Examen U2: Guaman Jhon

Ejercicio 2: API RESTful para Gestión de Tareas (To-Do)

Este ejercicio se enfoca en la correcta implementación de una API RESTful y la aplicación rigurosa de una arquitectura por capas.

Descripción del Ejercicio

Debes desarrollar una API RESTful completa para gestionar una lista de tareas (To-Do). La API debe permitir a los usuarios registrarse, iniciar sesión y luego realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) sobre sus propias tareas. Un usuario **no** debe poder ver o modificar las tareas de otros usuarios.

Requisitos Técnicos

- **Backend:** Node.js con Express.
- **Base de Datos:** MongoDB con Mongoose.
- **Autenticación:** JWT (JSON Web Tokens).
- **Arquitectura:** Aplicar principios de Arquitectura Limpia (separación de capas). El foco principal es este punto.

Requisitos Funcionales

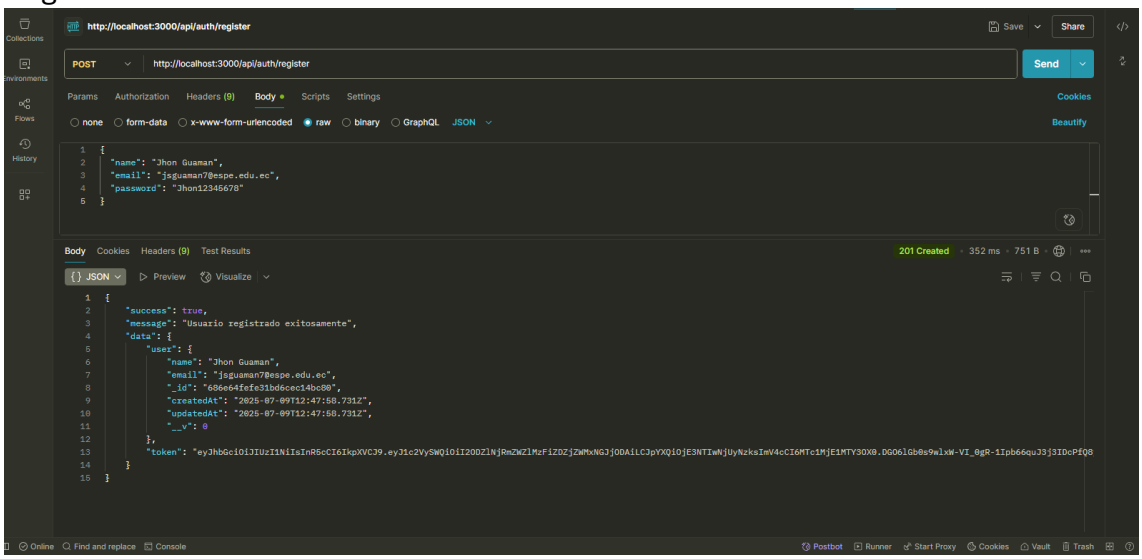
1. **Autenticación de Usuarios:**
    - Crear un endpoint POST /api/auth/register para registrar un nuevo usuario (nombre, email y password). La contraseña debe guardarse hasheada.
    - Crear un endpoint POST /api/auth/login que valide las credenciales y devuelva un JWT.
2. **Operaciones CRUD de Tareas (Endpoints Protegidos):**
    - POST /api/tasks: Crear una nueva tarea. El body debe contener el title y opcionalmente una description. La tarea debe quedar asociada al usuario autenticado.
    - GET /api/tasks: Obtener **únicamente** las tareas del usuario autenticado.
    - PUT /api/tasks/:id: Actualizar una tarea existente (ej. cambiar título, descripción o marcar como completada). Solo el dueño de la tarea puede actualizarla.
    - DELETE /api/tasks/:id: Eliminar una tarea. Solo el dueño de la tarea puede eliminarla.
3. **Modelos de Datos (Mongoose):**
    - **User:** name, email, password.
    - **Task:** title, description, completed (boolean, default: false), user (referencia al modelo User).

Estructura de Proyecto Sugerida (Arquitectura Limpia)

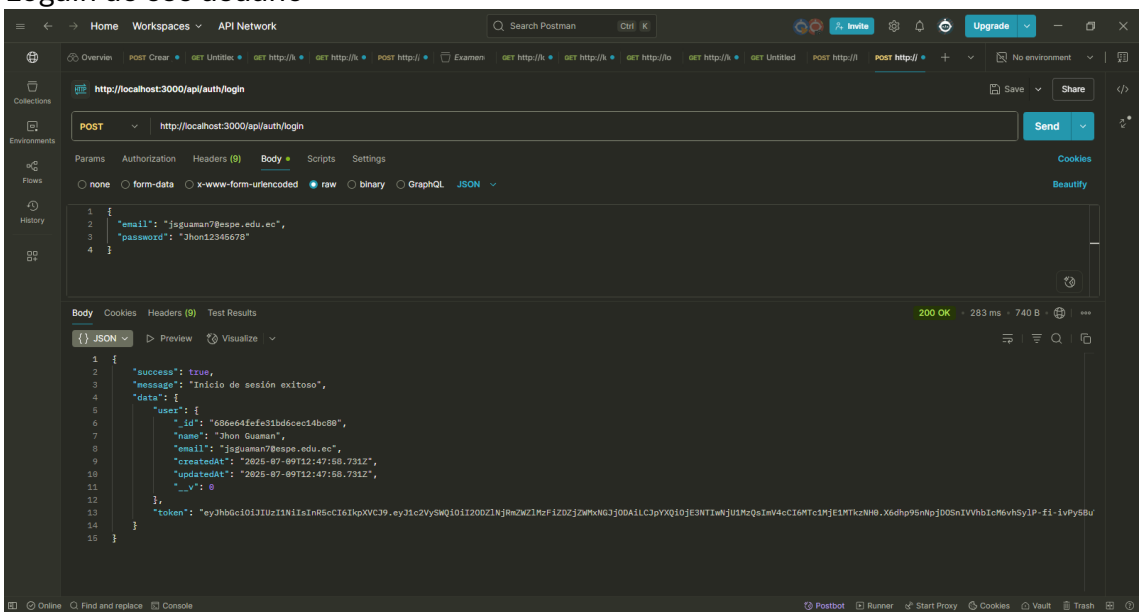
```
/src
|-- /api
|   |-- /routes          # Define las rutas de la API (auth.routes.js, task.routes.js)
|   |-- /controllers     # Controladores (req, res) (auth.controller.js, task.controller.js)
|-- /domain
|   |-- /models          # Esquemas de Mongoose (user.model.js, task.model.js)
|   |-- /use-cases       # Lógica de negocio (create-task.use-case.js, get-user-tasks.use-case.js)
|-- /infrastructure
|   |-- /repositories    # Lógica de acceso a datos (user.repository.js, task.repository.js)
|   |-- /middlewares     # Middlewares de Express (auth.middleware.js)
```

```
|-- /config          # Configuración (db, variables de entorno)
|-- app.js           # Archivo principal del servidor Express
```

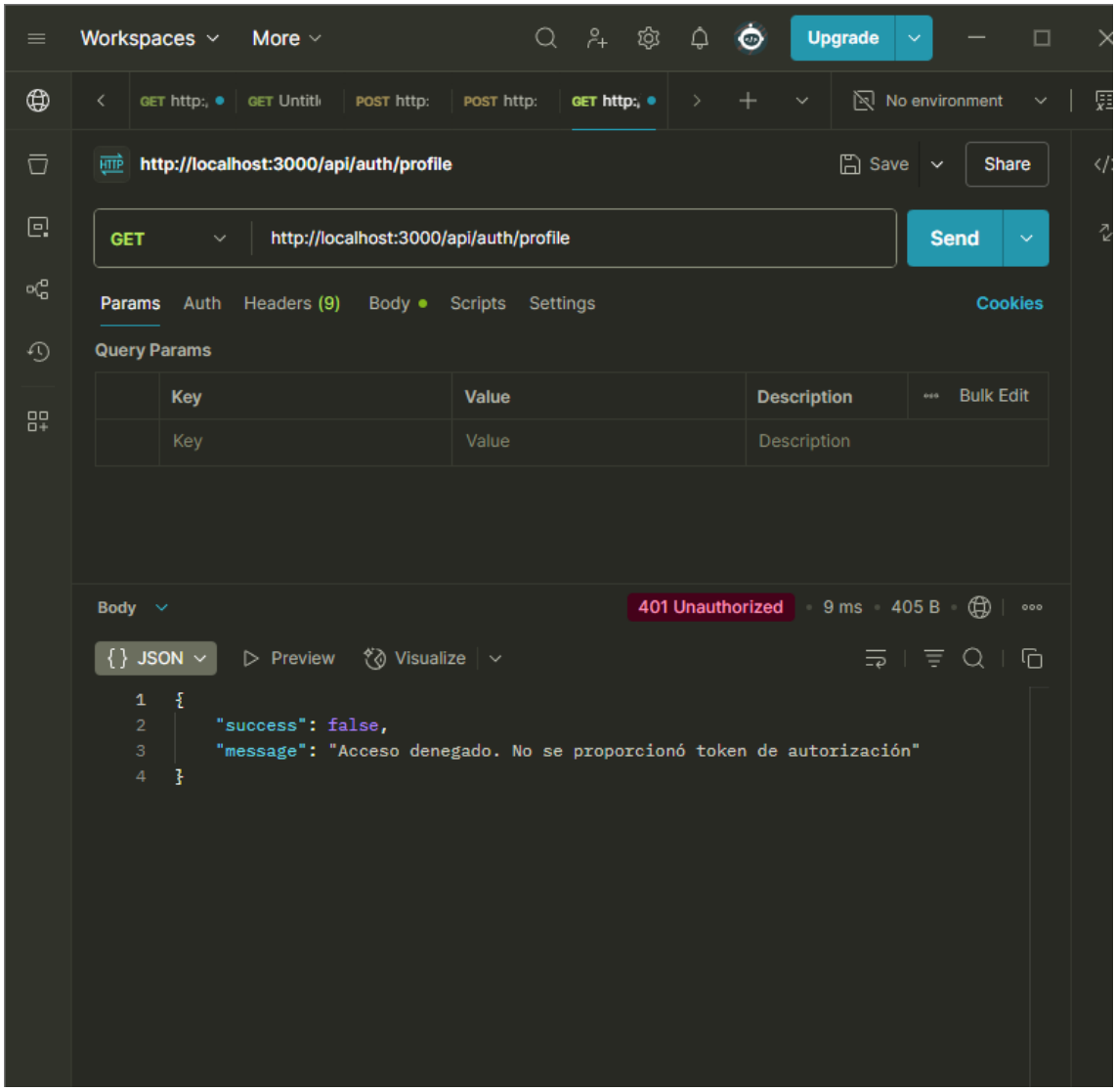
Registro de un usuario



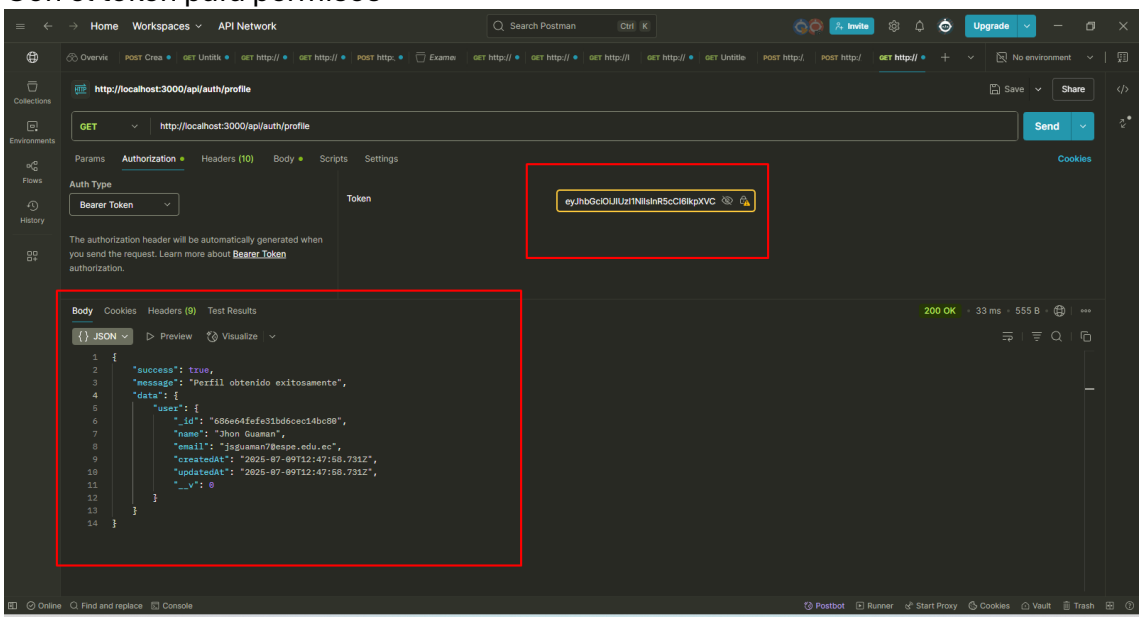
Loguin de ese usuario



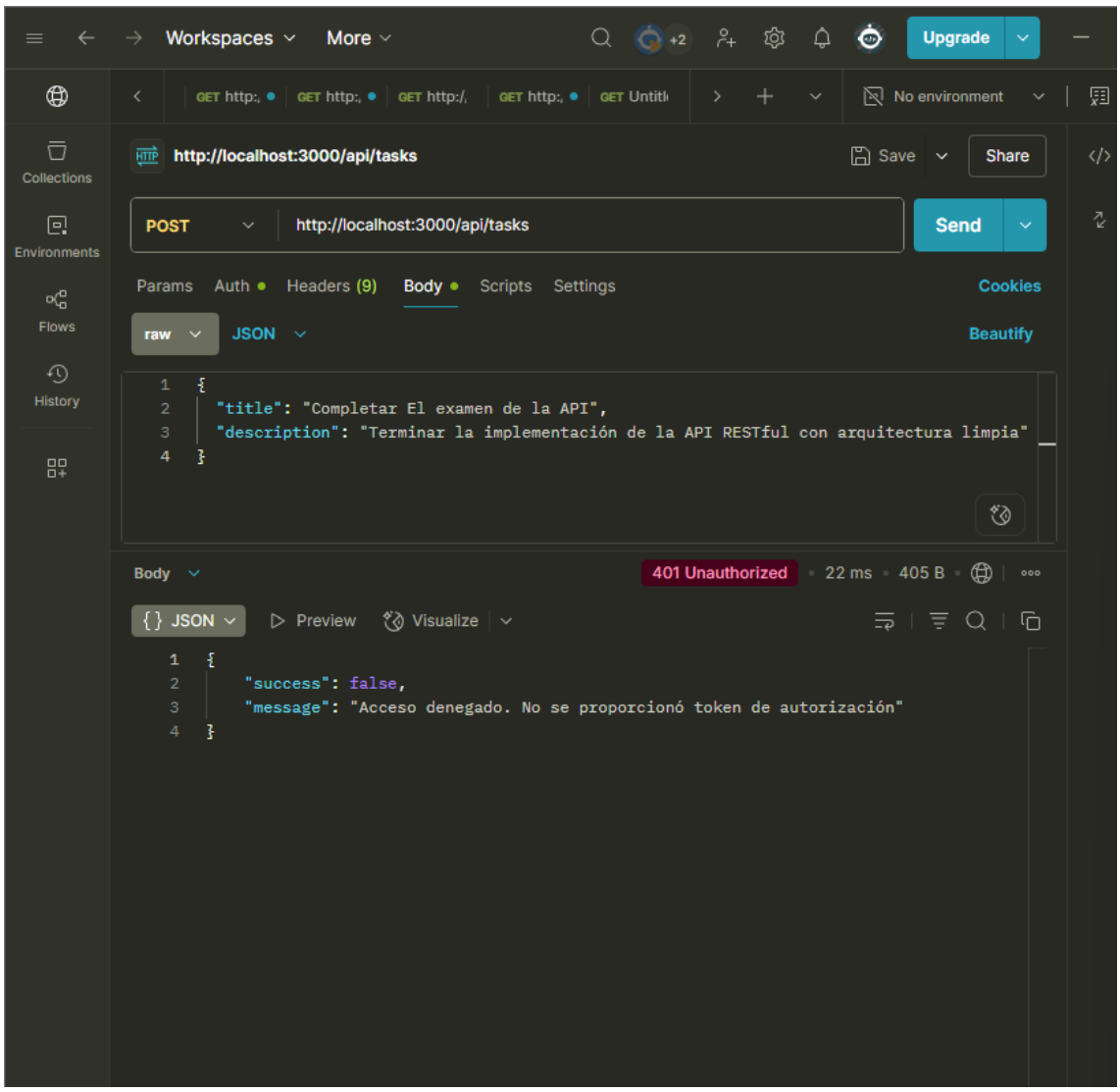
Visualizacion sin token



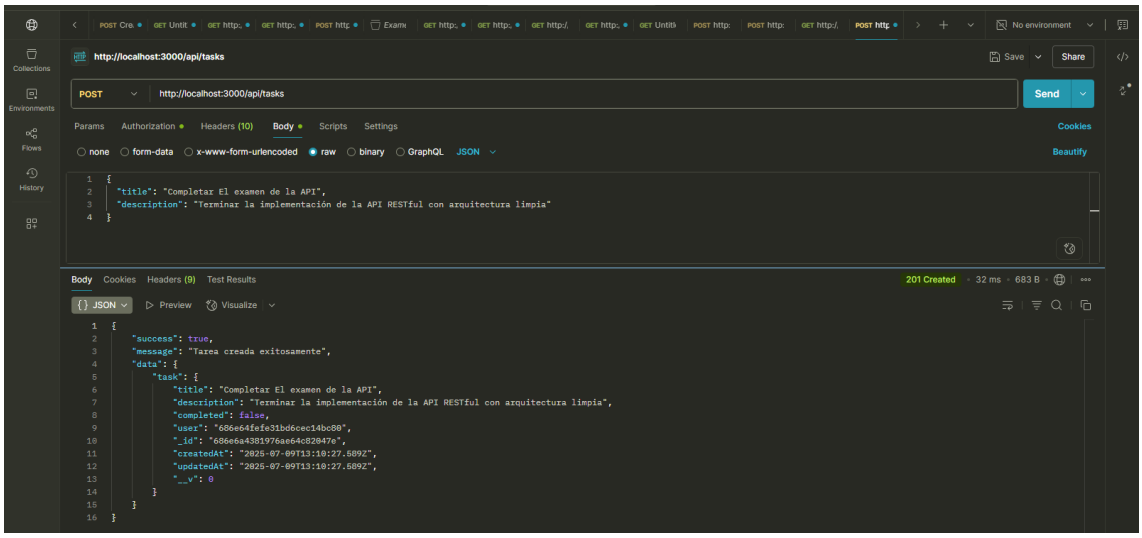
Con el token para permisos



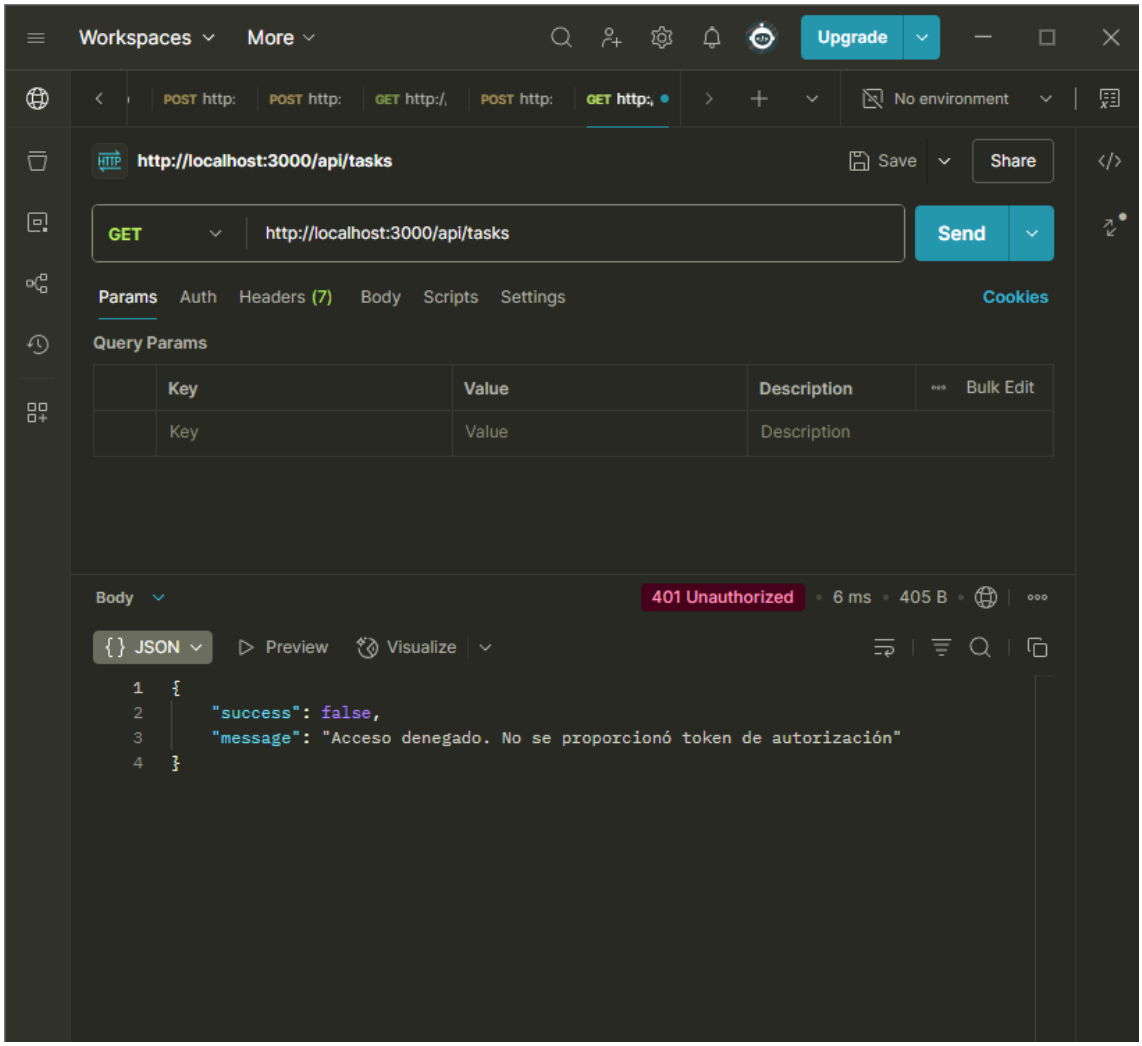
Crear una tarea
Sin el token



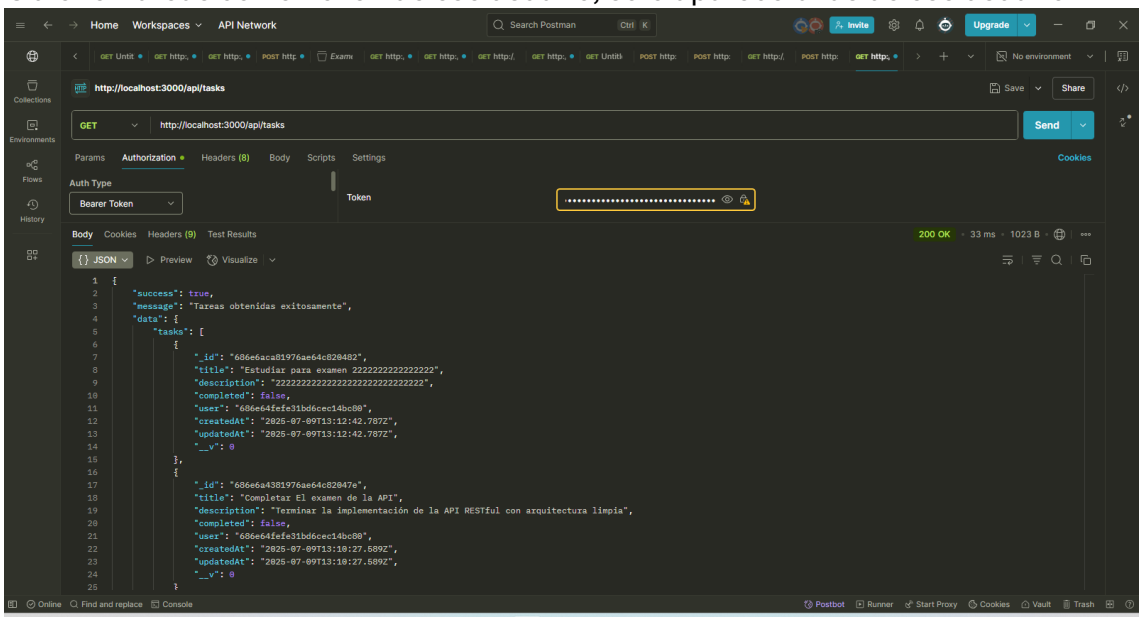
Con el token



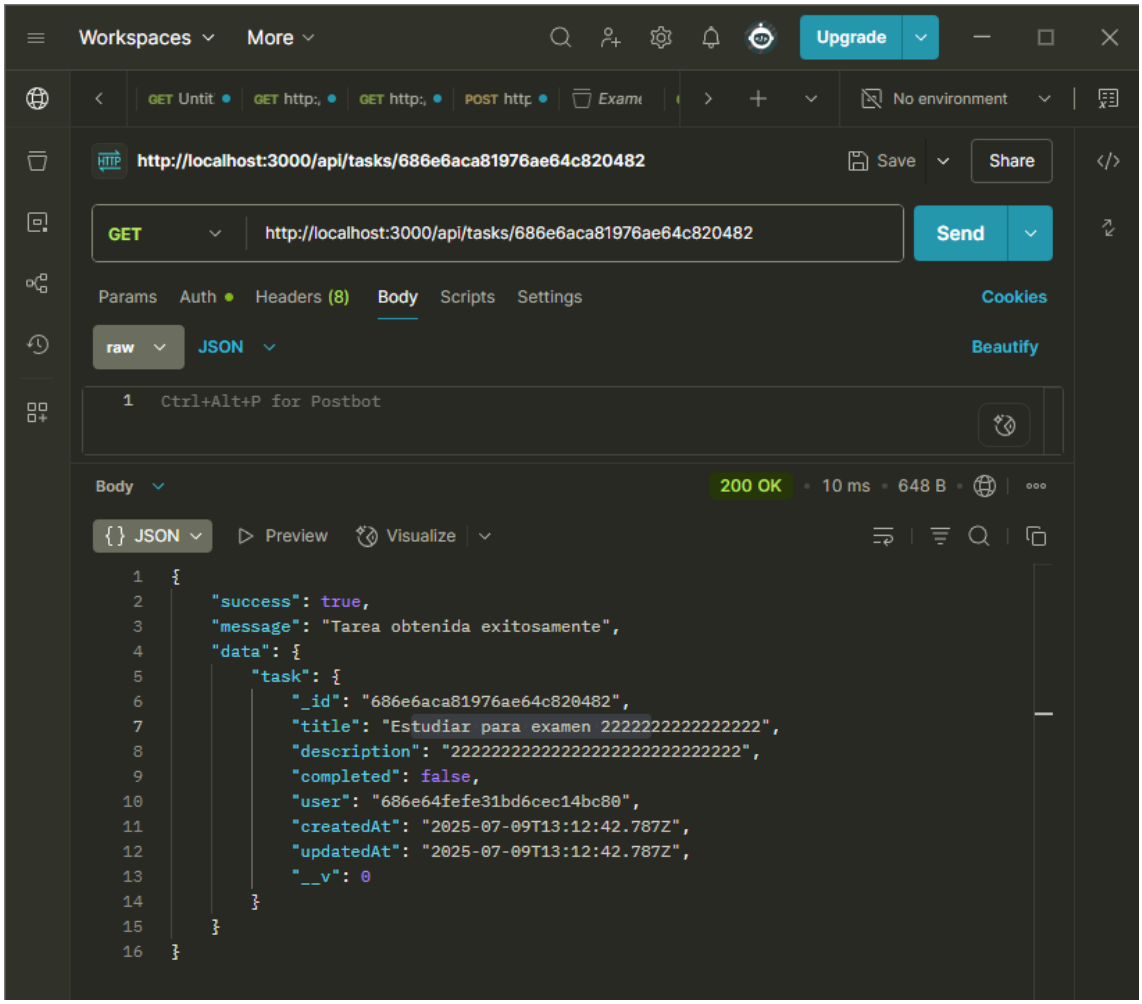
Obtener tareas sin el token



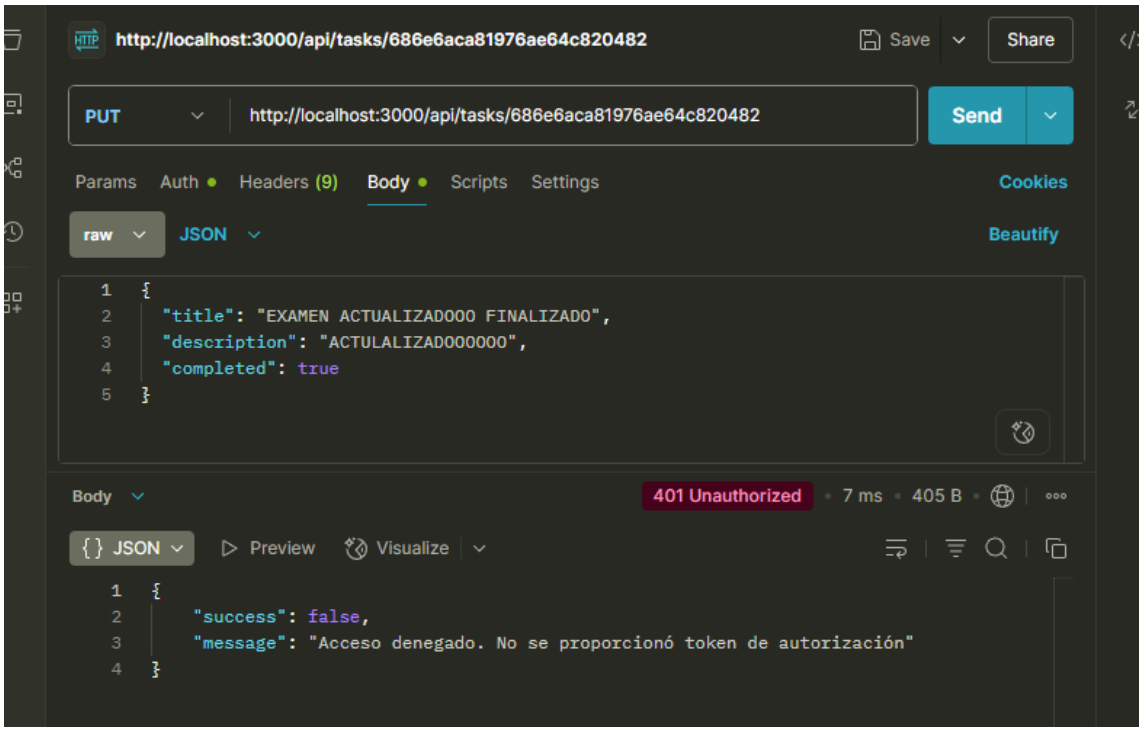
Obtener tareas con el token de ese usuario, solo aparecerá las de ese usuario



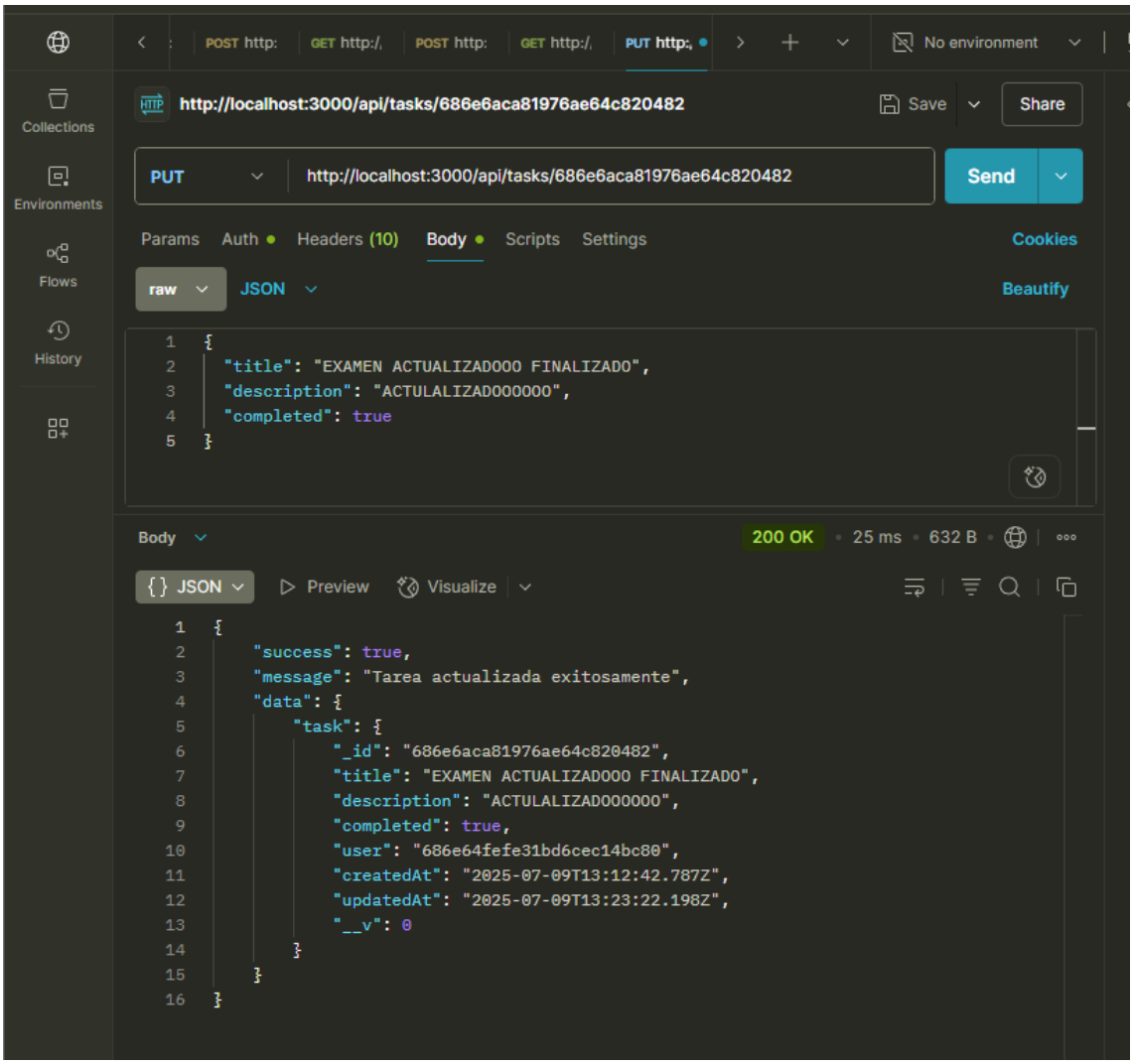
Obtener Tarea Específica



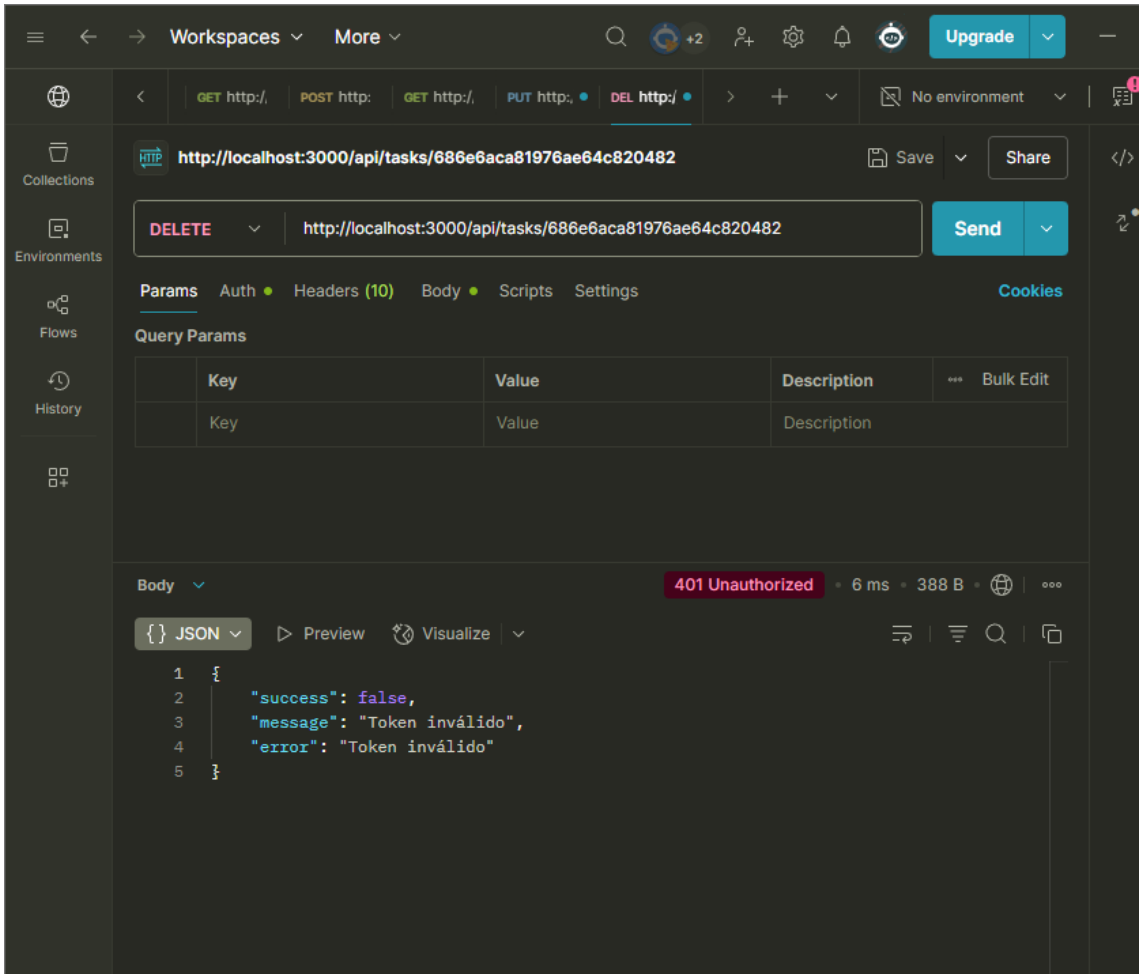
Actualizar Tarea sin token



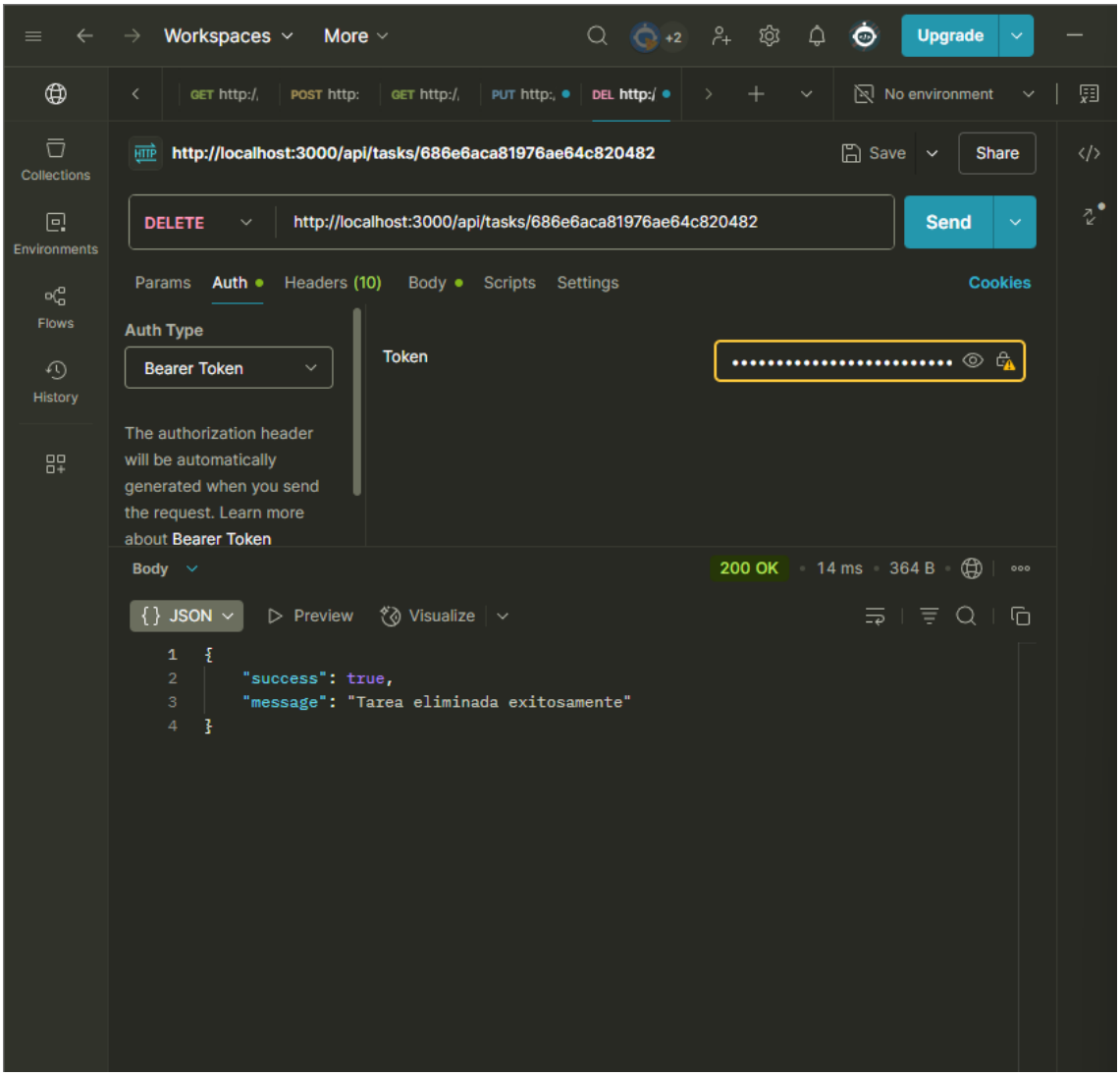
Actualizar Tarea con token (solo el usuario puede actualizar)

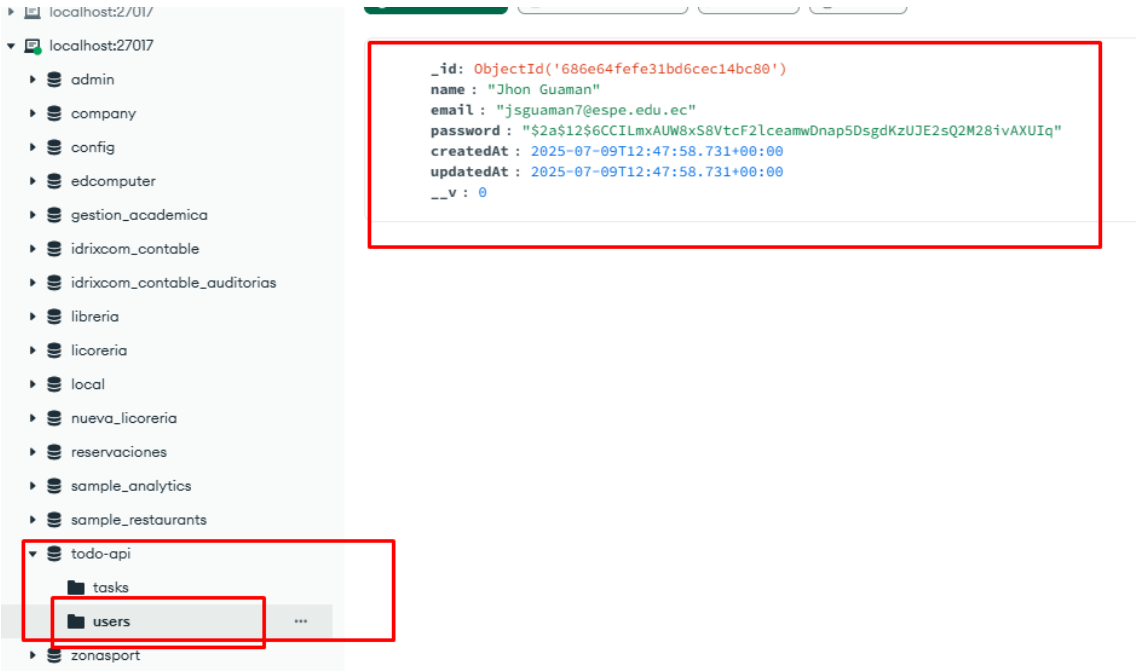


Intento de eliminar tarea sin tener el token



Eliminar tarea con el token(solo el usuario puede eliminarla)


Workspaces
More
Upgrade
No environment
Collections
Environments
Flows
History
http://localhost:3000/api/tasks/686e6aca81976ae64c820482
Save
Share
DELETE
http://localhost:3000/api/tasks/686e6aca81976ae64c820482
Send
Params
Auth
Headers (10)
Body
Scripts
Settings
Cookies
Auth Type
Bearer Token
Token
The authorization header will be automatically generated when you send the request. Learn more about Bearer Token
Body
200 OK
14 ms
364 B
JSON
Preview
Visualize
{
"success": true,
"message": "Tarea eliminada exitosamente"
}


localhost:27017
admin
company
config
edcomputer
gestion_academica
idrixcom_contable
idrixcom_contable_auditorias
libreria
licoreria
local
nueva_licoreria
reservaciones
sample_analytics
sample_restaurants
todo-api
tasks
users
zonasport
_id: ObjectId('686e64fefe31bd6cec14bc80')
name : "Jhon Guaman"
email : "jsguaman7@espe.edu.ec"
password : "$2a$12$6CCILmxAUW8xS8VtcF2lceamwDnap5DsgdKzUJE2sQ2M28ivAXUIq"
createdAt : 2025-07-09T12:47:58.731+00:00
updatedAt : 2025-07-09T12:47:58.731+00:00
__v : 0


localhost:27017
admin
company
config
edcomputer
gestion_academica
idrixcom_contable
idrixcom_contable_auditorias
libreria
licoreria
local
nueva_licoreria
reservaciones
sample_analytics
sample_restaurants
todo-api
tasks
_id: ObjectId('686e6a4381976ae64c82047e')
title : "Completar El examen de la API"
description : "Terminar la implementación de la API RESTful con arquitectura limpia"
completed : false
user : ObjectId('686e64fefe31bd6cec14bc80')
createdAt : 2025-07-09T13:10:27.589+00:00
updatedAt : 2025-07-09T13:10:27.589+00:00
__v : 0