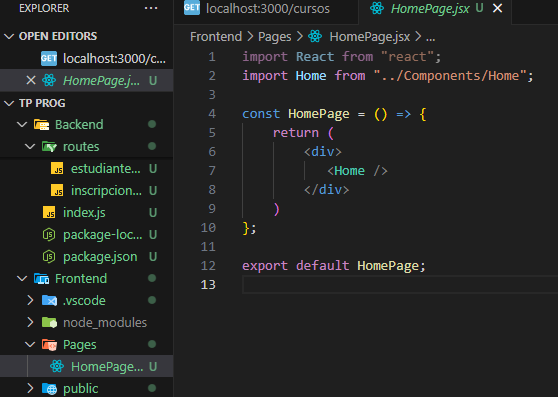
Nos ubicamos en la terminal de Visual Studio Code dentro de la carpeta Backend para verificar la conexión con la Base de datos.Luego instalamos la extensión ThunderClient para confirmar dicha conexión y utilizar las peticiones http (get,put,delete,post).



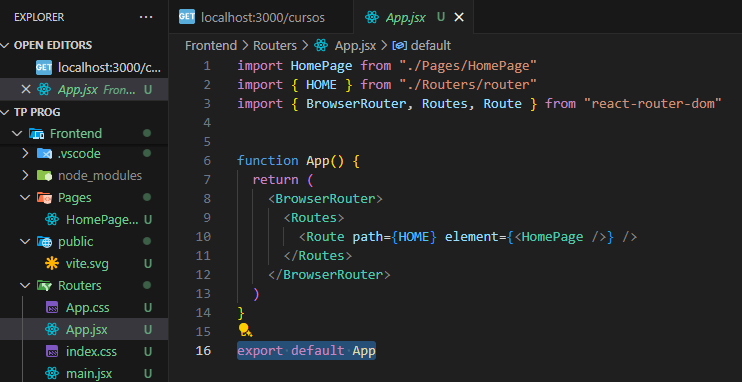


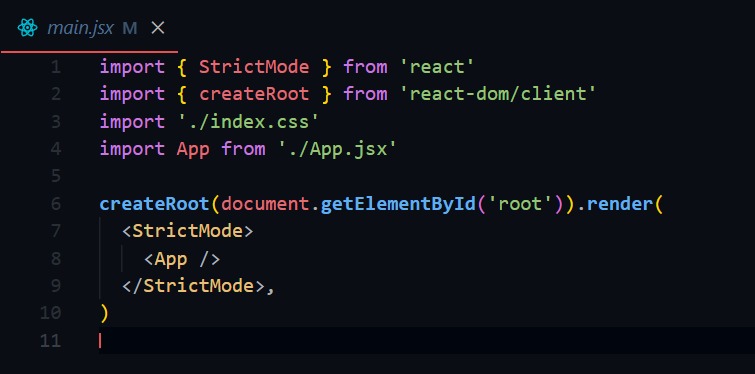


En FRONTEND: Luego de inicializarnos con dichos comandos mencionados anteriormente, realizamos la creación de la carpeta PAGES, que contiene el archivo “HomePage.jsx”



ROUTES:



MAIN.JSX: 

Conclusión

El desarrollo de este proyecto grupal nos permitió integrar los conocimientos adquiridos sobre diseño y gestión de bases de datos, así como la implementación de un frontend en React y un backend en Node.js conectados a MySQL. Durante el proceso, aprendimos a crear tablas, establecer relaciones y realizar consultas, además de manejar GitHub para trabajar de manera colaborativa y organizada. En conclusión, este proyecto no solo nos brindó experiencia práctica en el uso de herramientas y lenguajes actuales, sino que también reforzó nuestras habilidades de **colaboración, planificación y resolución de conflictos**, aspectos fundamentales para futuros desafíos académicos y profesionales.