Proyecto Colegio – Programación IV

Carlos Manuel Muñoz Muñoz

Universidad de Manizales

15/04/2025

# Introducción

El documento describe el desarrollo de un sistema básico de gestión académica, el cual permite registrar y administrar información relacionada con estudiantes, profesores, materias, y la asignación de estos, el proyecto fue realizado para la asignatura Programación IV, utilizando Node.js con Express y Sequelize para la conexión con una base de datos relacional.

# Descripción de la base de datos

La base de datos está compuesta por las siguientes tablas:  
  
- **Estudiante**: Contiene información como documento, nombre, apellido, fecha de nacimiento, email y celular.  
- **Profesor**: Incluye los campos documento, nombre, apellido, profesión, email y celular.  
- **Materia**: Registra datos de materias como nombre, horas por semana y descripción.  
- **MateriaEstudiante**: Relaciona estudiantes con materias, almacenando además fecha de inscripción, nota final y estado.  
- **MateriaProfesor**: Relaciona profesores con materias, registrando la fecha de asignación.

# Explicación de los endpoints

### /api/estudiante

Controlador: estudianteControlador.js  
Archivo de rutas: rutasEstudiante.js

1. **/registrar**  
   Registra un nuevo estudiante en la base de datos.
2. **/listar**  
   Lista todos los estudiantes registrados.
3. **/actualizar/:documento**  
   Actualiza los datos de un estudiante usando su documento como identificador.
4. **/borrar/:documento**  
   Elimina un estudiante de la base de datos por su documento.

### /api/profesor

Controlador: profesorControlador.js  
Archivo de rutas: rutasProfesor.js

1. **/registrar**  
   Crea un nuevo profesor.
2. **/listar**  
   Devuelve una lista de todos los profesores.
3. **/actualizar/:documento**  
   Permite actualizar la información de un profesor.
4. **/borrar/:documento**  
   Elimina un profesor por su documento.

### /api/materia

Controlador: materiaControlador.js  
Archivo de rutas: rutasMateria.js

1. **/registrar**  
   Registra una nueva materia.
2. **/listar**  
   Lista todas las materias disponibles.
3. **/actualizar/:documento**  
   Actualiza los datos de una materia por su ID.
4. **/borrar/:documento**  
   Elimina una materia por ID.

### /api/inscripcion\_materias

Controlador: materiaestudianteControlador.js  
Archivo de rutas: rutasMateriaEstudiante.js

1. **/asignar**  
   Asigna una materia a un estudiante.
2. **/listarEstudianteMateria/:documento\_estudiante**  
   Lista todas las materias inscritas por un estudiante específico.
3. **/listarMateriaEstudiante/:id\_materia**  
   Muestra todos los estudiantes inscritos en una materia específica.

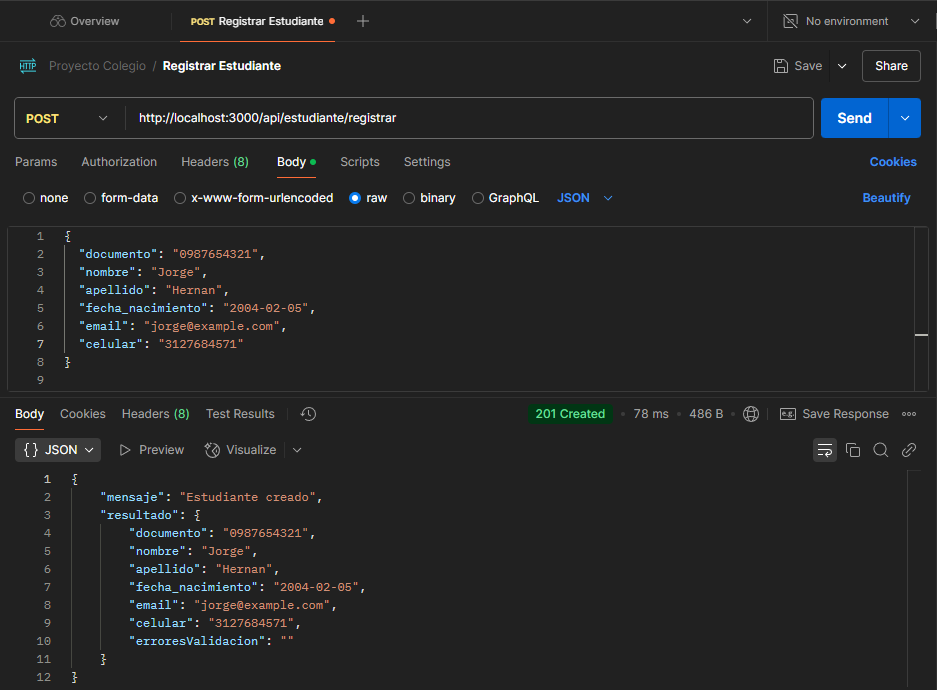
### /api/materia\_profesor

Controlador: materiaprofesorControlador.js  
Archivo de rutas: rutasMateriaProfesor.js

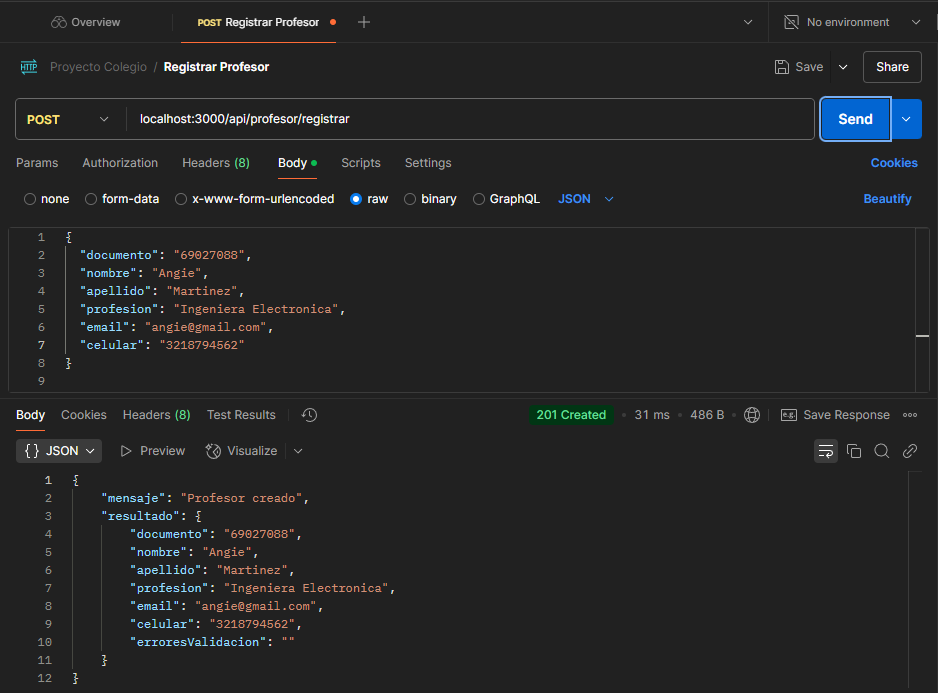
1. **/asignar**  
   Asigna una materia a un profesor.
2. **/listar/:documento\_profesor**  
   Devuelve la lista de materias asignadas a un profesor específico.

# Capturas de pruebas en Postman

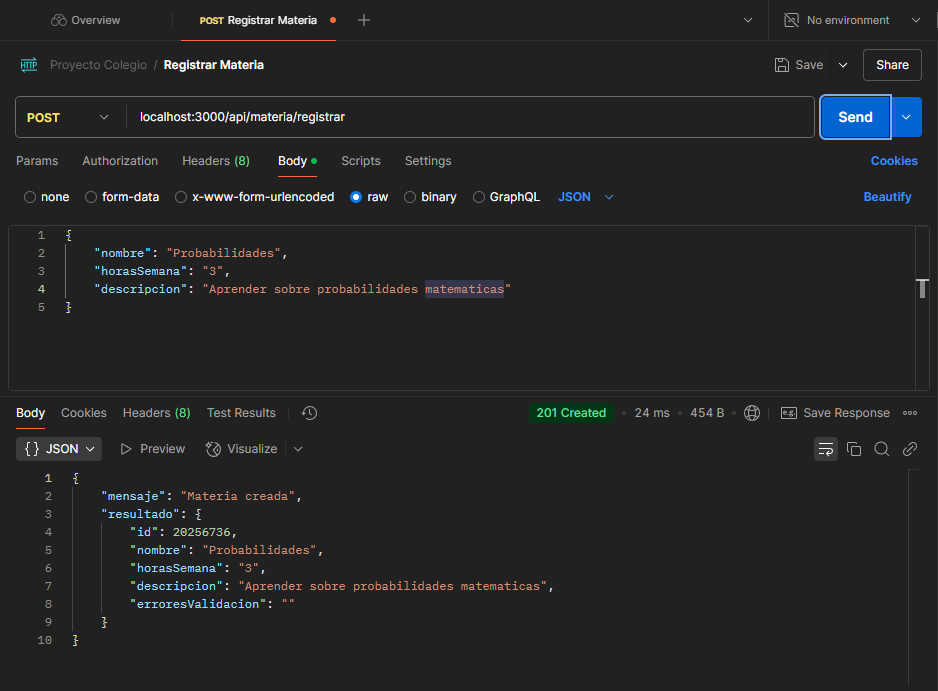
POST /estudiantes – Registrar un estudiante



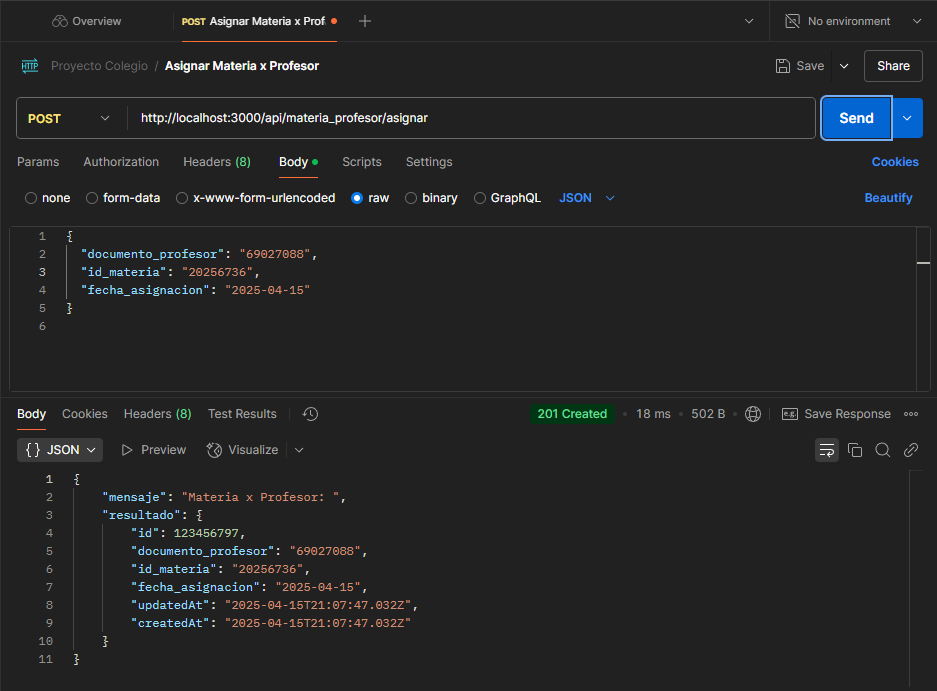
POST /profesores – Registrar un profesor



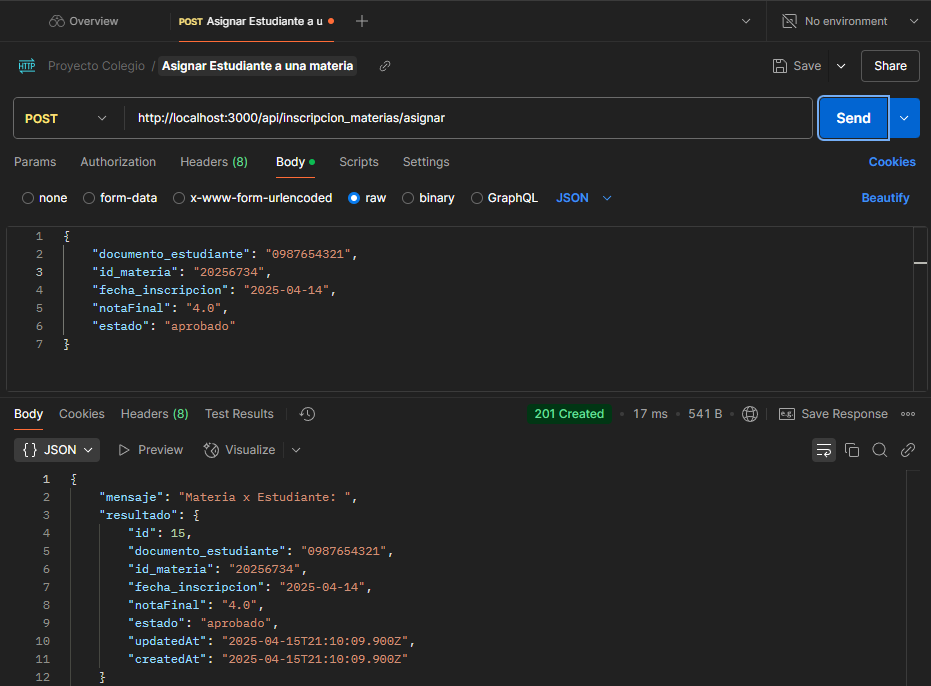
POST /materias – Crear una materia



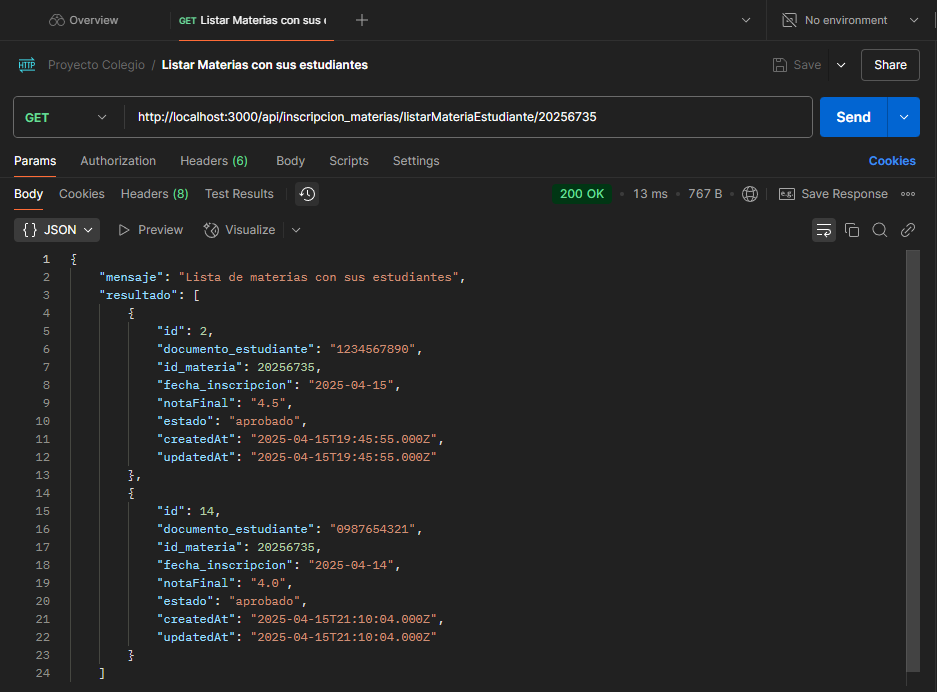
POST /asignar-profesor – Asignar un profesor a una materia



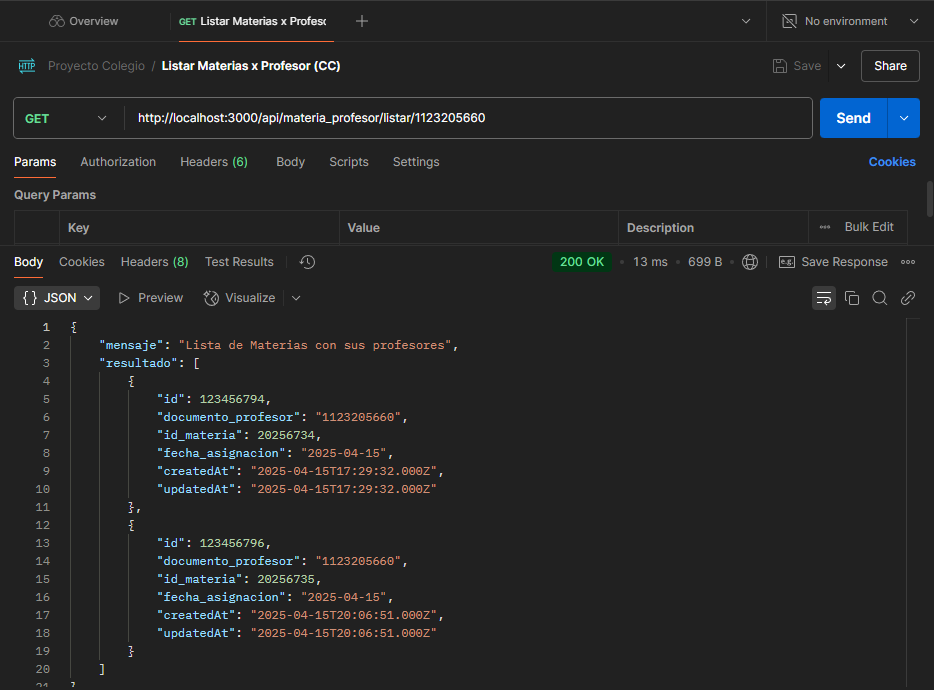
POST /inscribir-estudiante – Inscribir un estudiante en una materia



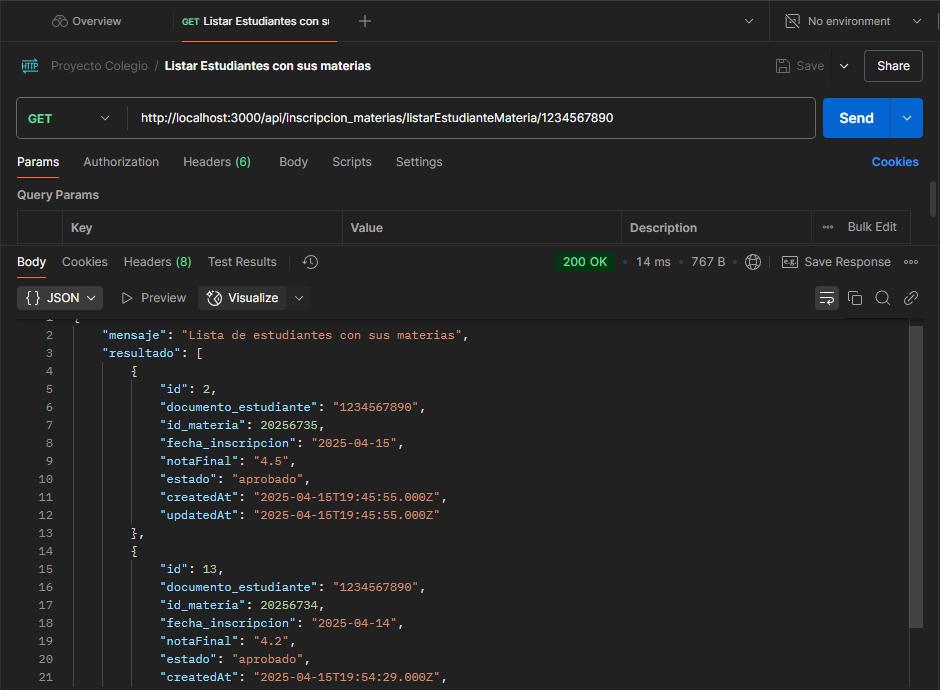
GET /materias/:id – Obtener detalles de una materia y sus inscritos



GET /profesores/:id – Obtener materias dictadas por un profesor



GET /estudiantes/:id – Obtener materias inscritas de un estudiante



# Conclusiones y dificultades encontradas

La implementación de este sistema permitió reforzar conocimientos en desarrollo backend con Node.js, manejo de bases de datos relacionales, validación de datos y estructura modular de aplicaciones. Una de las principales dificultades fue la sincronización de modelos y las relaciones entre tablas al utilizar Sequelize, así como el manejo de errores derivados de validaciones y restricciones únicas en la base de datos.