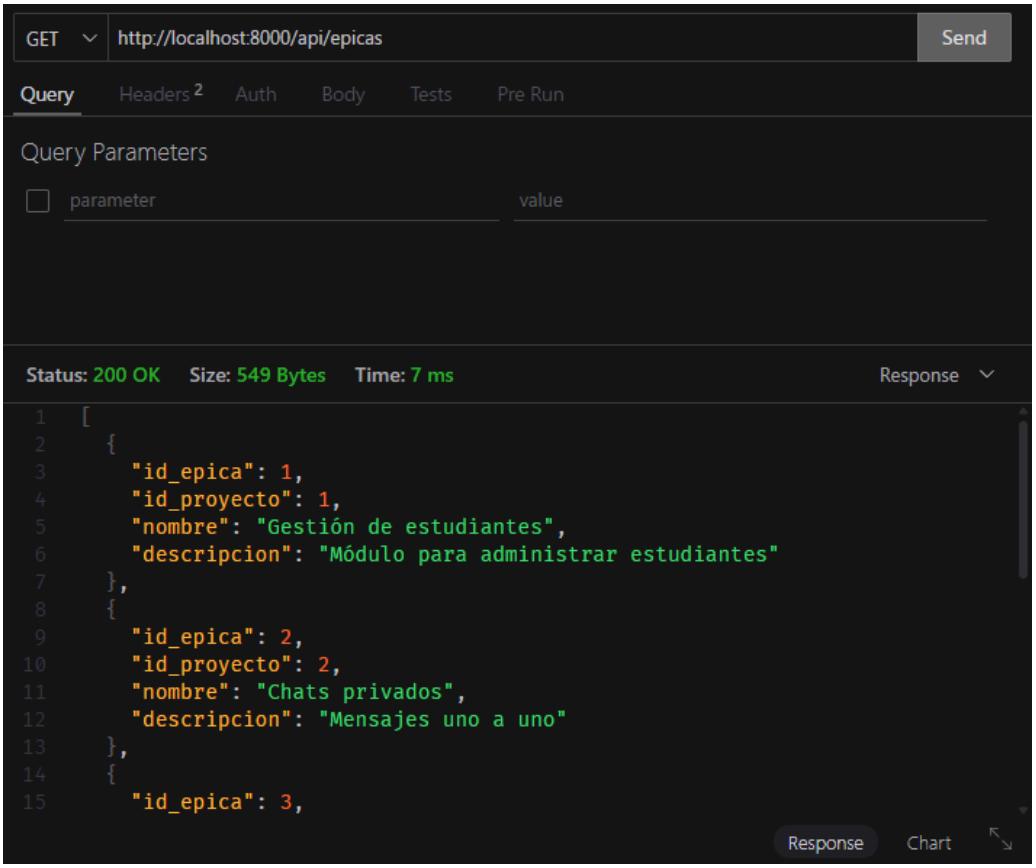
Mariana / Épicas

1. Obtener todas las épicas

GET ✅

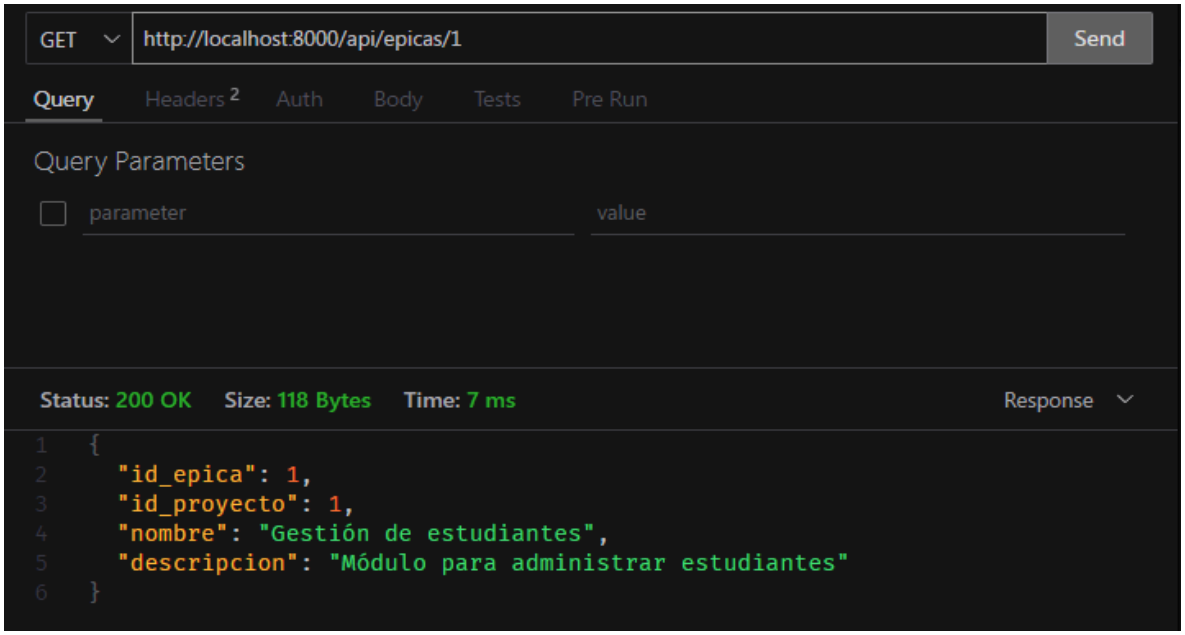
/api/epicas



2. Obtener una épica por ID

GET✅

/api/epicas/{id}



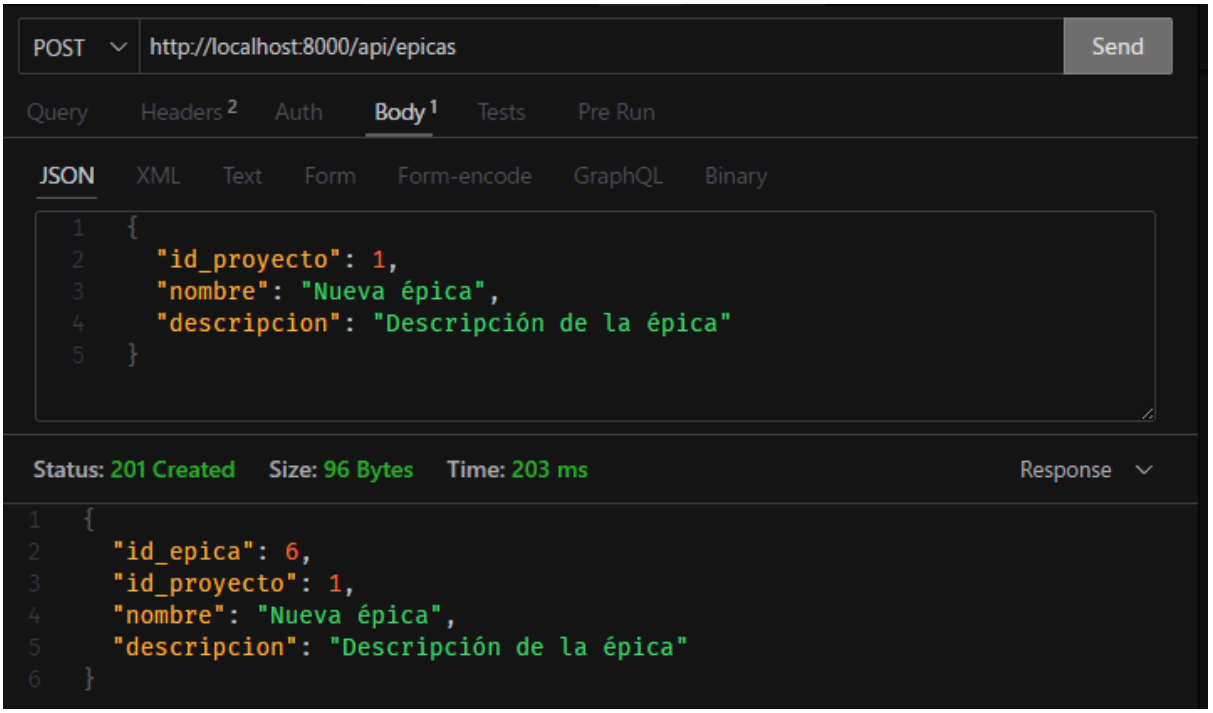
3. Crear una nueva épica

POST✅

/api/epicas

Body (JSON):

{
"id_proyecto": 1,
"nombre": "Nueva épica",
"descripcion": "Descripción de la épica"
}



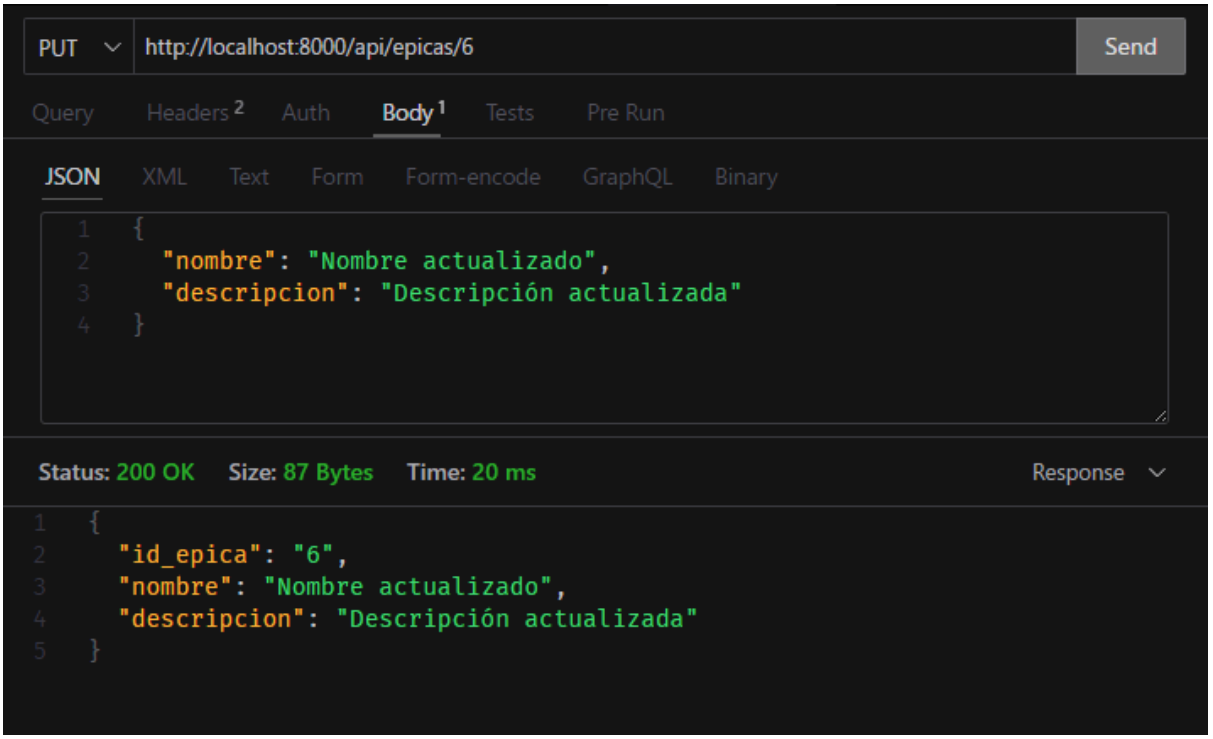
4. Actualizar una épica existente

PUT✅

/api/epicas/{id}

Body (JSON):

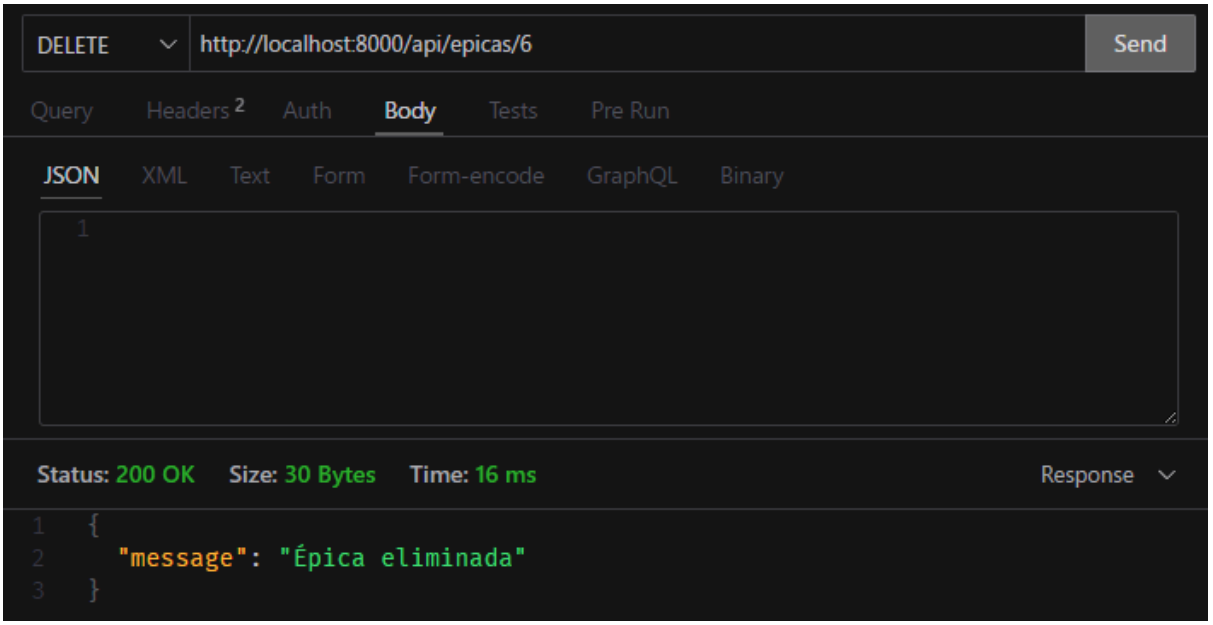
{
"nombre": "Nombre actualizado",
"descripcion": "Descripción actualizada"
}



5. Eliminar una épica

DELETE✅

/api/epicas/{id}

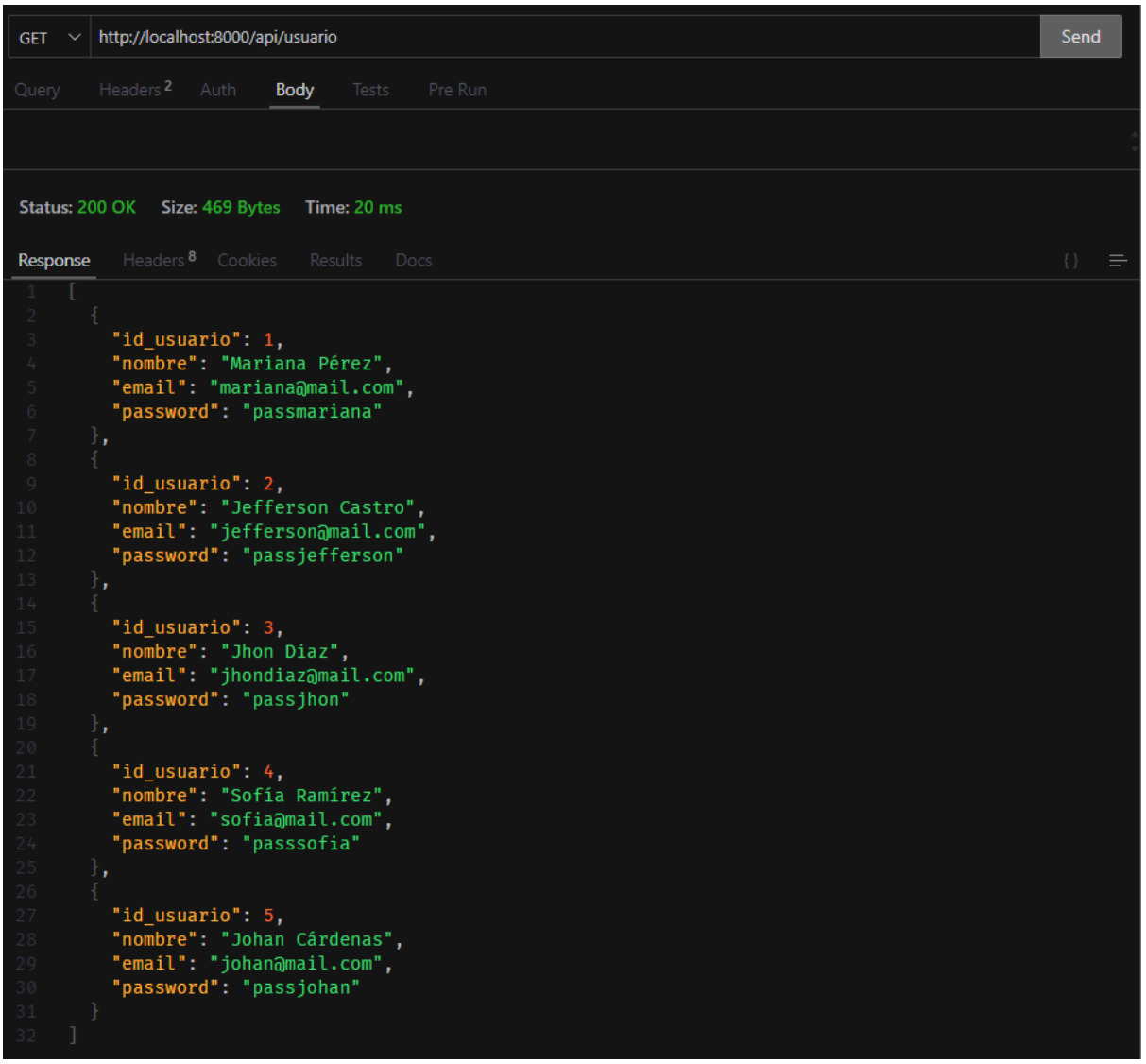


Sofia / Usuario

1. Listar todos los usuarios

GET✅

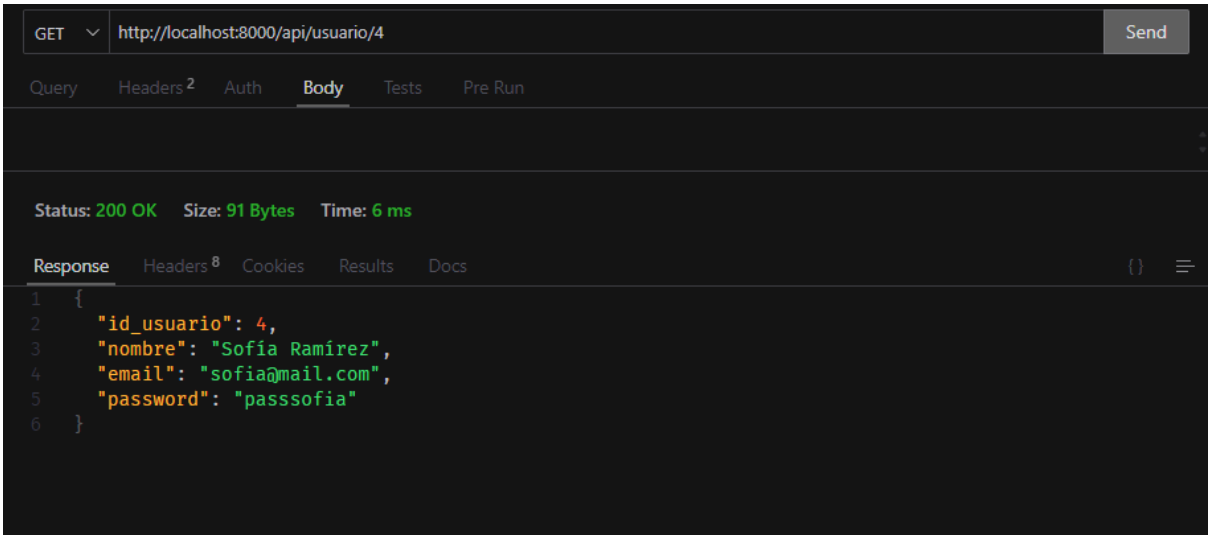
/api/usuario



2. Obtener un usuario por ID

GET✅

/api/usuario/4



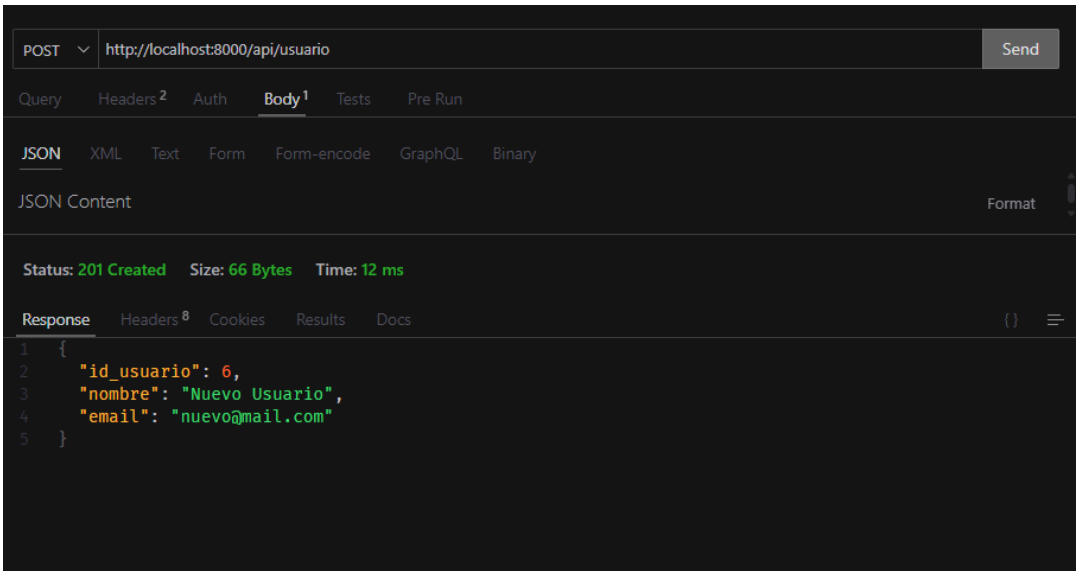
3. Crear un usuario nuevo

POST✅

/api/usuario

Body (JSON):

```
{
  "nombre": "Nuevo usuario a",
  "email": "nuevo@gmail.com",
  "password": "12332423"
}
```



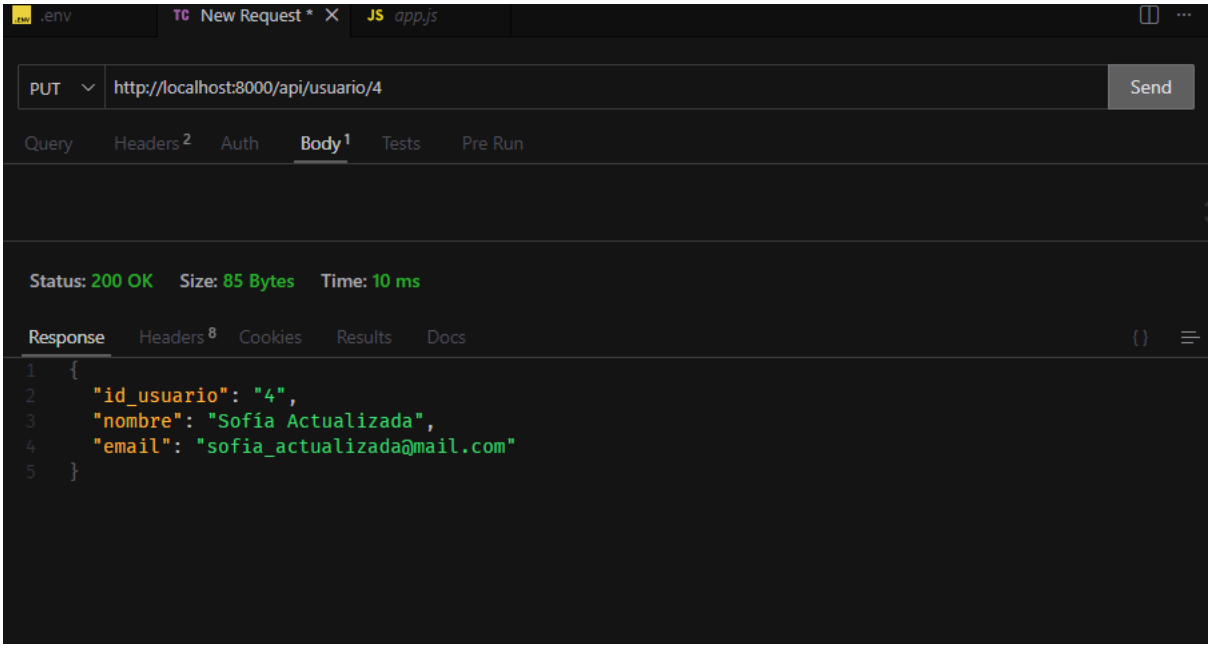
4. Actualizar un usuario

PUT✅

/api/usuario/4

Body (JSON):

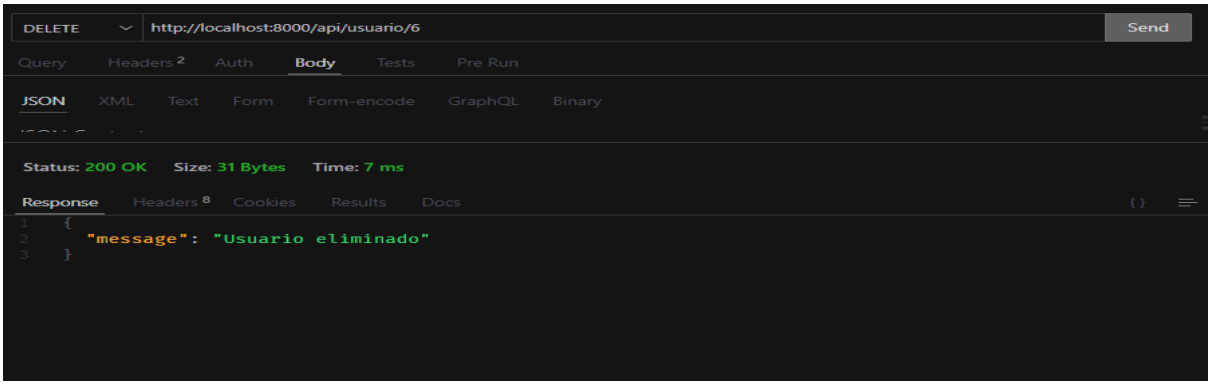
```
{
  "nombre": "Sofía Actualizada",
  "email": "sofia_actualizada@mail.com"
}
```



5. Eliminar un usuario

DELETE✅

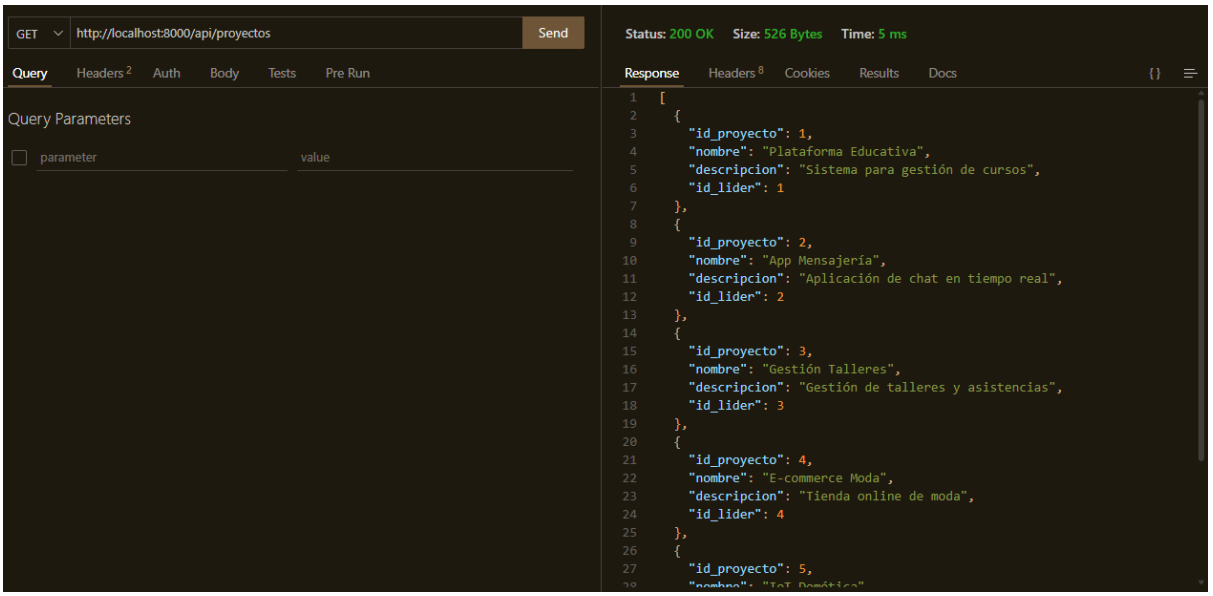
/api/usuario/6



Jefferson / Proyecto

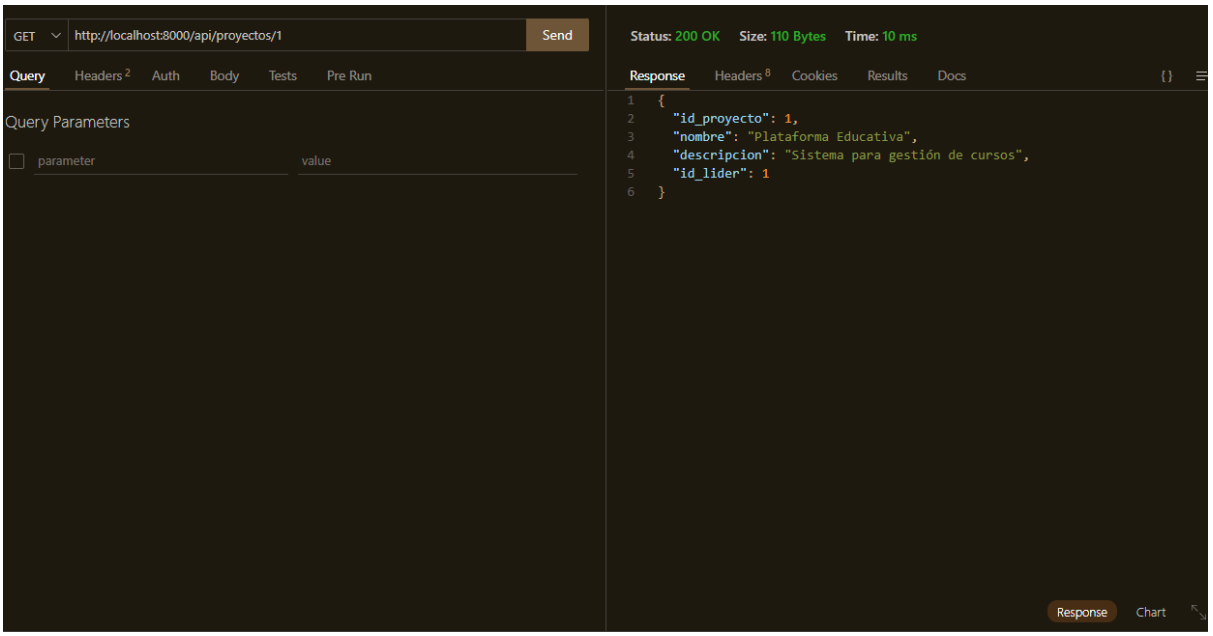
1. Listar todos proyectos: GET:✅

/api/proyectos



2. Obtener por id los proyectos: GET✅

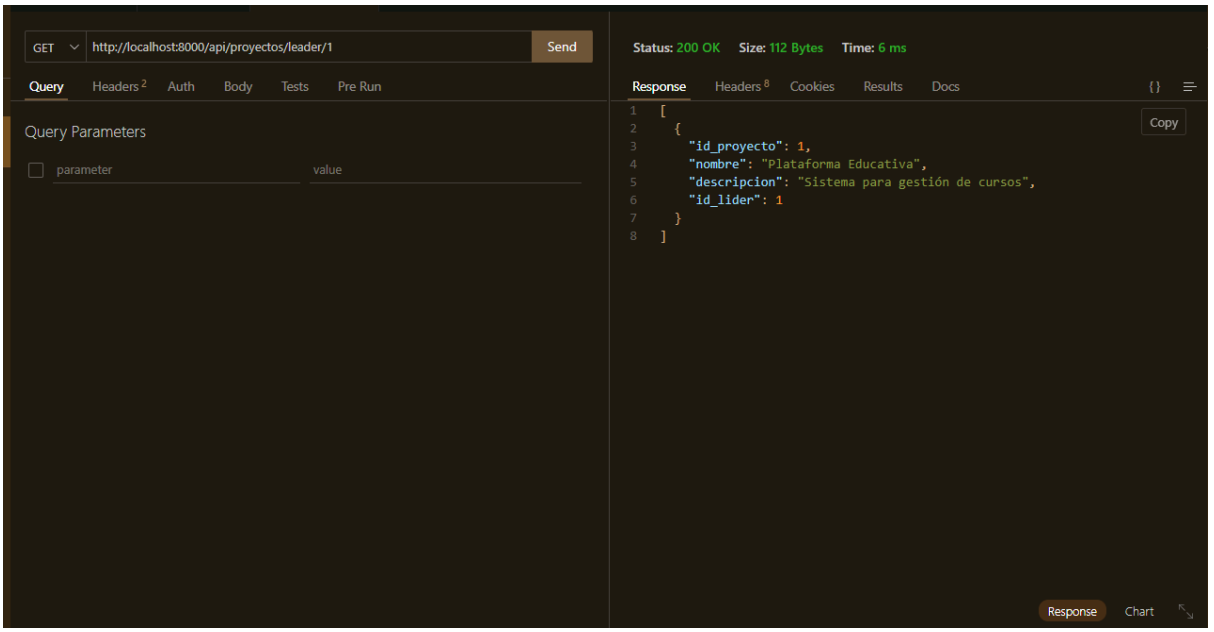
api/proyectos/1



3. **Listar por líder de proyecto**: GET:✅

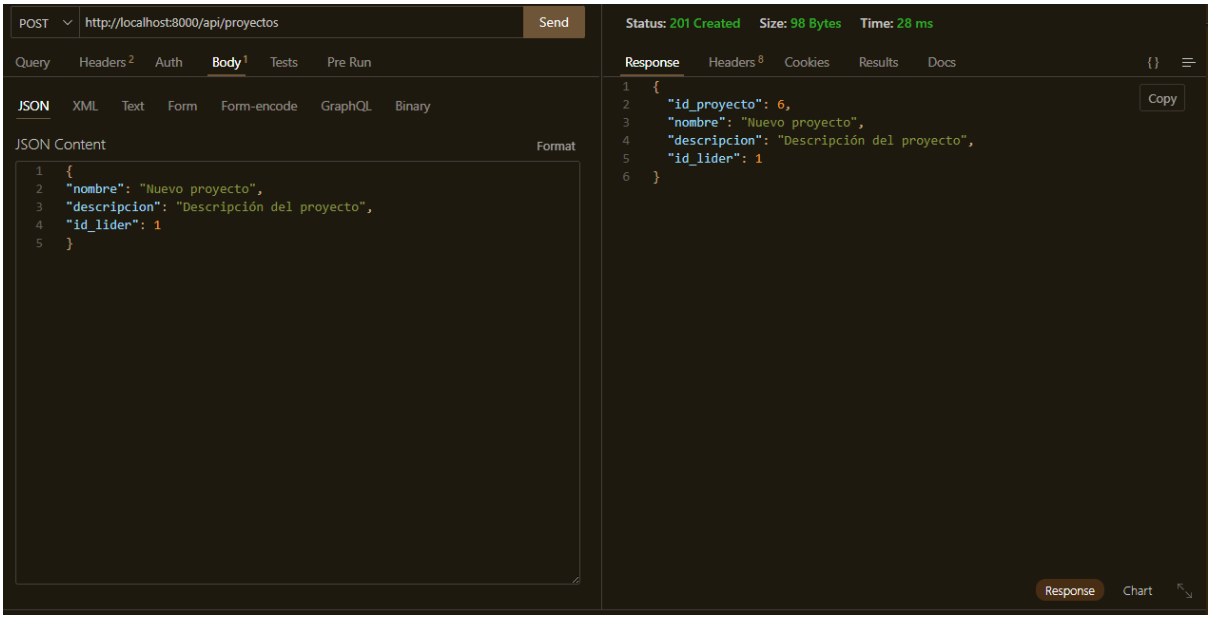
   (ejemplo para id_lider=1)

   /api/proyectos/leader/1



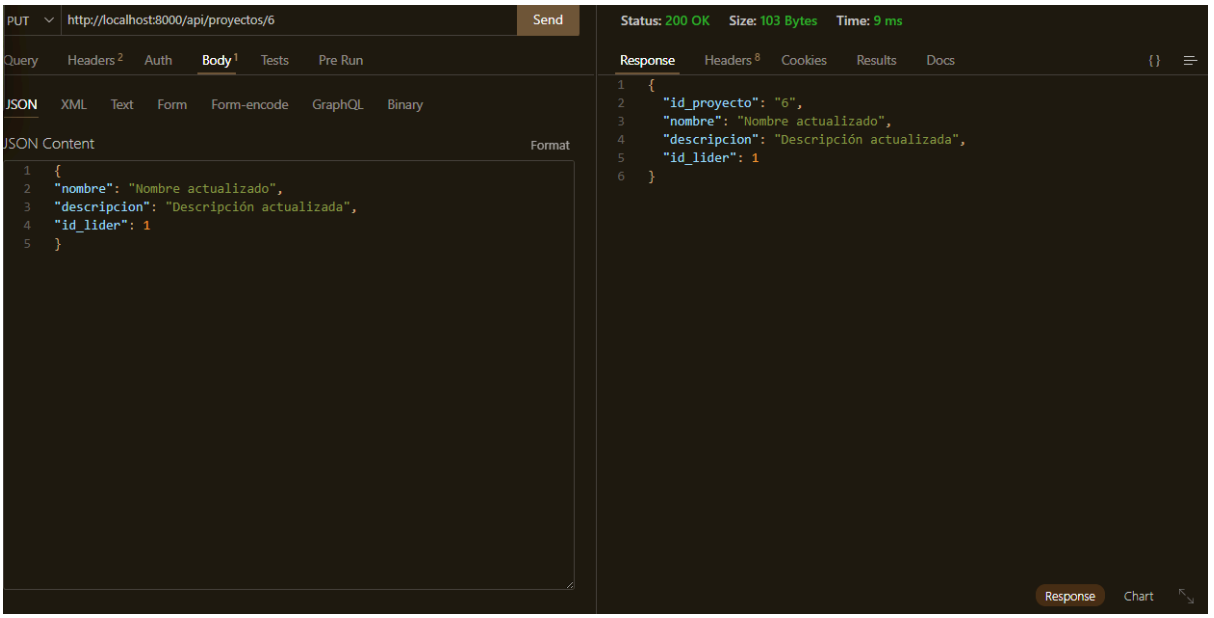
4. **Crear proyecto**: POST✅

   /api/proyectos



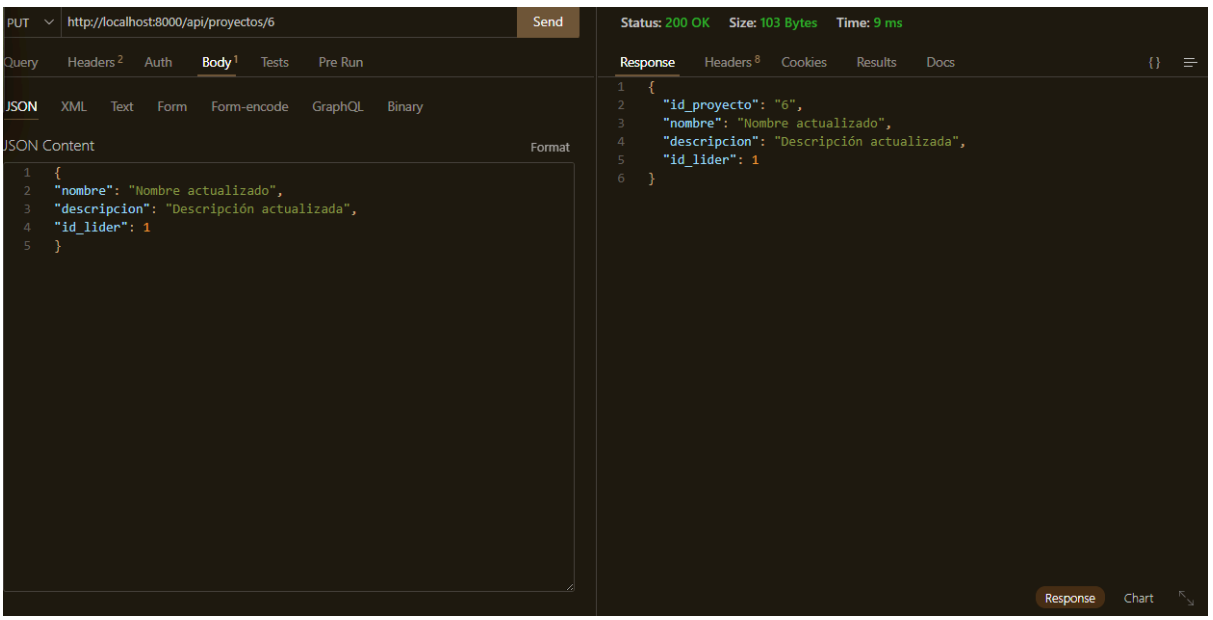
5. **Actualizar Proyecto**: PUT:✅

   /api/proyectos/6



6. **Eliminar proyecto:** DELETE✅

   /api/proyectos/6



Johan / historia_usuario & tarea

1. **Crear nueva historia:**POST✅

   /api/historias

Body (JSON):

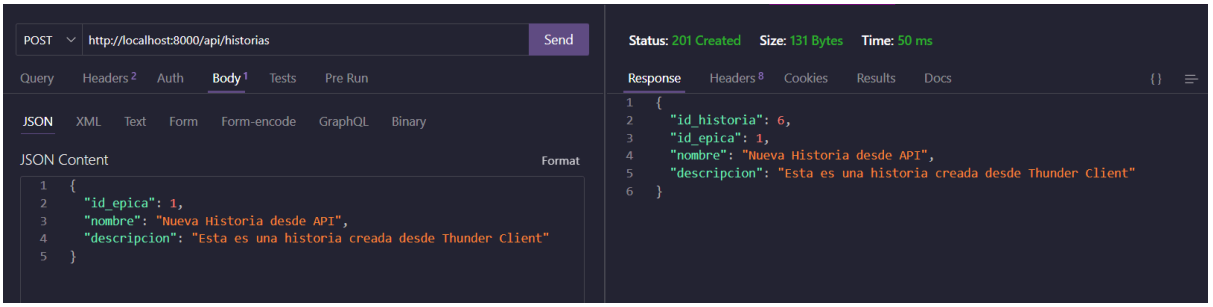
{

"id_epica": 1,

"nombre": "Nueva Historia desde API",

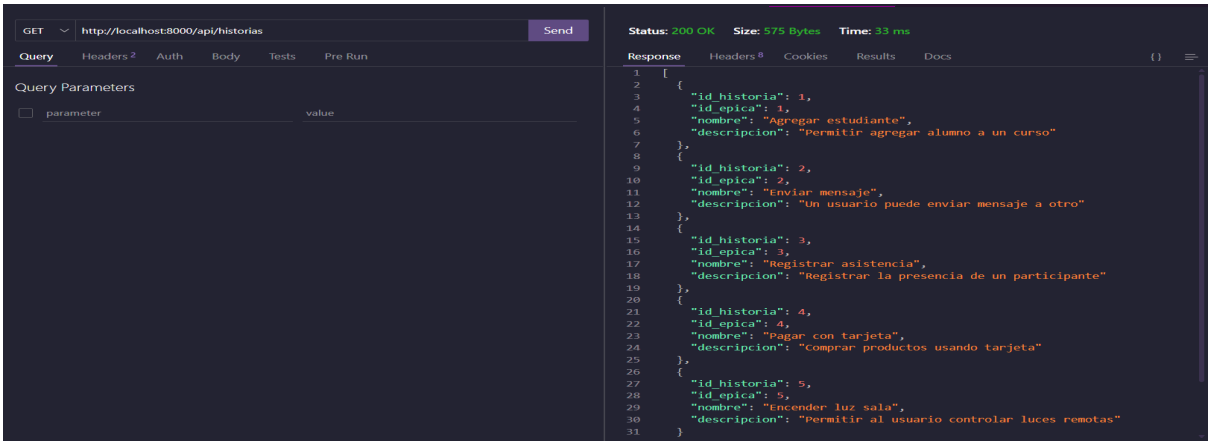
"descripcion": "Esta es una historia creada desde Thunder Client"

}



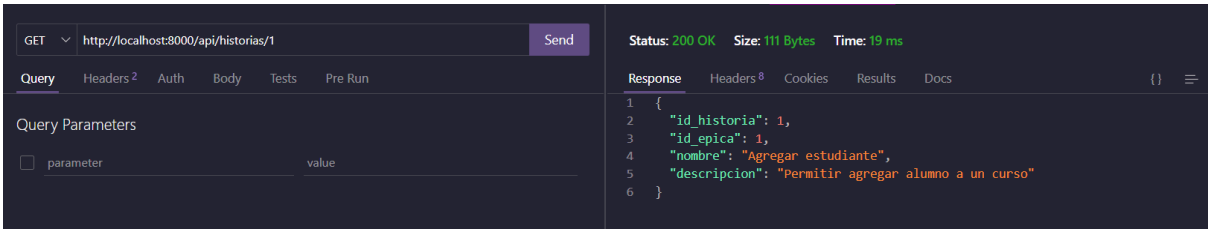
2. **Listar todas las historias:**GET✅

   /api/historias



3. Obtener historia por id:GET✅

/api/historias/{id}



4. Actualizar historia:PUT✅

/api/historias/{id}

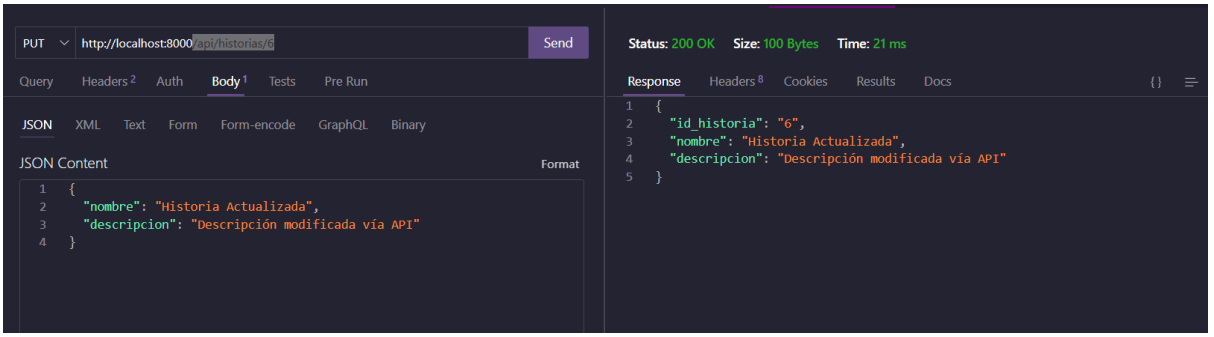
Body (JSON):

{

"nombre": "Historia Actualizada",

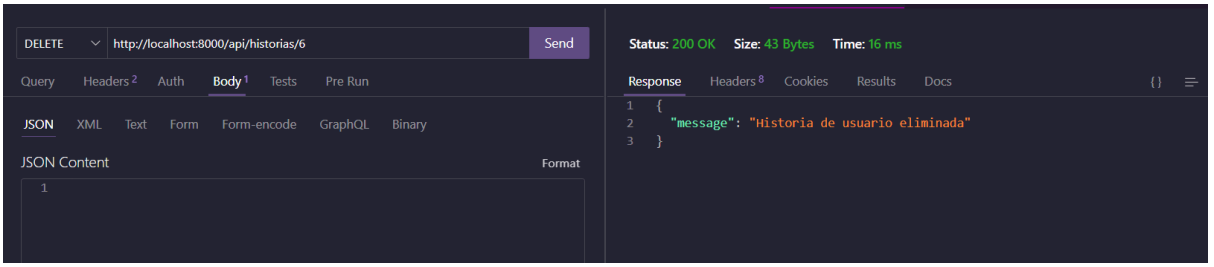
"descripcion": "Descripción modificada vía API"

}



5. Eliminar historia:DELETE✅

/api/historias/{id}



1. Crear nueva tarea:POST✅

/api/tareas

Body (JSON):

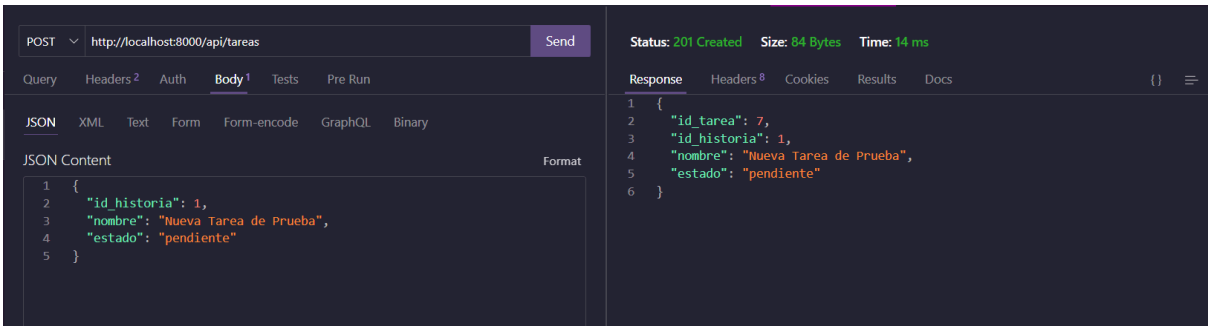
{

"id_historia": 1,

"nombre": "Nueva Tarea de Prueba",

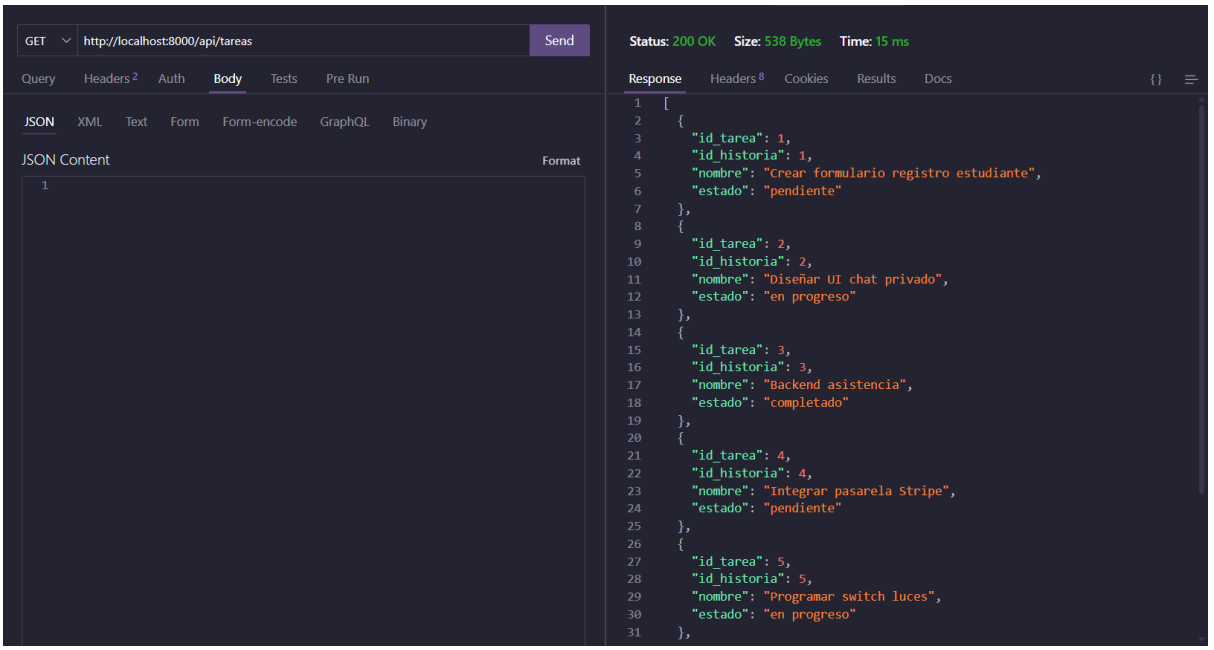
"estado": "pendiente"

}



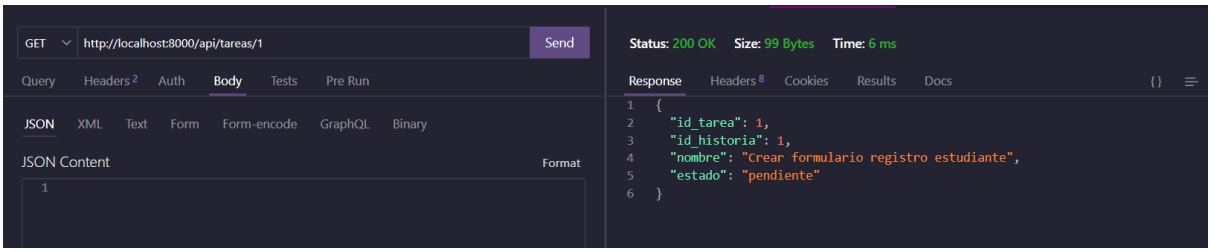
2. Listar todas las tareas:GET✅

/api/tareas



3. Obtener una tarea por ID:GET✅

/api/tareas/{id}



4. Actualizar tarea:PUT✅

/api/tareas/{id}

Body (JSON):

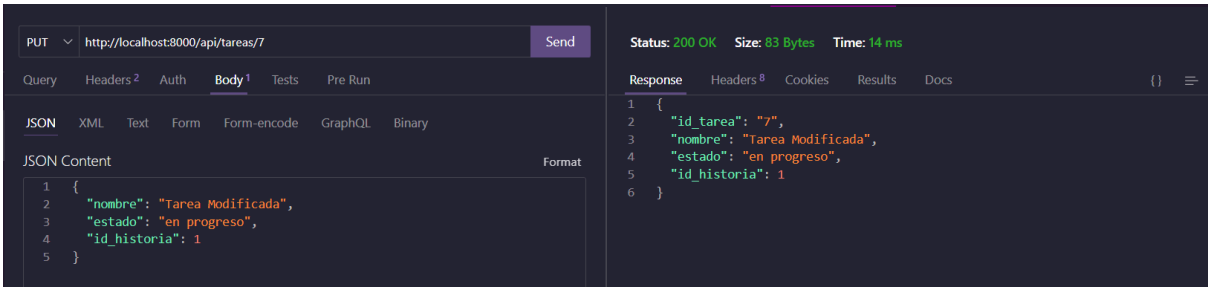
{

"nombre": "Tarea Modificada",

"estado": "en progreso",

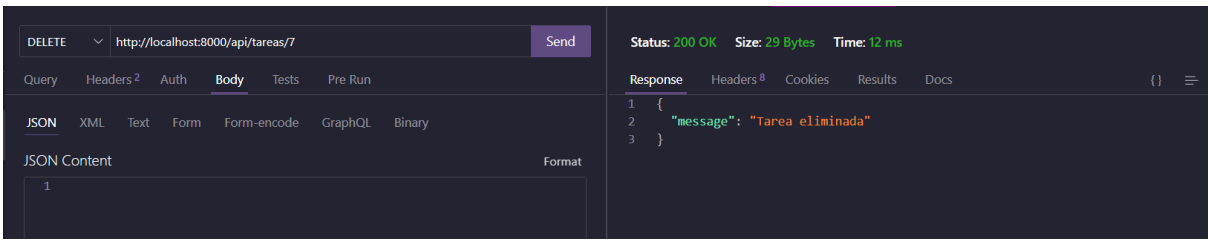
"id_historia": 1

}



5. Eliminar tarea:DELETE✅

/api/tareas/{id}

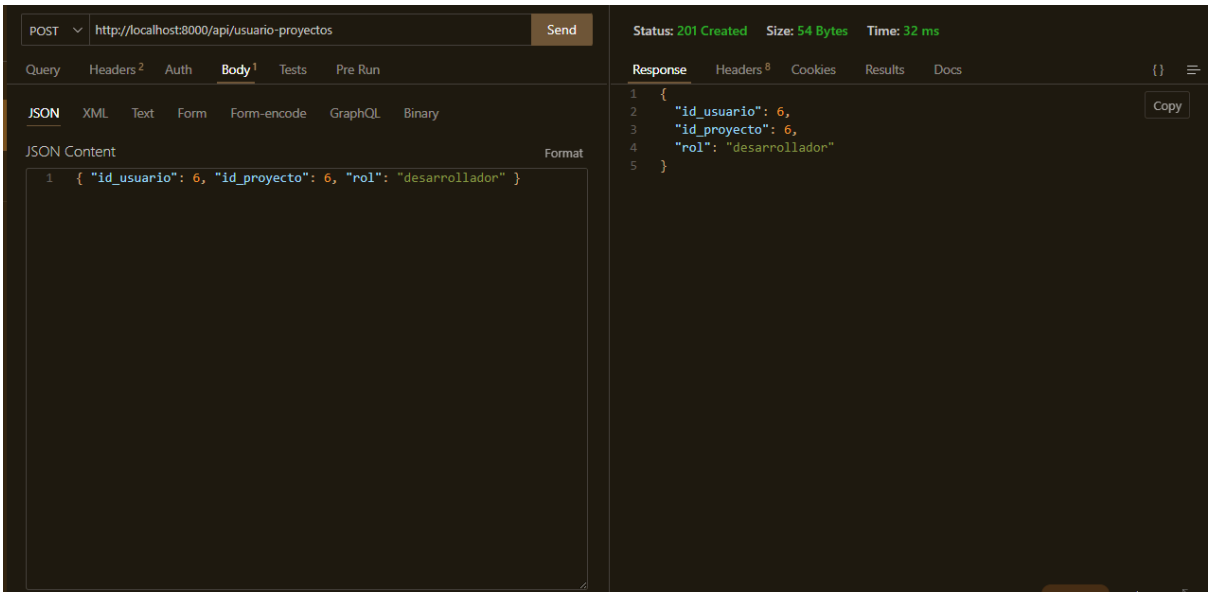


Jhon / usuario_proyecto & usuario_tarea

usuario_proyecto

1. **Asignar usuario a proyecto:** POST✅

   api/usuario-proyectos

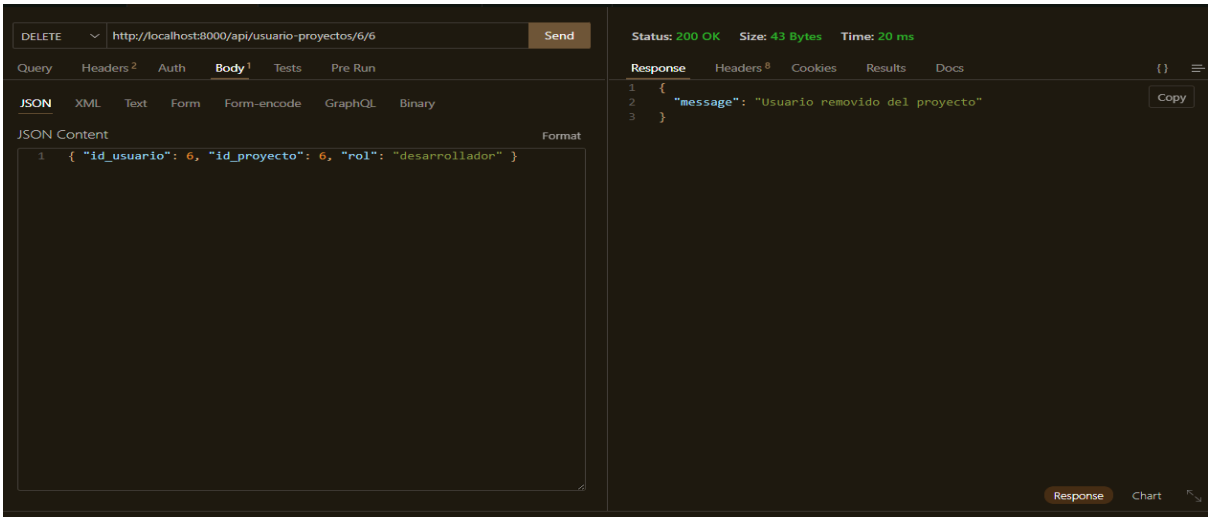


2. Quitar usuario de proyecto:DELETE✅

/api/usuario-proyectos/:id_usuario/:id_proyecto

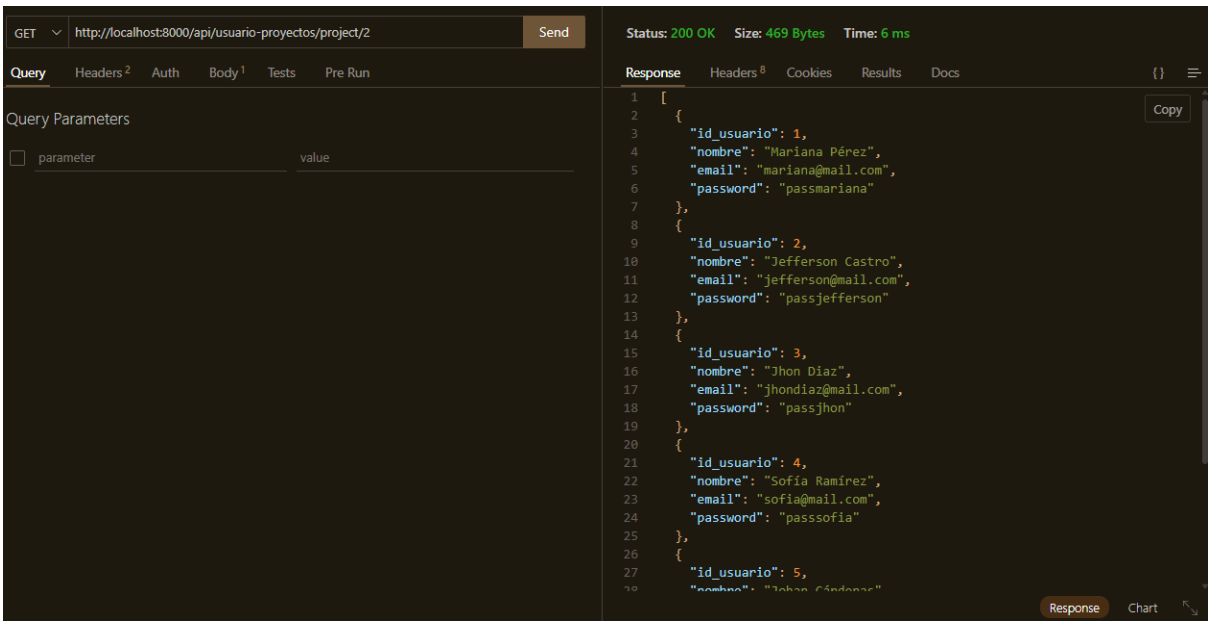
Ejemplo:

/api/usuario-proyectos/6/6



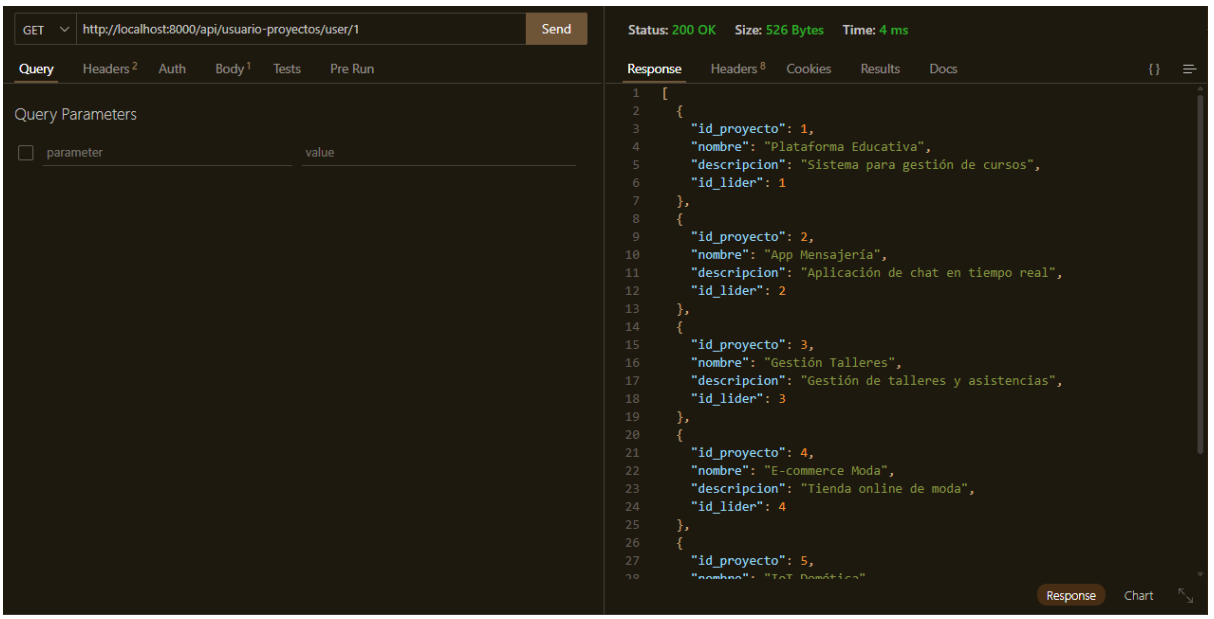
3.Listar usuarios de un proyecto:GET✅

api/usuario-proyectos/project/2



4. Listar proyectos de un usuario — GET✅

/api/usuario-proyectos/user/1

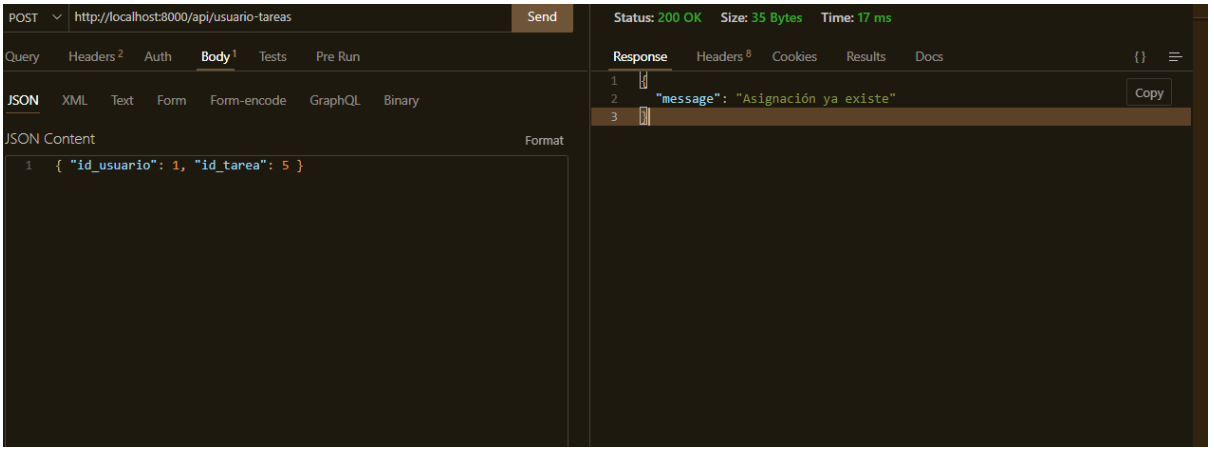


usuario_tarea

1. **Asignar usuario a tarea** :POST✅

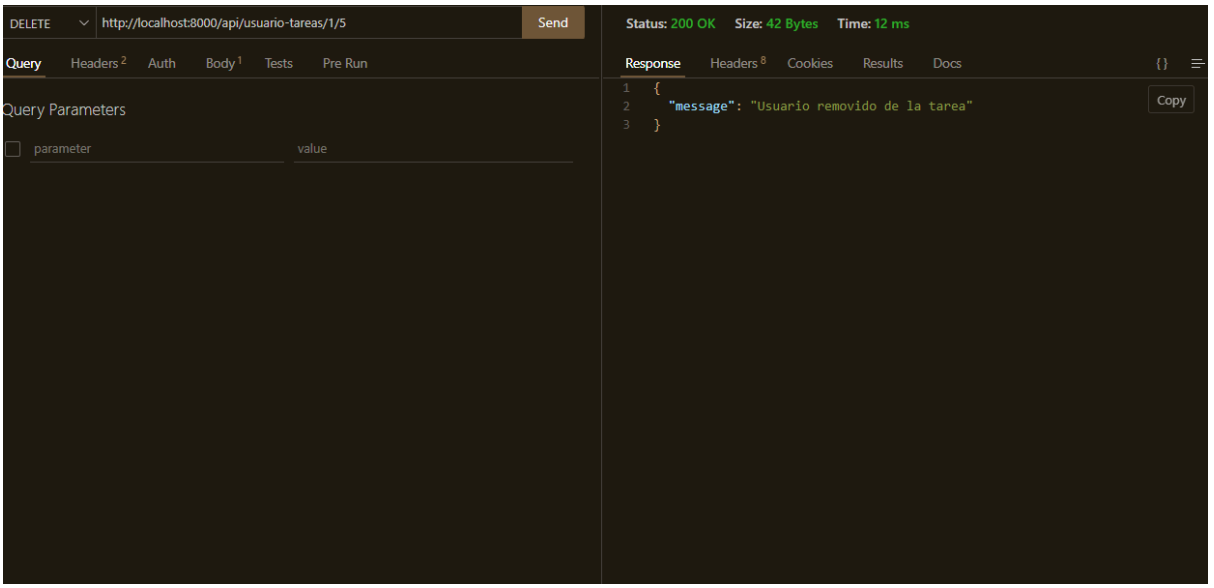
   api/usuario-tareas

- Si la asignación ya existe → responde 200 { message: "Asignación ya existe" }
- Si falta usuario o tarea → responde 400 con mensaje explicando que faltan registros padre
- Si se crea correctamente → 201 con `{ id_usuario, id_tarea }`



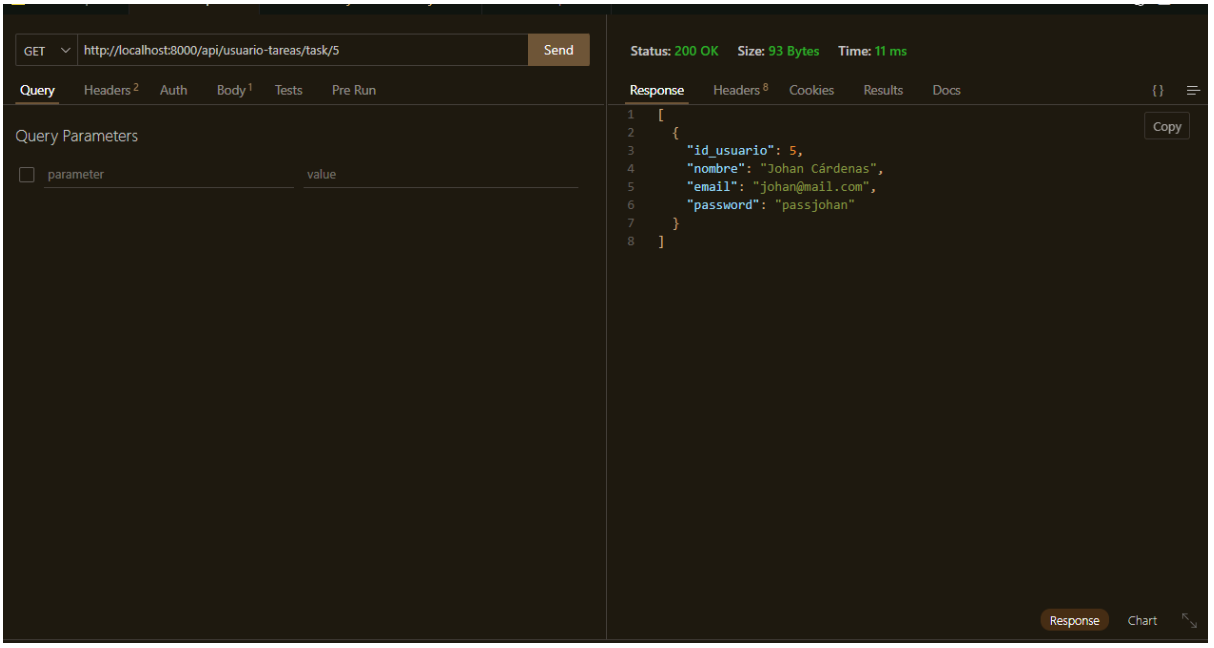
2. **Quitar usuario de tarea** :DELETE✅

   /api/usuario-tareas/1/5



3. Listar usuarios de una tarea :GET✅

/api/usuario-tareas/task/5



4. Listar tareas de un usuario: GET✅

/api/usuario-tareas/user/1