

**Caso de Estudio (Informe Etapa 2): Microservicios de “EduTech Innovators SPA”**

**Alumnos:**

Franco Pisani

Vicente Riquelme

Ignacio López

**Docente:**

Eduardo Baeza

**Asignatura:**

Desarrollo Fullstack I

26-05-2025

**ÍNDICE**

**1 .Introducción**

**2. Diagrama de Arquitectura de Microservicios**

**3. Estructura del Proyecto**

3.1. Dependencias

3.2. Componentes implemetandos

**4. Base de Datos**

4.1. Motor

4.2. Estructura

**5. Implementación de los Servicios**

**6. Git / Github**

6.1. Comandos utilizados

**7. Conclusión**

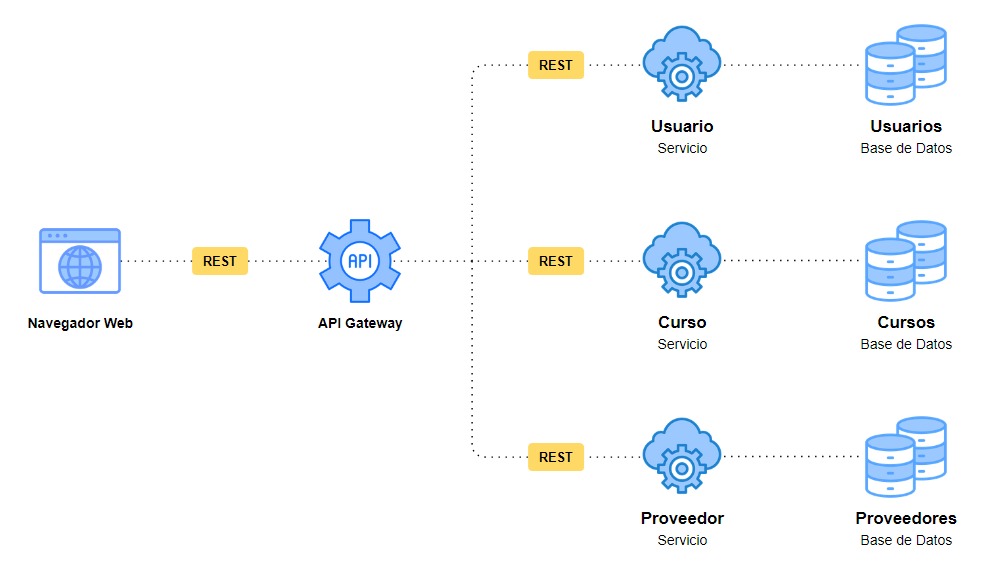
**1. INTRODUCCIÓN**

En este trabajo, que corresponde a nuestra segunda evaluación, desarrollamos una implementación de microservicios utilizando la arquitectura API REST a partir de lo que hemos visto en clase. Particularmente, avanzamos en el diseño e implementación de tres servicios: los de Usuario, Curso y Proveedor.

Desarrollamos este proyecto utilizando herramientas como *SpringBoot* para la inicialización y configuración, *Maven* para las dependencia, *Git y GitHub* para el control de versiones y *Postman* para validar las funciones implementadas. Construimos un sistema modular, pensando en su escalabilidad y mantenibilidad e intentando aplicar buenas prácticas durante el proceso.

Este informe presenta la estructura del proyecto, así como su desarrollo y la validación de funcionalidad y persistencia mediante *Postman*.

**2. DIAGRAMA DE ARQUITECTURA DE MICROSERVICIOS**

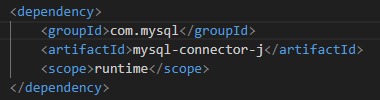
****

**3. ESTRUCTURA DEL PROYECTO**

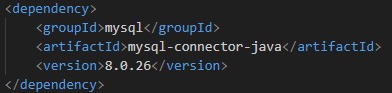
**3.1. Dependencias**

Despues de establecidas las dependencias a traves de Spring Initializr, se modifica pom.xml para utilizar la librería apropiada para MySQL Server.

**ANTES:**

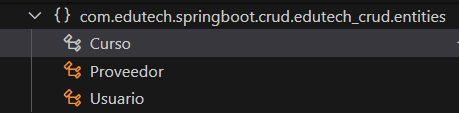


**CON LA MODIFICACIÓN:**

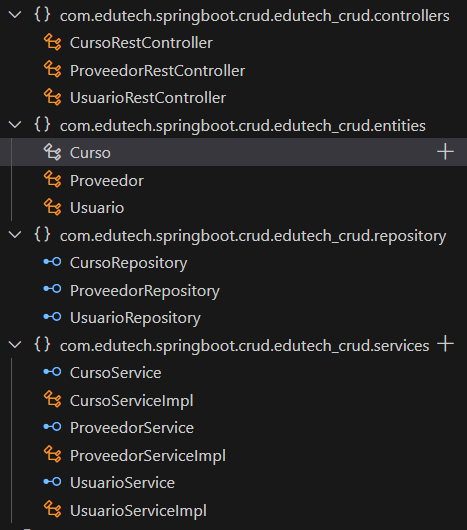


**3.2. Componentes Implementados**

Se implementaron 3 clases para este proyecto, las cuales estan dentro de su package llamado Entities, el que alberga todo el codigo de las clases, junto con los atributos. Estas clases son: **Usuario** (que almacenará informacion tanto de los alumnos como la de los instructores), **Curso** (que almacena la asignatura a la que corresponde cada curso, descripcion y el instructor que la imparte) y **Proveedor,** que almacena la información importante de estos, como el rut, la razón social, el giro y el email para contacto.



Ademas, se crearon 3 packages, **Controllers**, componente encargado de recibir, procesar y responder a las solicitudes del usuario o del cliente, **Services**, que contiene la logica de la aplicación y **Repository**, encargado de acceder a la base de datos. Estos existen para cada una de las clases, como se verá en la siguiente captura.



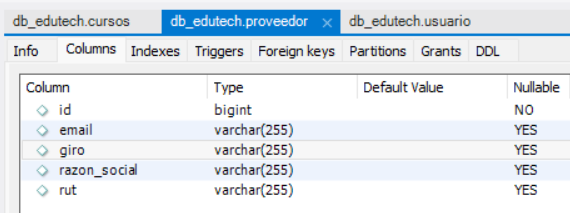
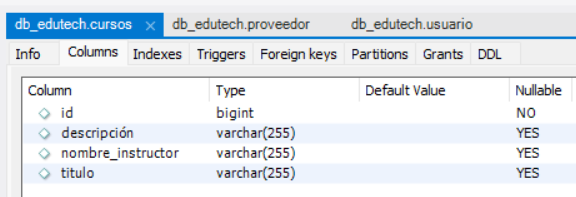
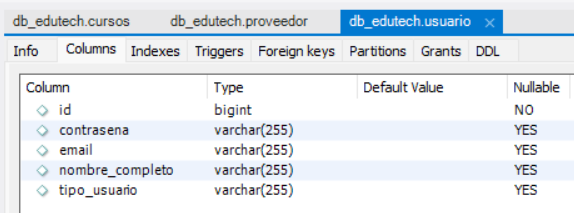
**4. BASE DE DATOS**

**4.1. Motor**

El motor de base de datos utilizado en el codigo es MySQL.

**4.2. Estructura**

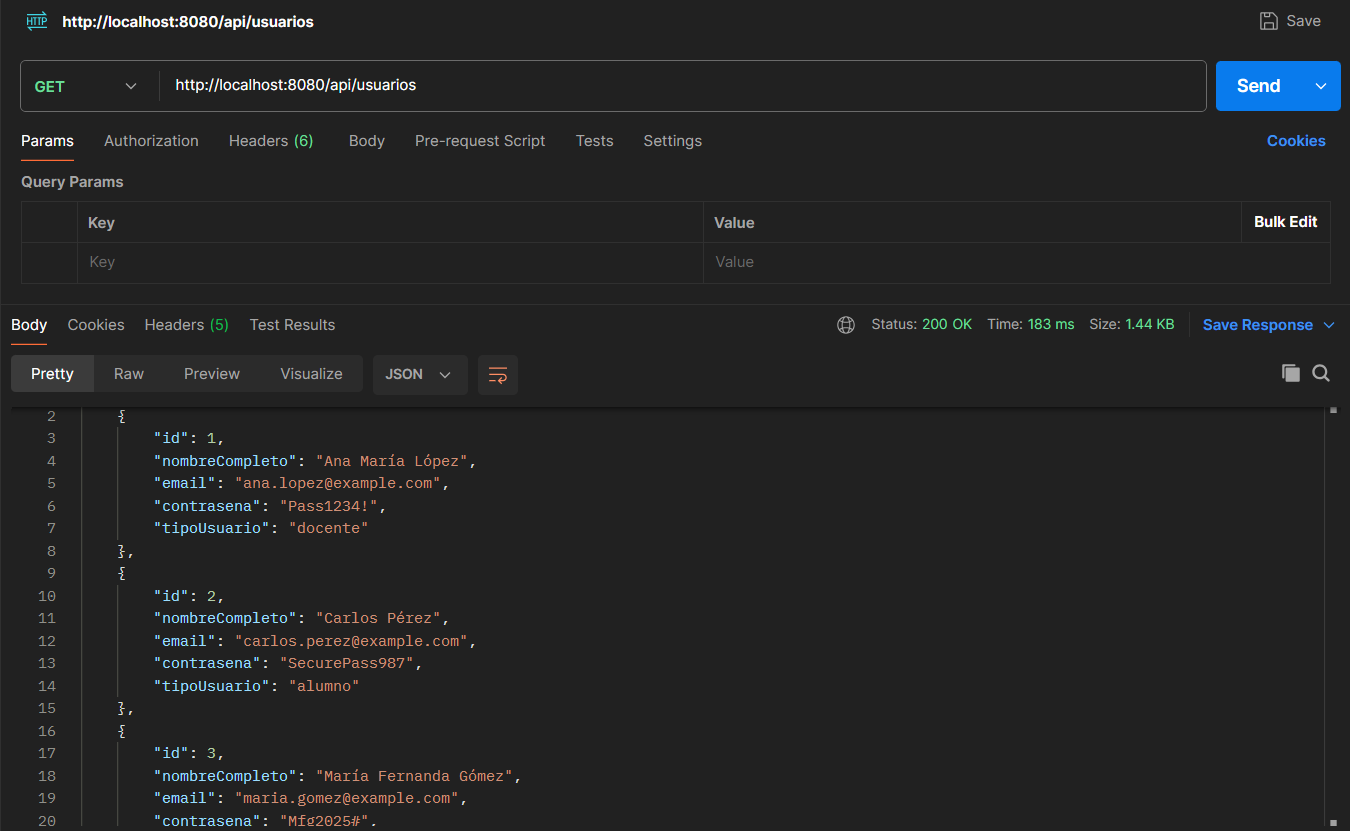
A continuacion se mostrará la estructura de la base de datos, junto con sus tablas y atributos.

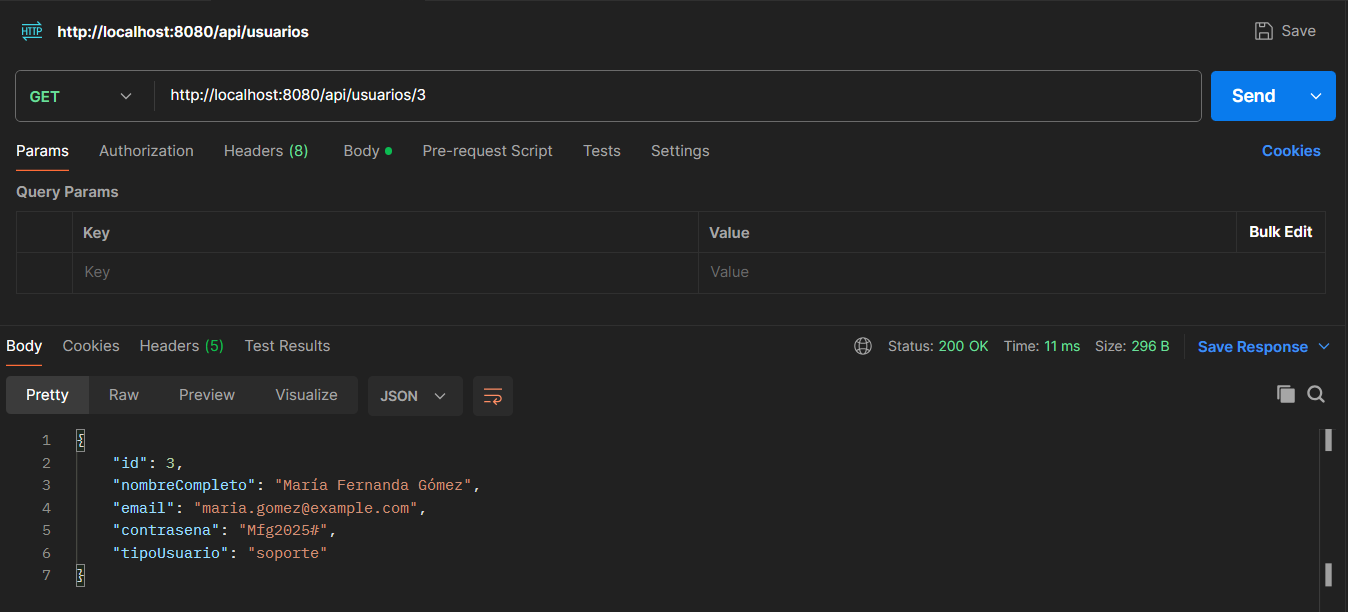


**5. IMPLEMENTACIÓN DE LOS SERVICIOS**

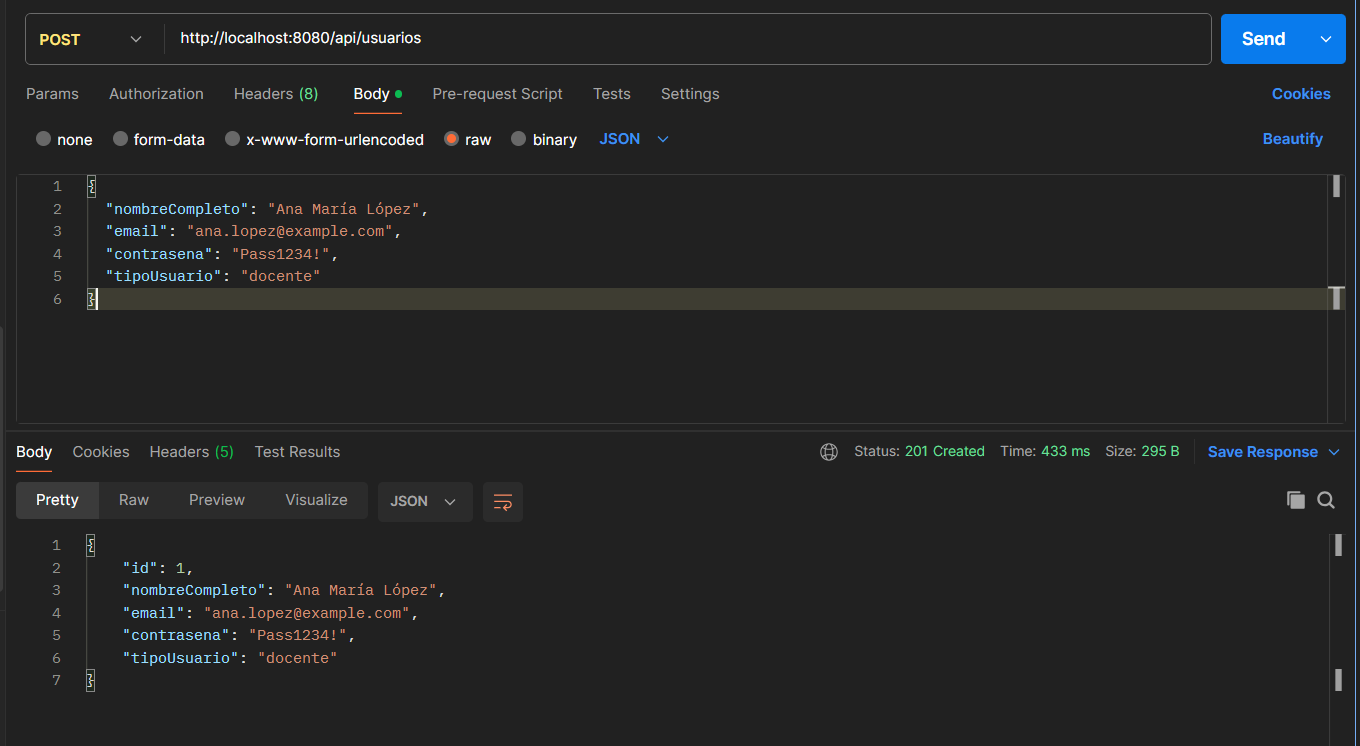
**USUARIO:**

**GET**

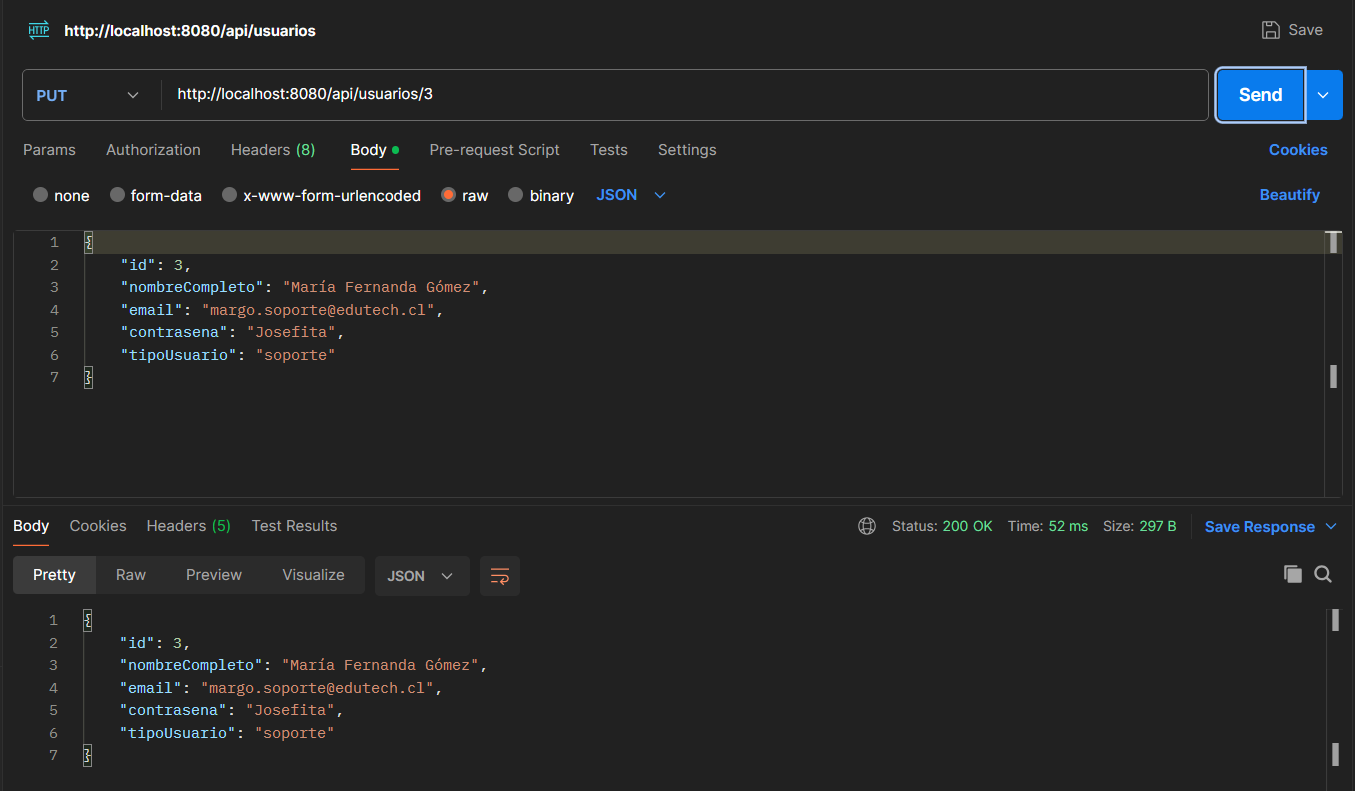


****

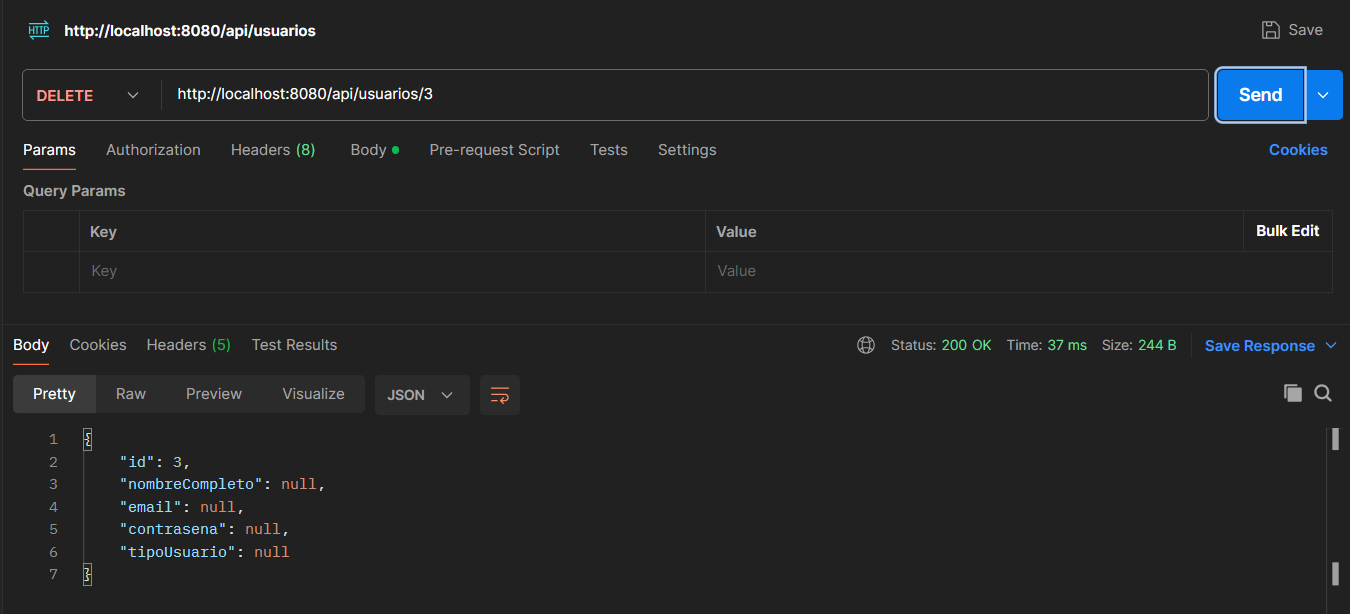
**POST**



**PUT**

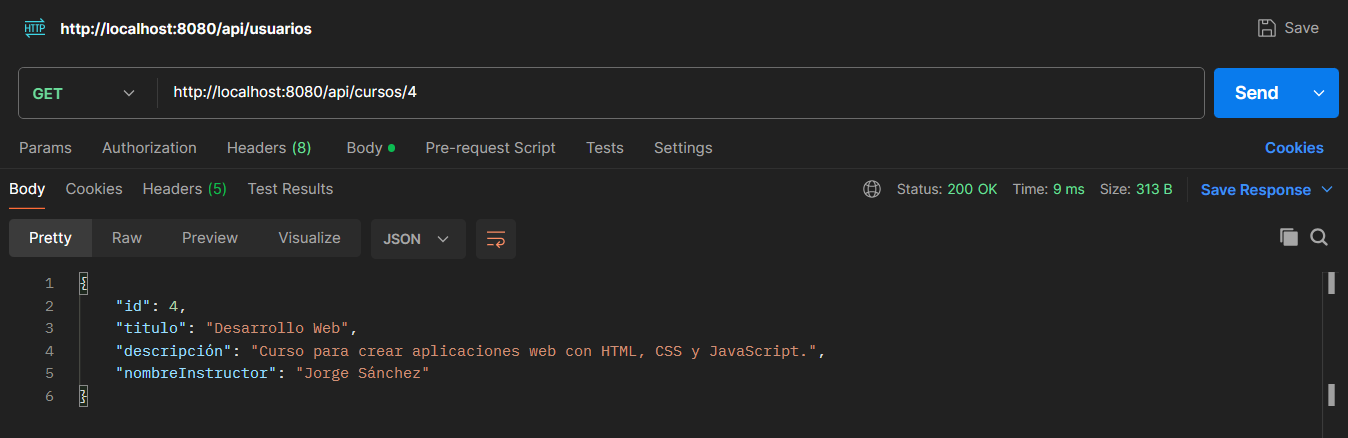
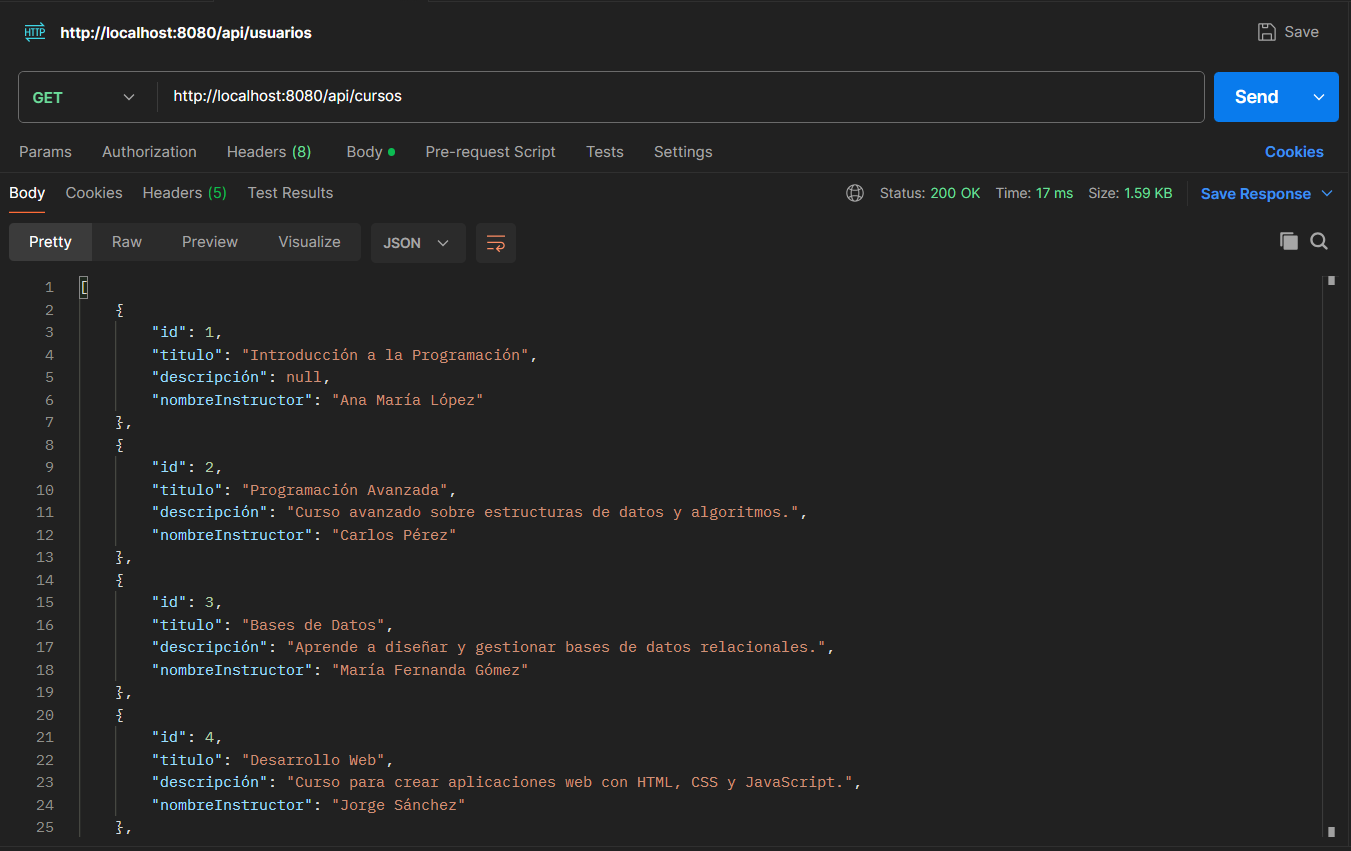


**DELETE**

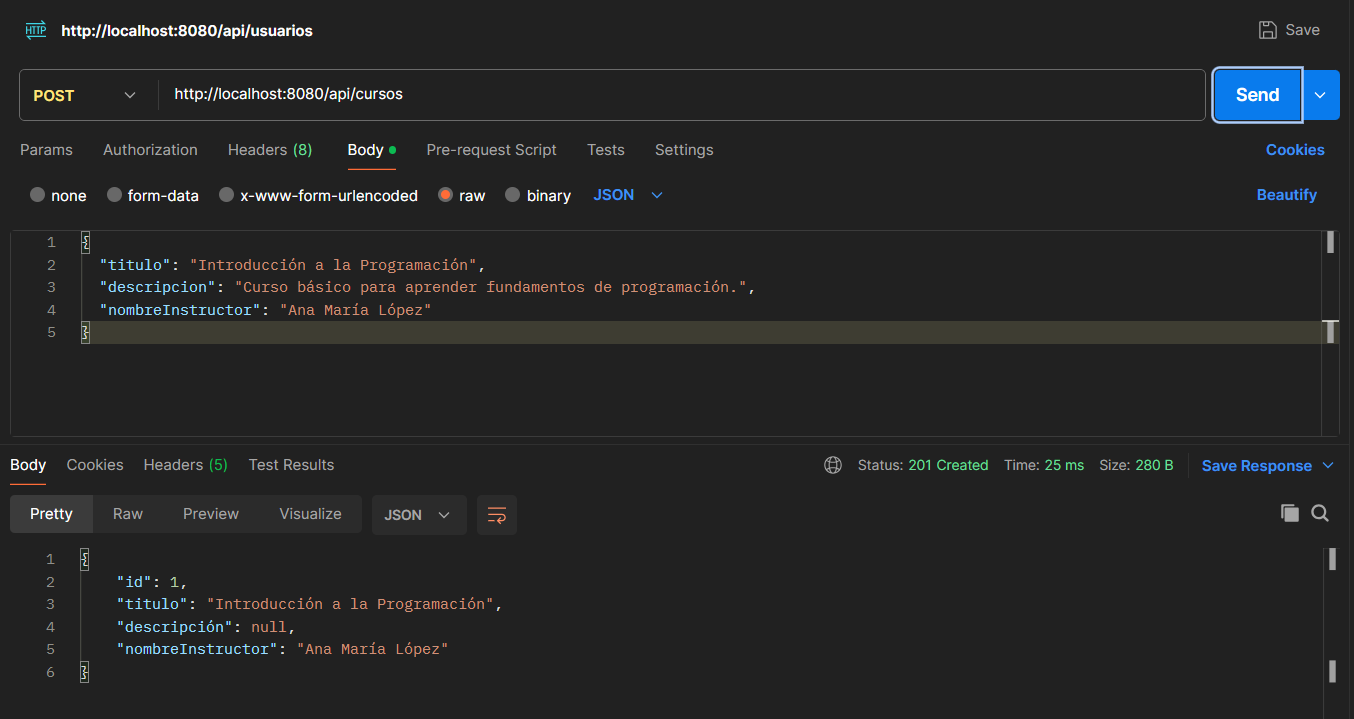


**CURSO:**

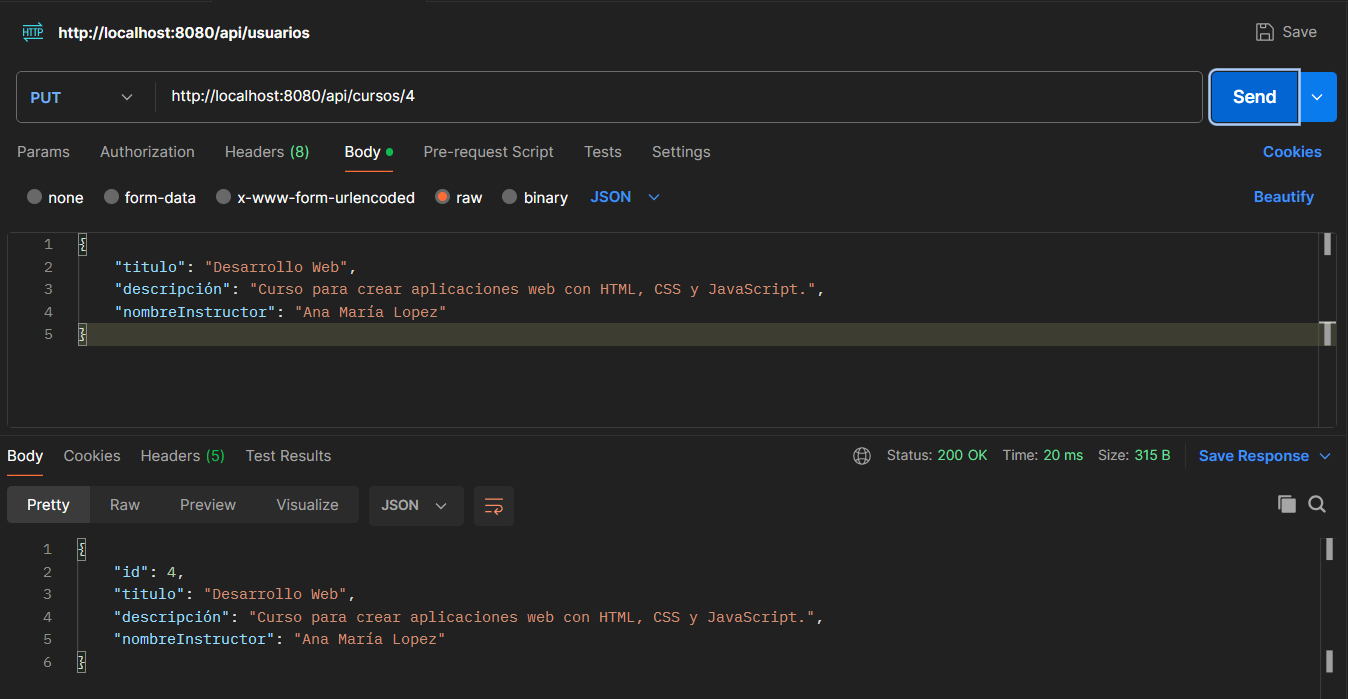
**GET**



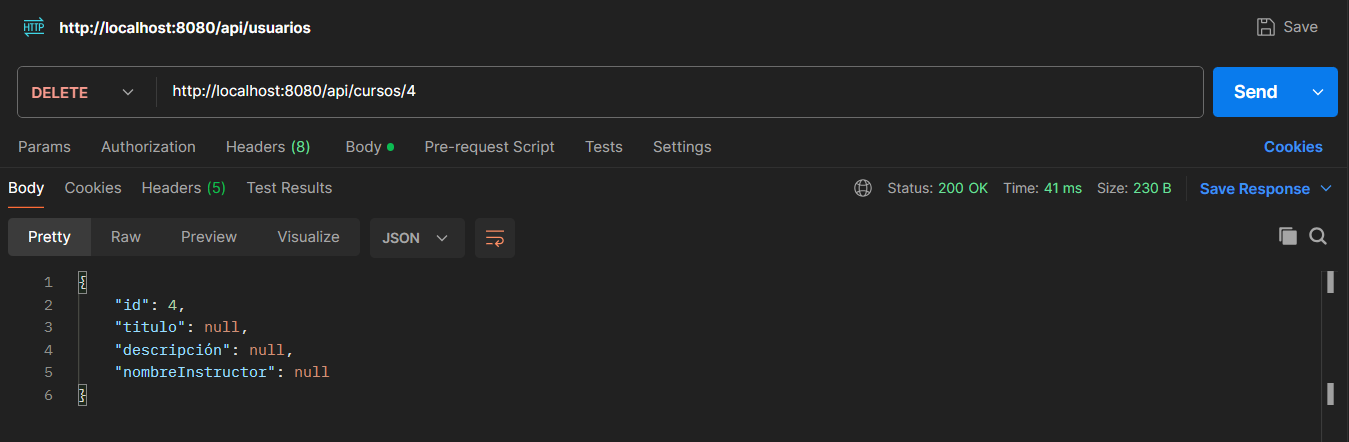
**POST**



**PUT**

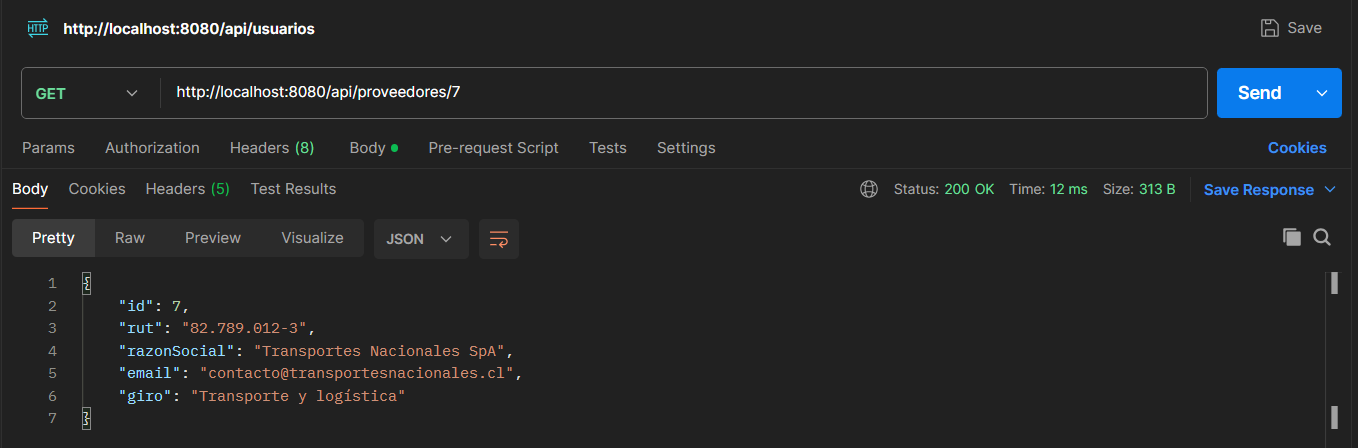
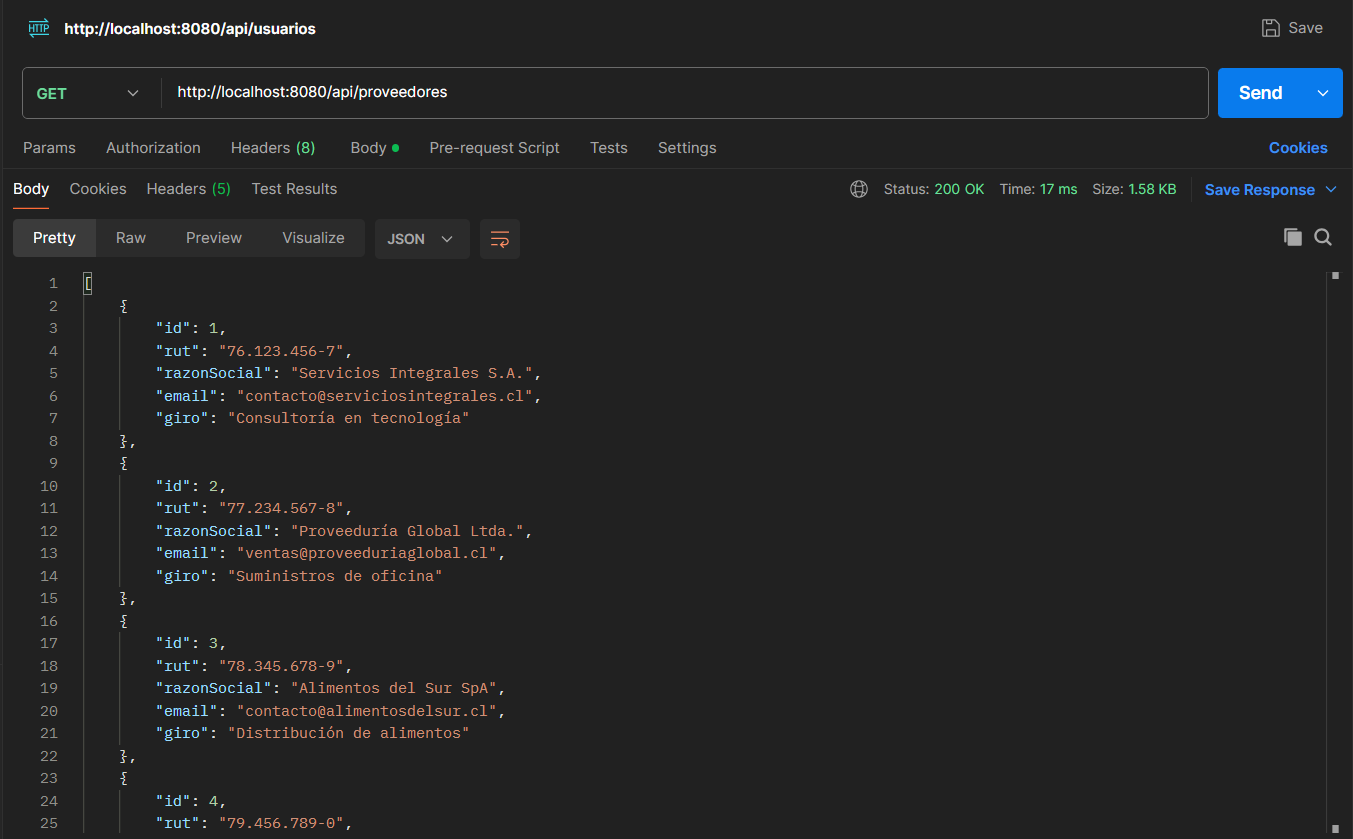


**DELETE**

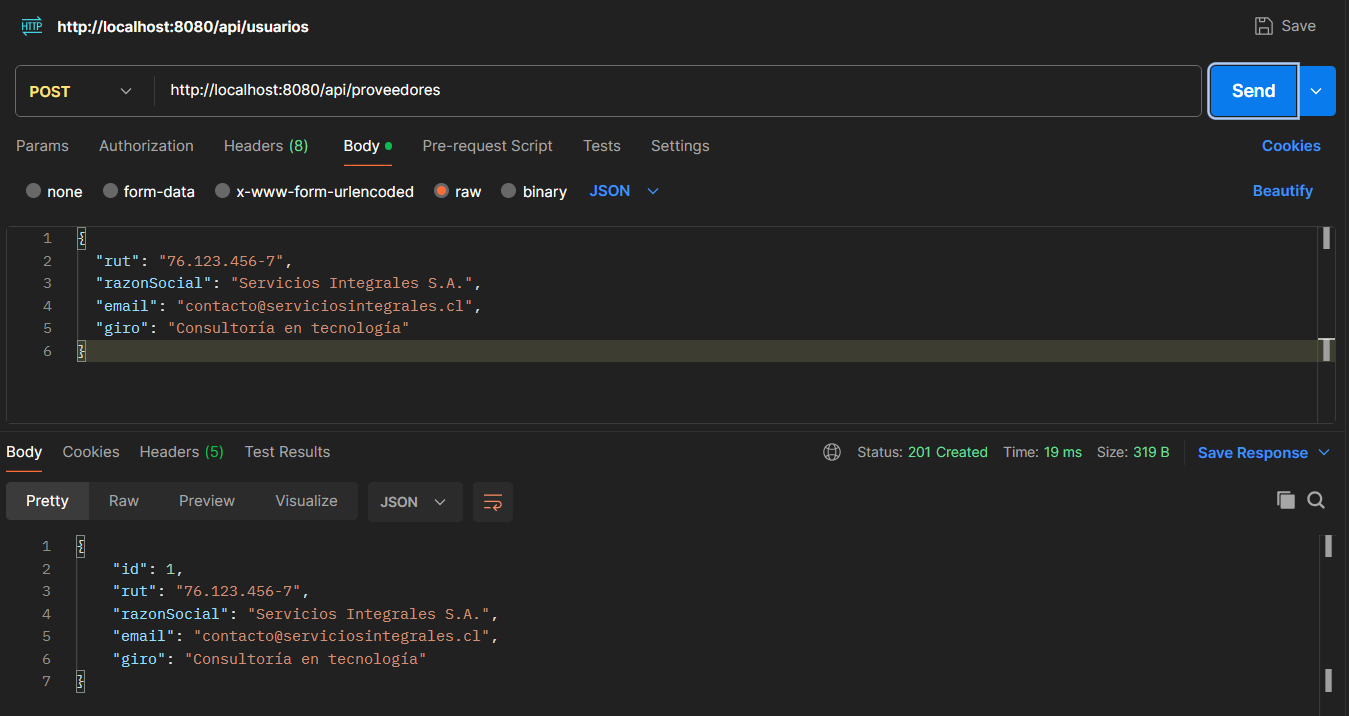


**PROVEEDOR:**

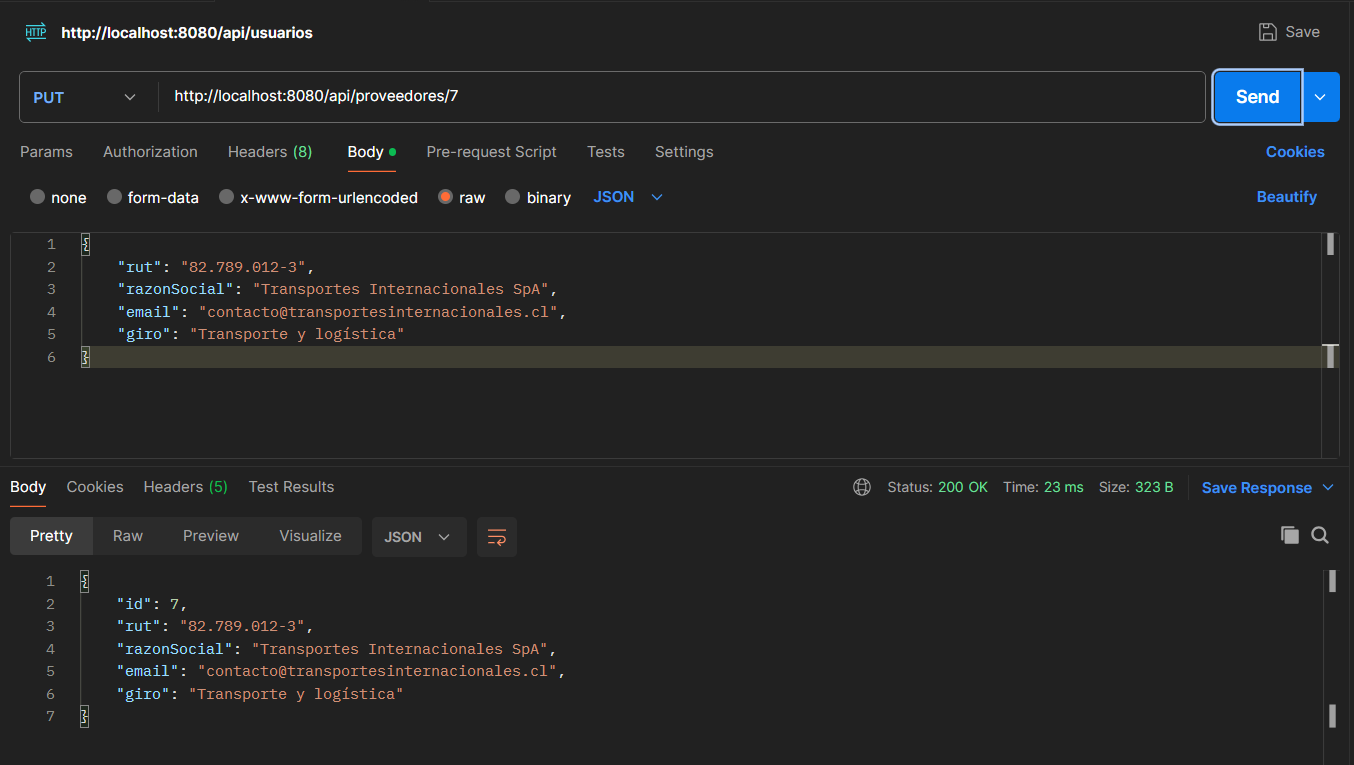
**GET**



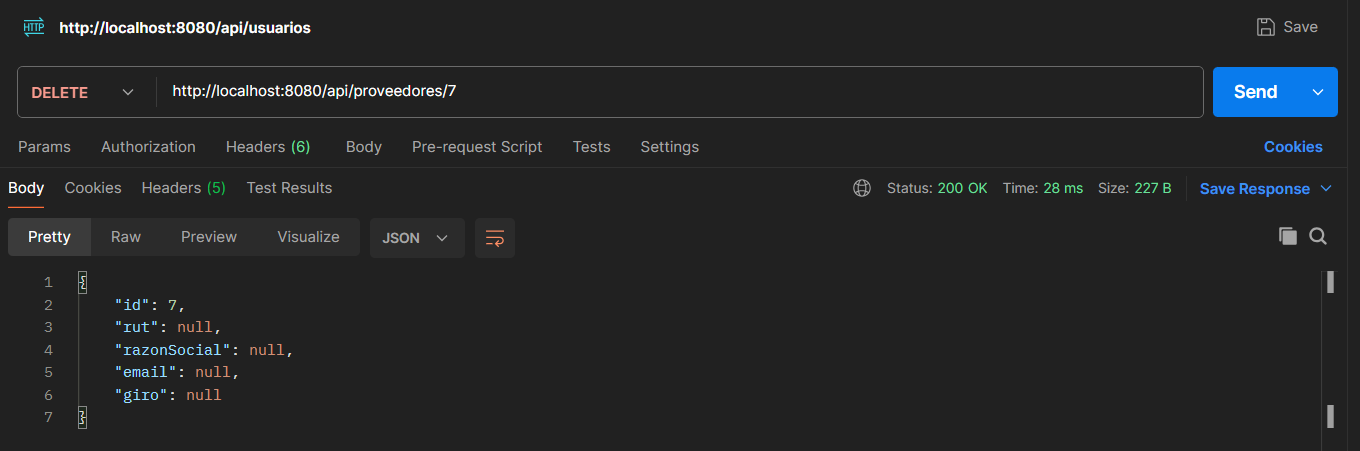
**POST**



**PUT**



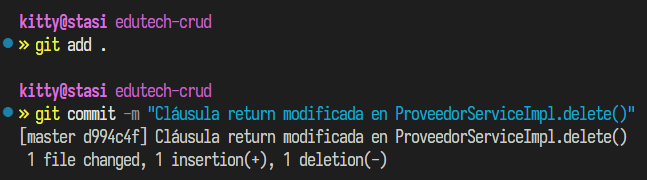
**DELETE**



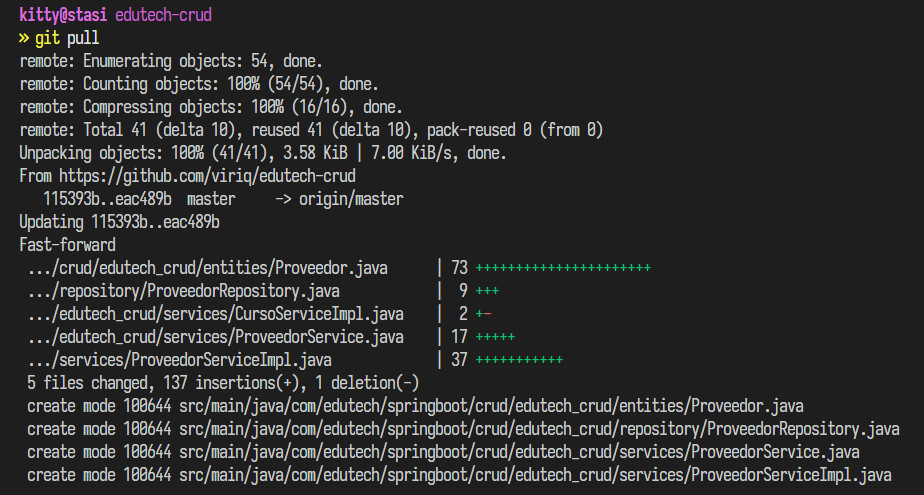
**6. GIT / GITHUB**

**6.1. Comandos utilizados**

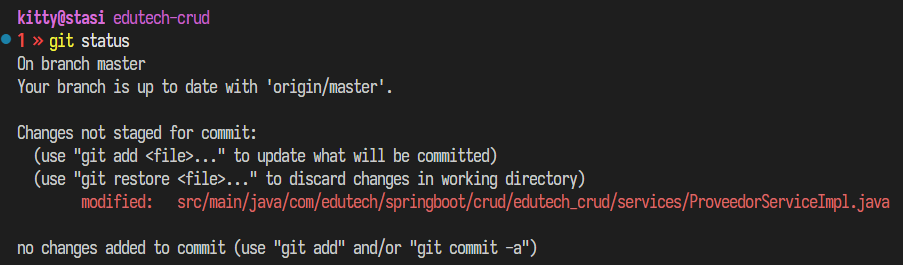
GIT ADD COMMIT



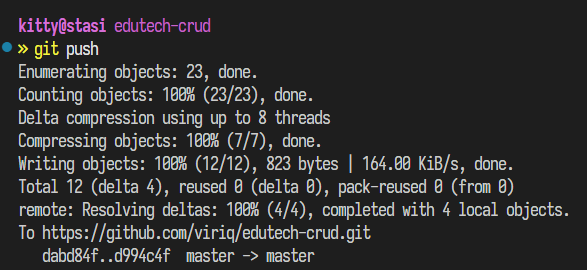
GIT PULL



GIT STATUS



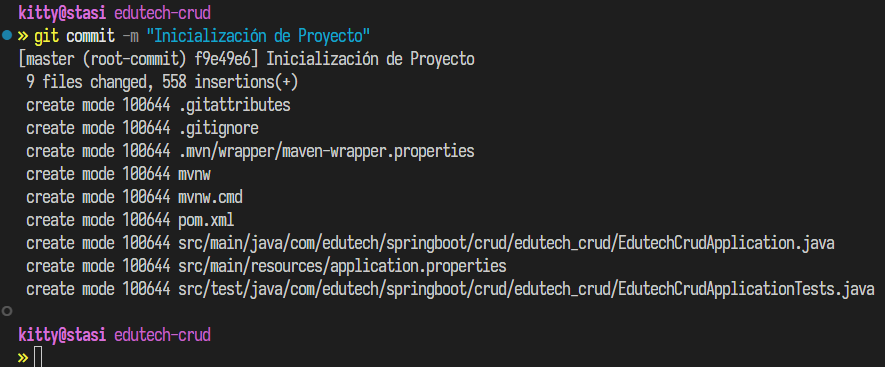
GIT PUSH



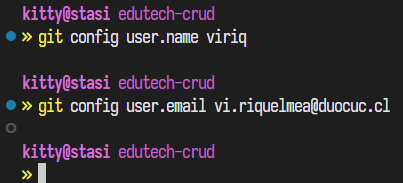
INIT GIT ADD



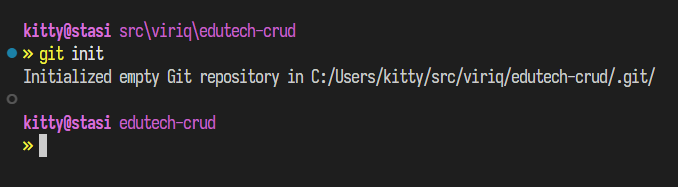
INIT GIT COMMIT



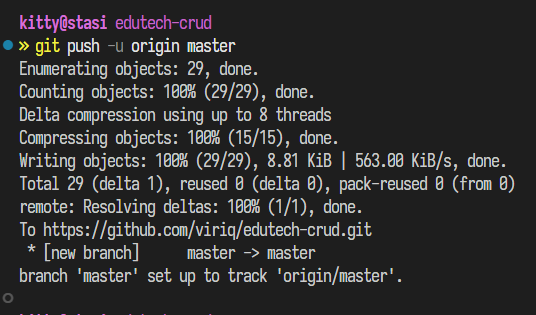
INIT GIT CONFIG



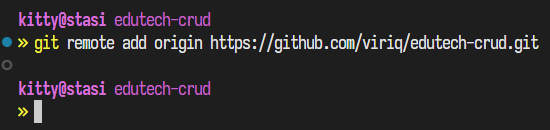
INIT GIT INIT



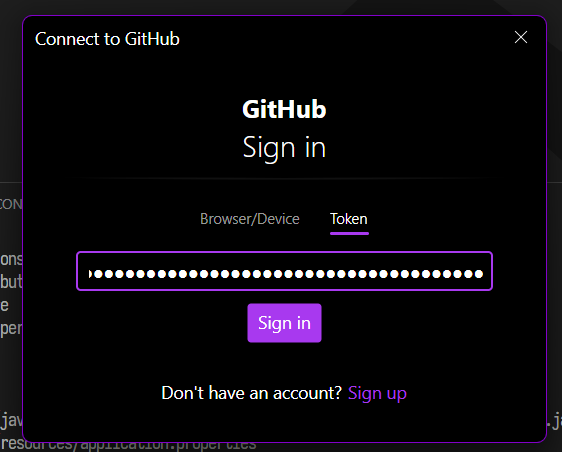
INIT GIT PUSH U



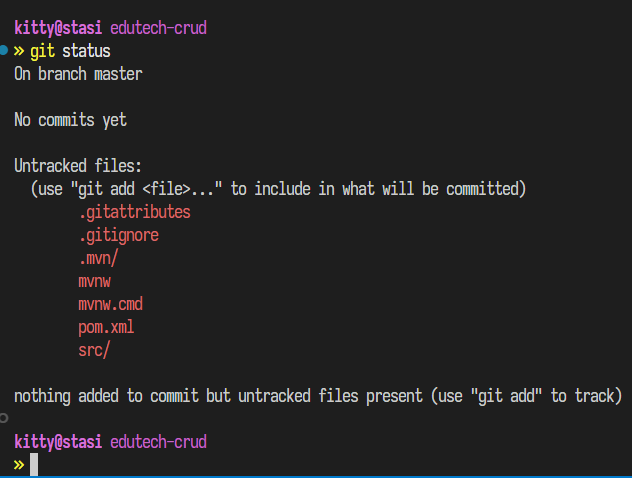
INIT GIT REMOTE ADD ORIGIN



INIT GIT SIGNIN TOKEN



INIT GIT STATUS



**CONCLUSIÓN**

Esta segunda etapa del proyecto nos permitió profundizar en nuestros conocimientos sobre la arquitectura de microservicios, su implementación práctica mediante el uso de *SpringBoot* y la validación de servicios REST utilizando *Postman.*

Utilizamos *Git y GitHub* como nuestro sistema de control de versiones, facilitando el trabajo colaborativo y mejorando nuestra capacidad de organización de nuestro trabajo. No fue necesario juntarnos en un mismo lugar; tuvimos la capacidad de trabajar en nuestro propio tiempo, limitando las reuniones al diseño y organización de nuestro esfuerzo.

Esta experiencia logró fortalecer nuestras habilidades técnicas y de trabajo en equipo, ojalá preparándonos para enfrentar los desafíos del campo laboral. Formamos una base sólida para continuar nuestro desarrollo como profesionales.