



Ingeniería en  
Sistemas Computacionales  
TecNM Campus San Martín Texmelucan

## **DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

### **NOMBRE DE LA MATERIA: TECNOLOGIAS EMERGENTES DE APIS**

Título del Trabajo: PRACTICA 3 THUNDER CLIENT

**PRESENTA:**

**JULIO CESAR RUIZ PEREZ**

**Grupo: 8B**

**DOCENTE: VIANNEY MORALES ZAMORA**

Declaro que el siguiente trabajo de evaluación es el fruto de análisis y trabajo personal. El uso de todos los recursos como citas, ilustraciones, scripts de terceros, se menciona de forma clara y estricta su origen, tanto en el cuerpo del texto como en la bibliografía. En este sentido, soy (somos) consciente(s) de que el hecho de no respetar los derechos de autor y hacer plagio, son objeto de sanciones universitarias y/o legales

Julio Cesar Ruiz Pérez

## Contenido

INTRODUCCIÓN .....	2
DESARROLLO APIS .....	3
CONCLUSIÓN .....	11



## INTRODUCCIÓN

En el siguiente documento de Word se presenta la practica 3 de la materia de tecnologías emergentes de Apis, la cual consistía en poder hacer el CRUD de estudiantes y maestros mediante Thunder Client. Esta práctica tiene como objetivo entender cómo interactuar con una API RESTful mediante las operaciones básicas de un CRUD, permitiendo realizar consultas, inserciones, ediciones y eliminaciones en la base de datos.

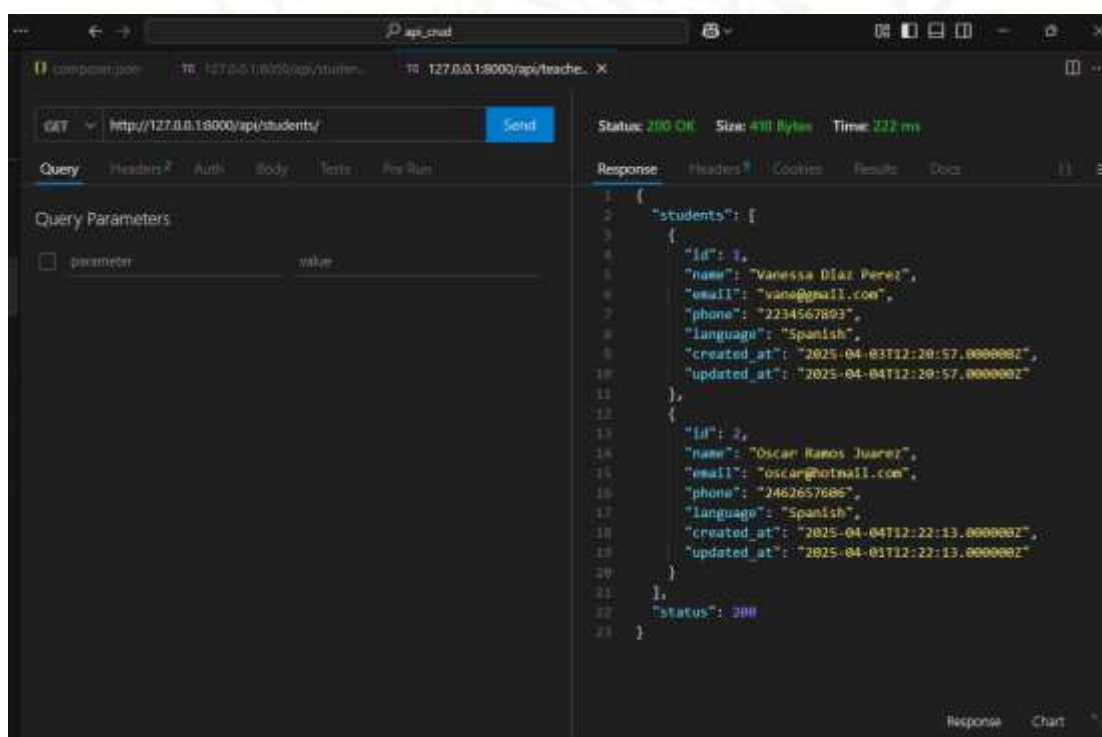
En este documento se detalla el proceso de cada una de estas operaciones, tanto para estudiantes como para profesores, incluyendo los resultados obtenidos en Thunder Client y las configuraciones necesarias para cada petición.

## DESARROLLO APIS

Primero hicimos una consulta desde Thunder Client para ver todos los estudiantes que ya estaban registrados en la base de datos. Esta petición se realizó utilizando el método GET y se ejecutó correctamente mostrando un listado con los datos completos de cada estudiante.

**Figura 1.**

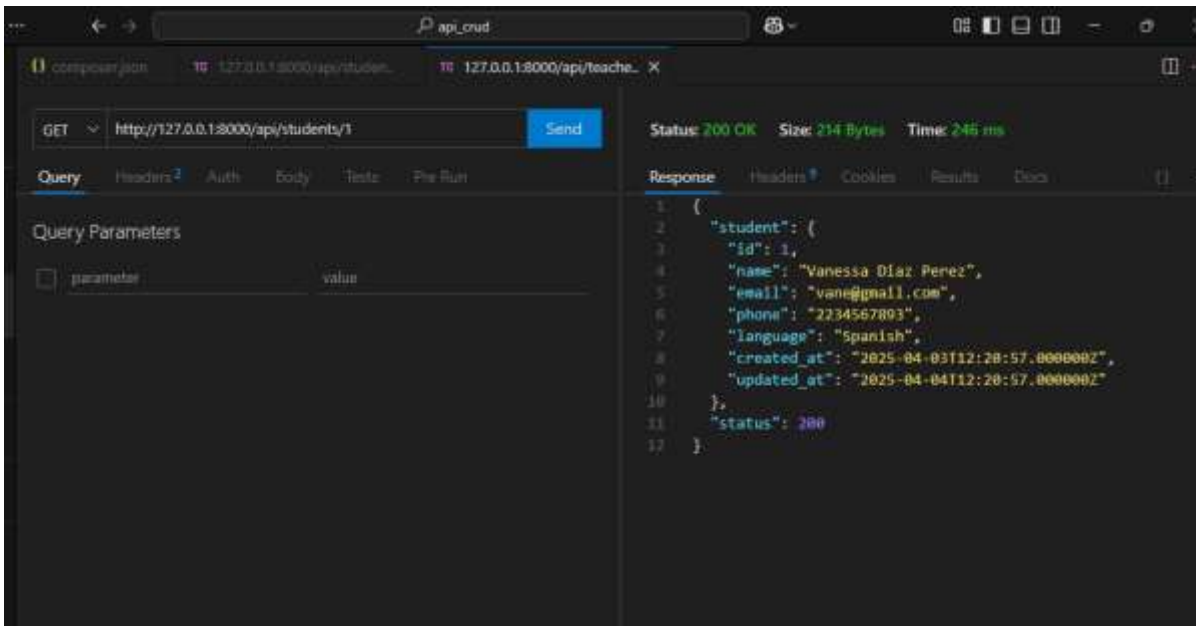
*Consulta con método GET para obtener todos los estudiantes en Thunder Client*



Después, realizamos una segunda consulta, pero esta vez especificando el ID del estudiante. Esto nos permite obtener la información de un solo registro. Usamos el método GET junto con la ruta `http://127.0.0.1:8000/api/students/1`, y así recuperamos los datos del estudiante con ID 1.

**Figura 2.**

*Consulta con método GET para obtener un estudiante específico.*



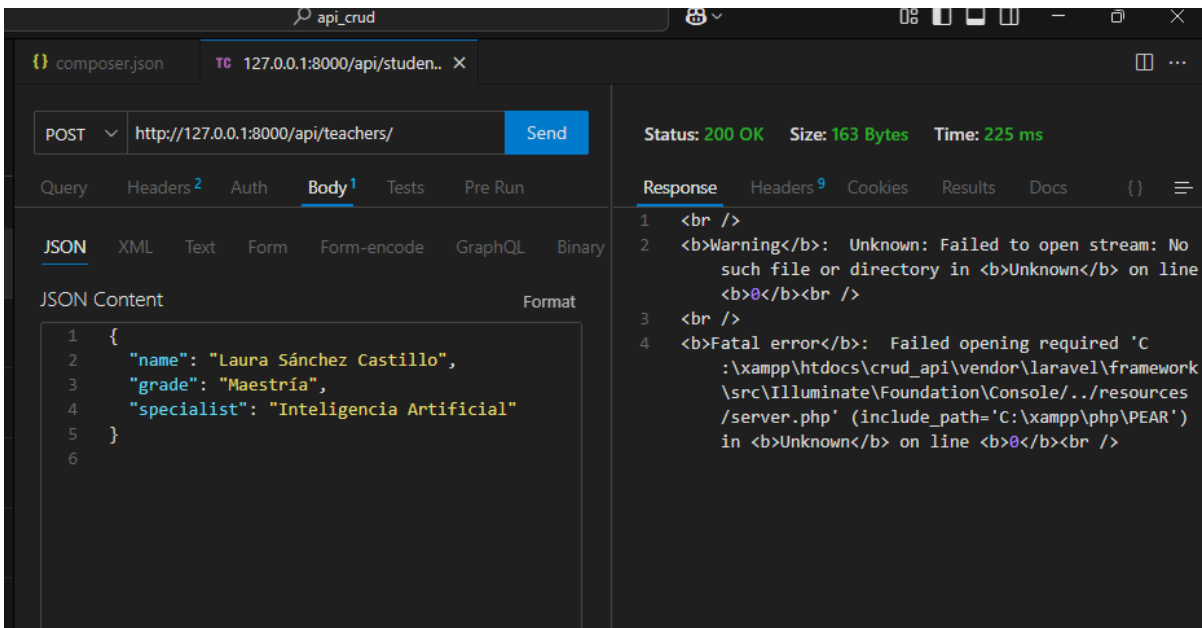
Para probar la funcionalidad de inserción, utilizamos el método POST en Thunder Client. En la petición enviamos los datos de un nuevo estudiante en formato JSON. Al ejecutar esta operación, se registró correctamente el nuevo estudiante en la base de datos.

Este fue lo que enviamos:

```
{  
  "name": "Julio Cesar Ruiz",  
  "email": "julio@gmail.com",  
  "phone": "2223344556",  
  "language": "Spanish"  
}
```

**Figura 3.**

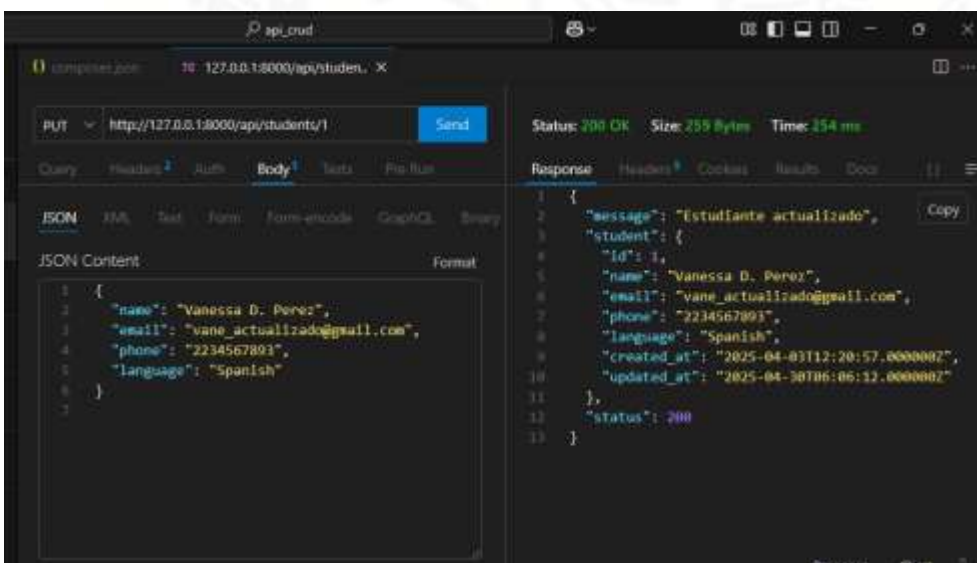
*Petición POST para insertar un nuevo estudiante desde Thunder Client.*



Más adelante, probamos la funcionalidad de edición para actualizar los datos de un estudiante ya registrado. En este caso usamos el método PUT, especificando el ID del estudiante a modificar, y mandamos los nuevos datos en formato JSON.

**Figura 4.**

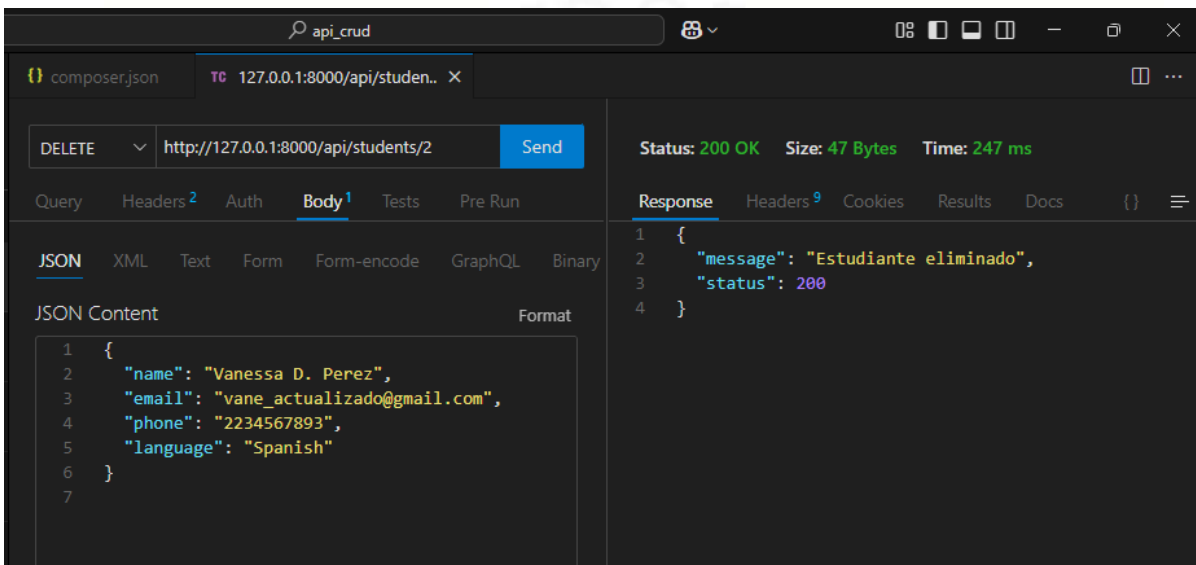
*Petición PUT para editar un estudiante existente desde Thunder Client.*



Por último, realizamos una prueba para eliminar un estudiante de la base de datos. Para esto usamos el método DELETE indicando el ID del estudiante que deseábamos eliminar. La petición fue ejecutada exitosamente y el registro fue removido.

### Figura 5.

*Petición DELETE para eliminar un estudiante desde Thunder Client.*

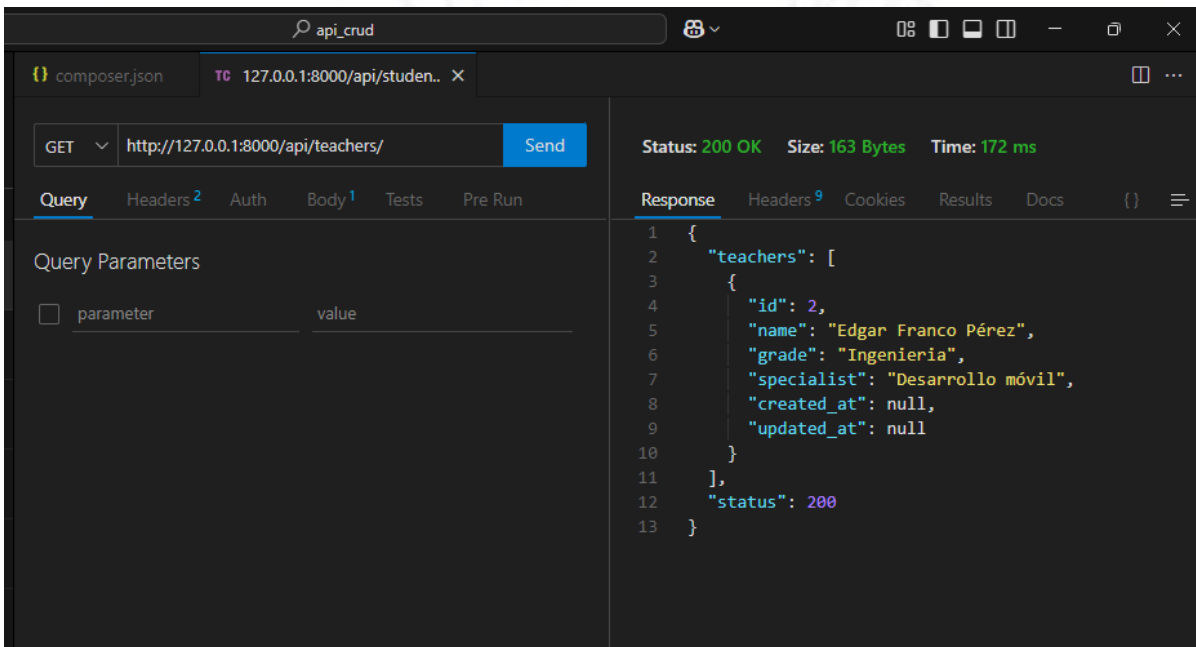


## TAMBIEN REALIZAMOS EL CRUD DE MAESTROS DE LA MISMA MANERA

Primero, realizamos una consulta desde Thunder Client para obtener todos los profesores registrados en la base de datos. Usamos el método **GET**, y con esto obtuvimos un listado con los datos de todos los profesores registrados.

**Figura 1.**

*Consulta con método GET para obtener todos los profesores en Thunder Client.*

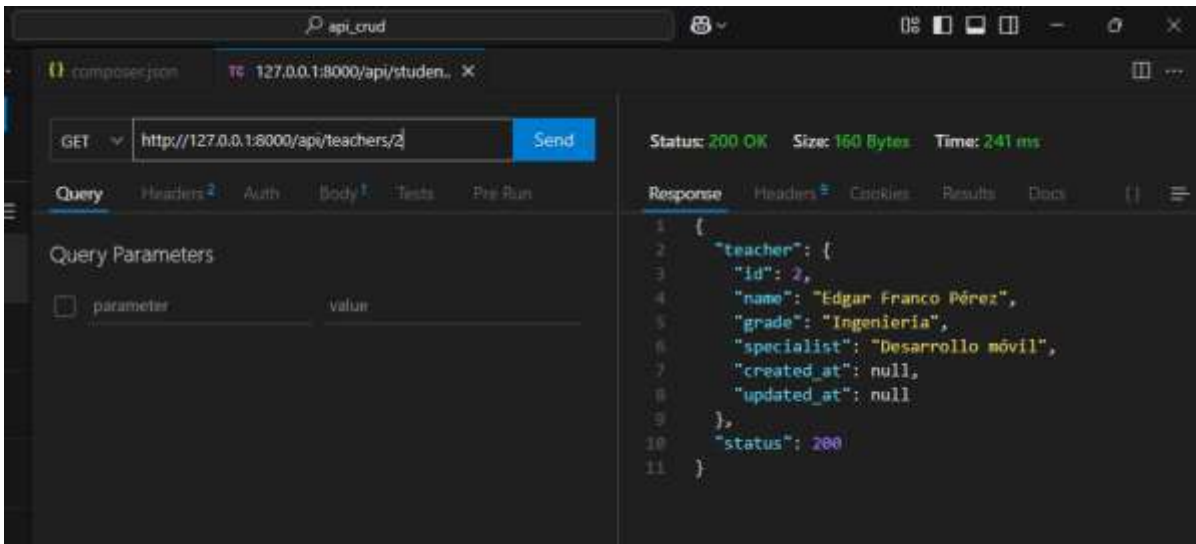


Después, probamos la consulta individual de un profesor usando el método GET y especificando el ID del profesor. De esta forma, podemos recuperar los detalles de un solo registro. En este caso, realizamos la petición con el ID 2, y obtuvimos la información del profesor correspondiente.



**Figura 2.**

*Consulta con método GET para obtener un profesor específico por ID en Thunder Client.*



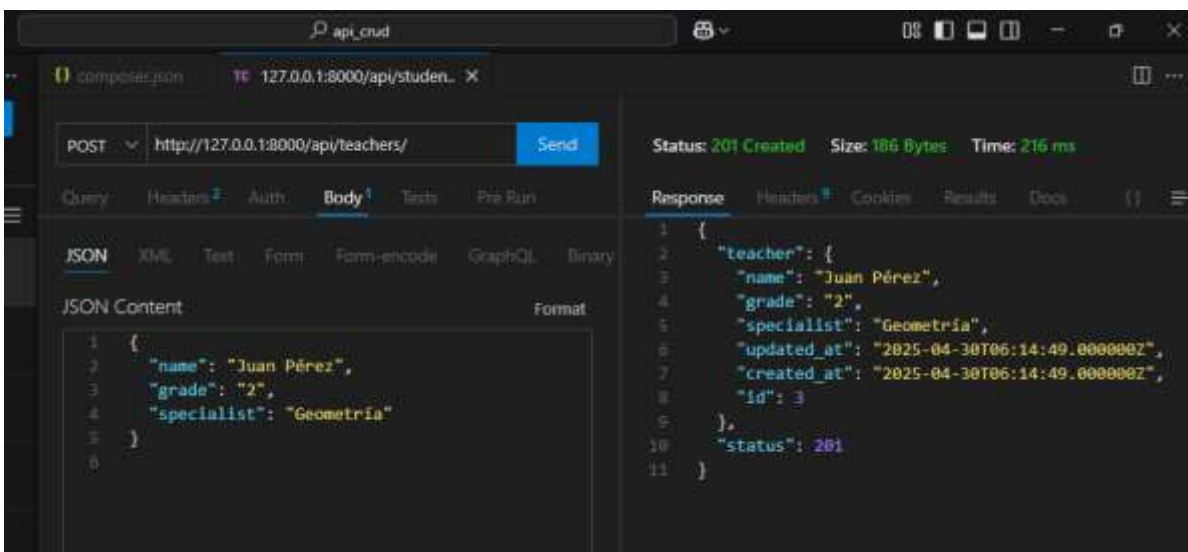
Para probar la inserción de un nuevo docente, utilizamos el método POST en Thunder Client. En la petición enviamos los datos del nuevo profesor en formato JSON. Al ejecutar esta operación, el nuevo profesor fue registrado correctamente en la base de datos.

Este fue el contenido que enviamos:

```
{
  "name": "Juan Pérez",
  "grade": "Matemáticas",
  "specialist": "Geometría"
}
```

**Figura 3.**

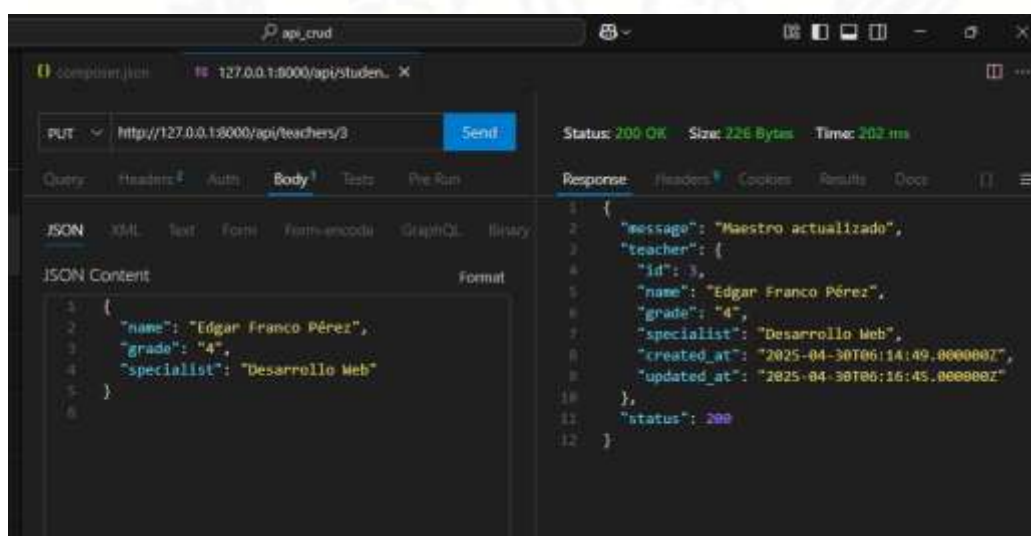
*Petición POST para insertar un nuevo profesor desde Thunder Client.*



Después, realizamos una edición para actualizar los datos de un profesor. Usamos el método PUT y enviamos los datos modificados en formato JSON, junto con el ID del profesor a editar. Al ejecutar esta operación, los datos del profesor fueron actualizados correctamente en la base de datos.

**Figura 4.**

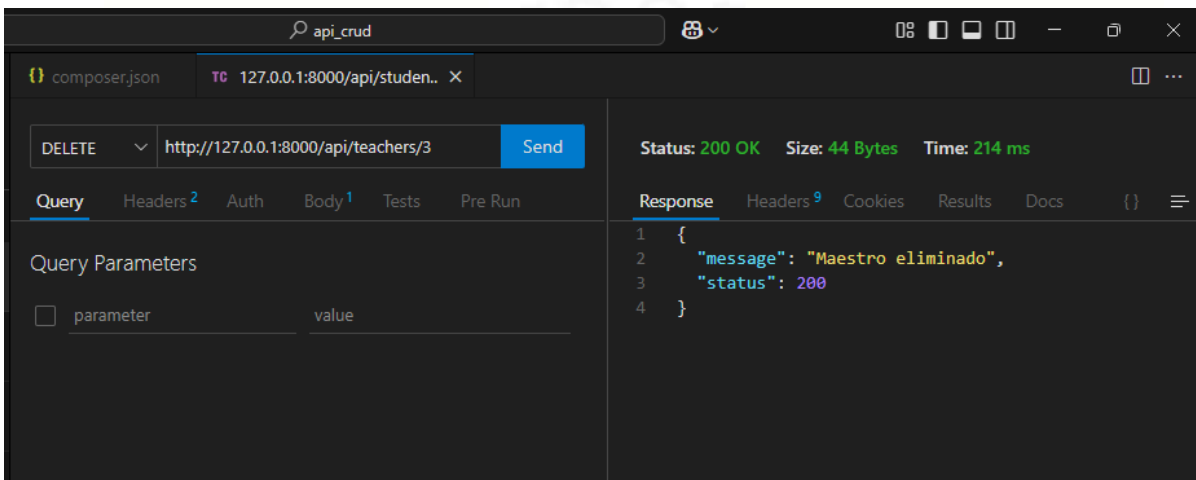
*Petición PUT para editar un profesor existente desde Thunder Client.*



Finalmente, realizamos una prueba para eliminar un profesor de la base de datos. Utilizamos el método DELETE, especificando el ID del profesor a eliminar. La solicitud fue ejecutada exitosamente, y el registro fue removido de la base de datos.

### Figura 5.

*Petición DELETE para eliminar un profesor desde Thunder Client.*



## CONCLUSIÓN

En conclusión, la práctica realizada nos permitió obtener una comprensión más profunda sobre cómo interactuar con una API RESTful utilizando Thunder Client. A través de las operaciones CRUD, pudimos gestionar de manera efectiva la información de estudiantes y profesores en una base de datos.

