# Resumen

Manual de utilización de api con postman junto a pruebas unitarias. El alcance principal es que debe seguir los pasos de forma secuencial para probar el app. El app cuenta con acceso tanto para roles configurado para administrador y usuarios.

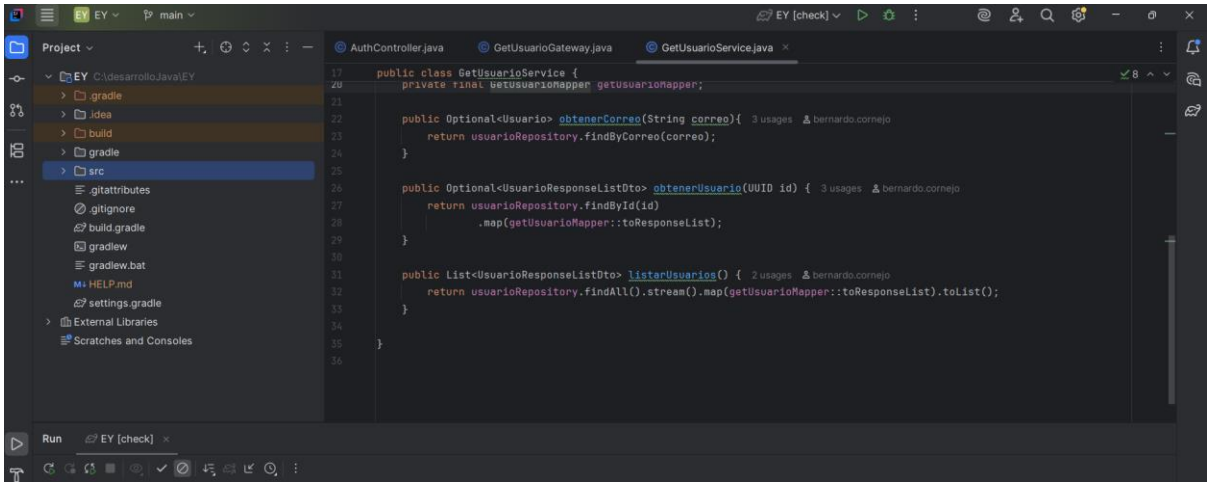
## Programas

Para gestionar la prueba de la API en java debe tener Postman y Intelij IDE

| Postman | Intelij IDE |
| --- | --- |
| POSTMAN | IJ |

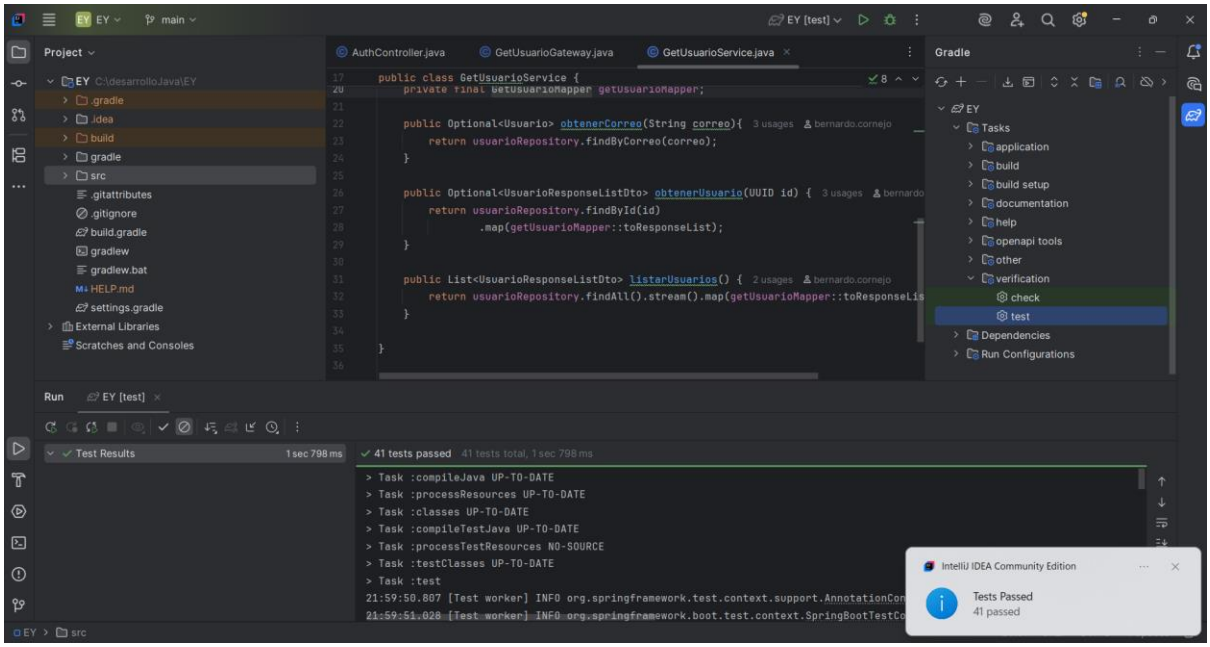
### Intelij IDE

Debe abrir Intelij IDE y seleccionar el proyecto EY

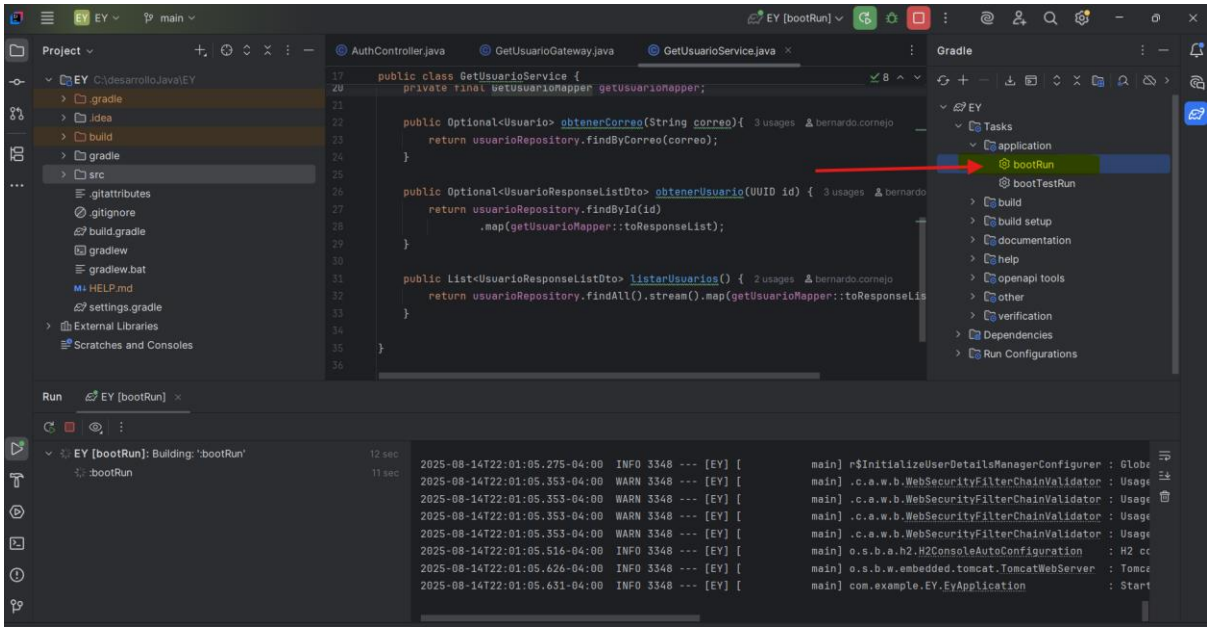



Debe presionar el botón Open para abrir el proyecto y esperar que cargar las dependencias de gradle. Terminada de carga de las dependencias y constructor del proyecto, ir a Main de java y ejecutar el app.

Se revisan los test




Se ejecuta dos veces para iniciar el app. El proyectos se ejecutara en el puerto 8080



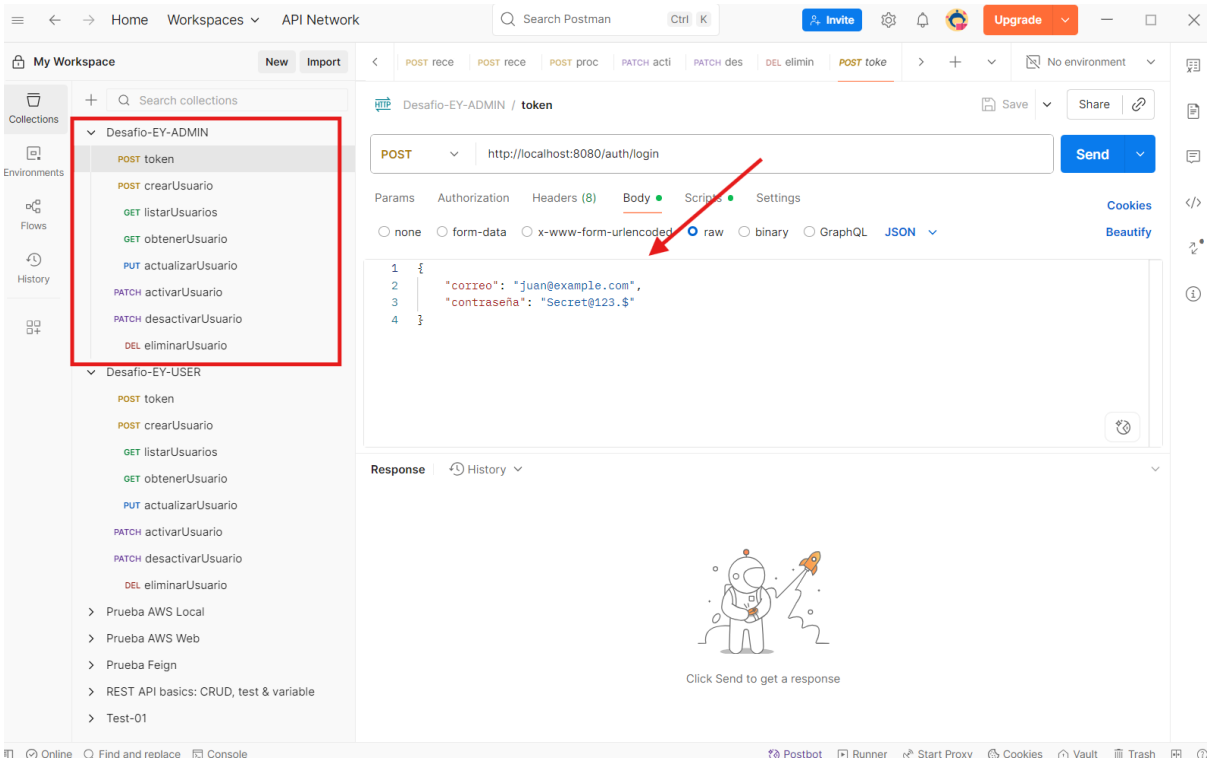
## Postman

Abrir el app postman y presionar Import :

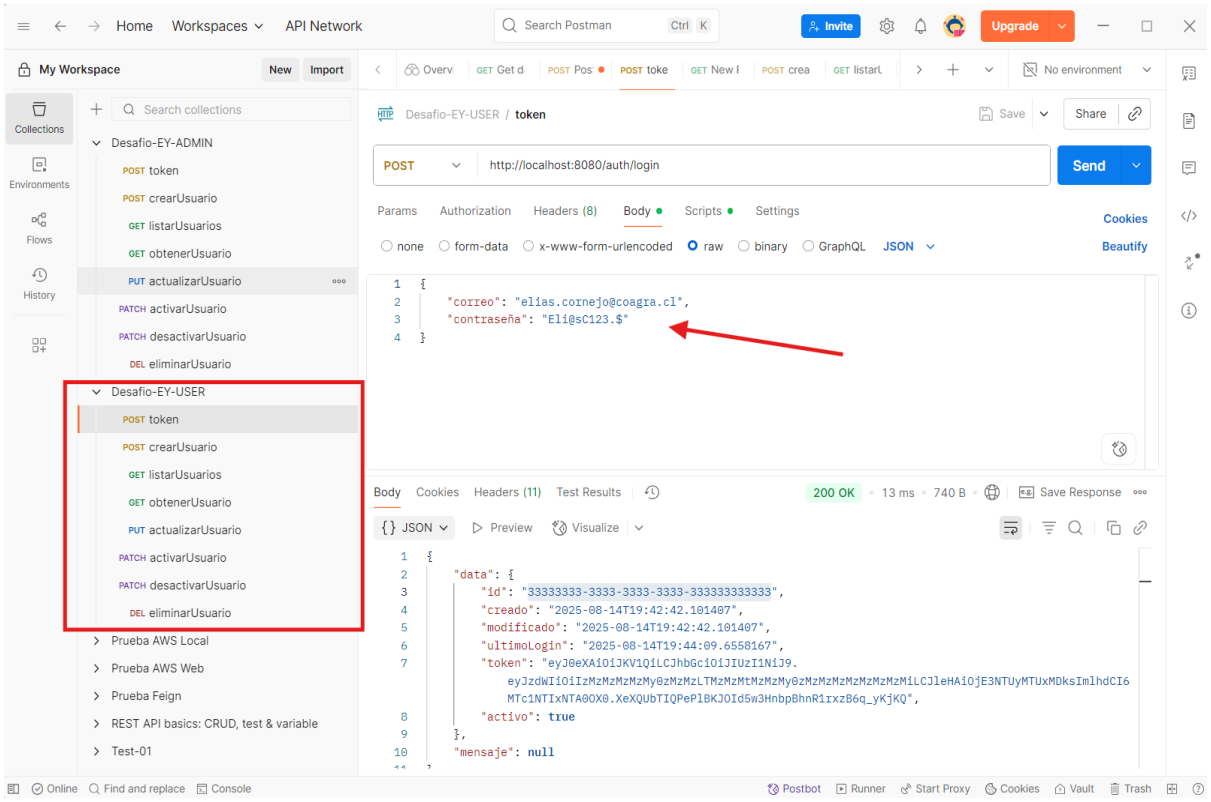
- Desafio-EY-ADMIN.postman_collection.json
- Desafio-EY-USER.postman_collection.json

Para una prueba completa se sugiere abrir los dos

### *ADMIN*




### *USER*

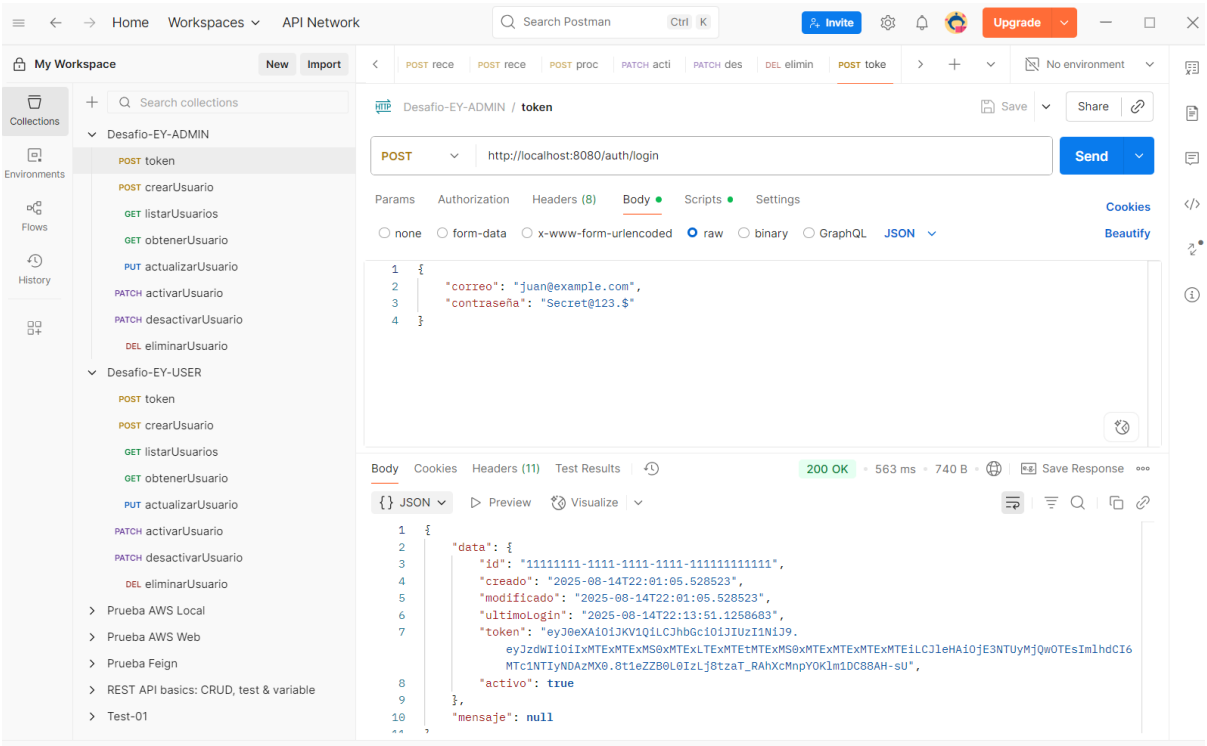


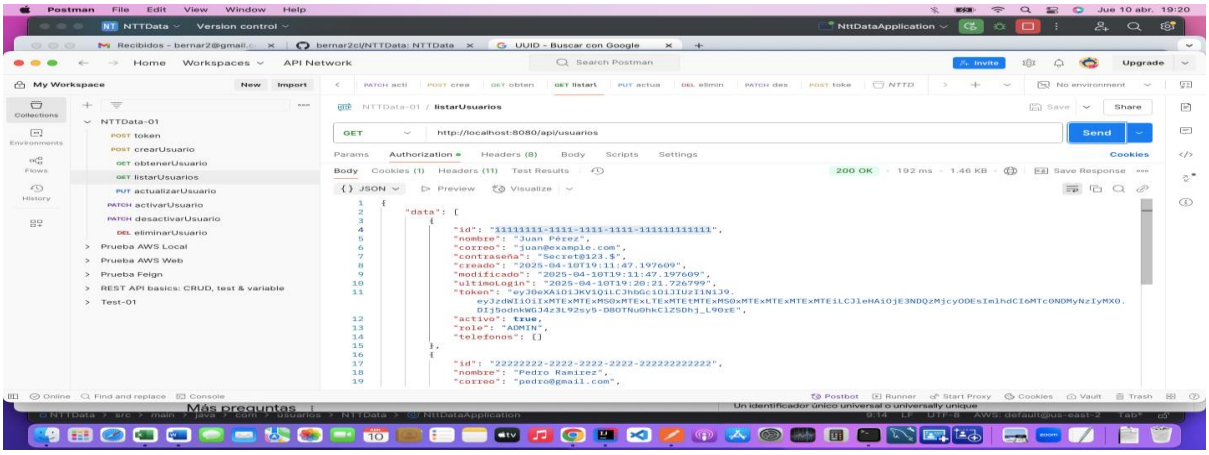

Se cargaran todo los endpoint

| Metodo | Nombre | Descripción |
|---|---|---|
| POST | Token | Genera token con autentificación JWT |

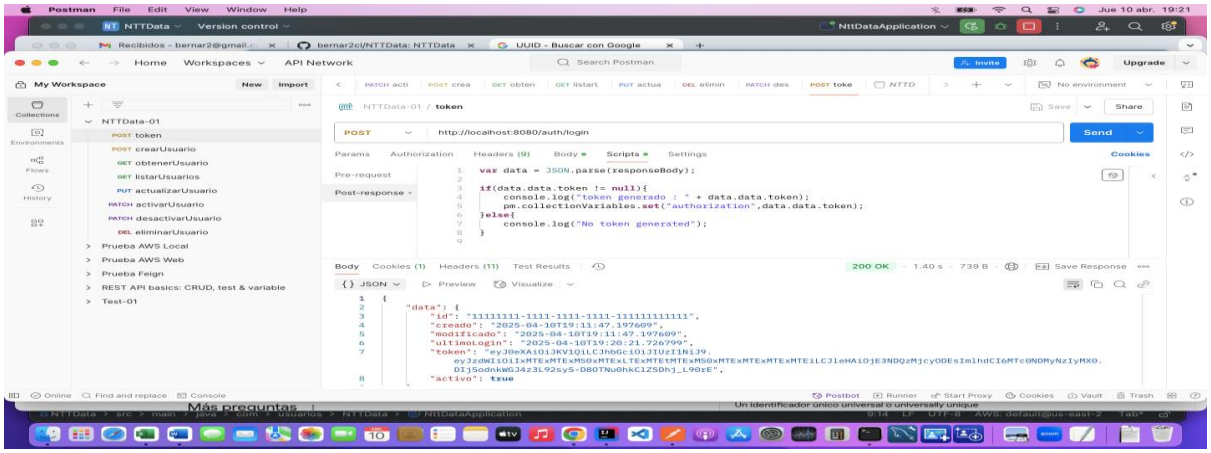
| | | |
|---|---|---|
| POST | crearUsuario | Crea un nuevo usuario |
| GET | obtenerUsuario | Recupera un usuario por UUID |
| GET | listarUsuarios | Lista todos los usuarios |
| PUT | actualizarUsuario | Actualiza un usuario por UUID |
| PATCH | activarUsuario | Activa usuario por UUID |
| PATCH | desactivarUsuario | Desactiva usuario por UUID |
| DEL | eliminarUsuario | Elimina usuario por UUID |

En el post Token, la clave se establece de forma automática en la variable. Esto permite generar una prueba de no impacto.




Para validar si la autentificación es válida se debería sacar la variable {{authorization}} o sacar el establecimiento de la variable.




## Configuración

Usuarios: Se tiene registros 3 usuarios que se encuentra en un data.sql

En la tabla usuarios tenemos los siguientes datos:

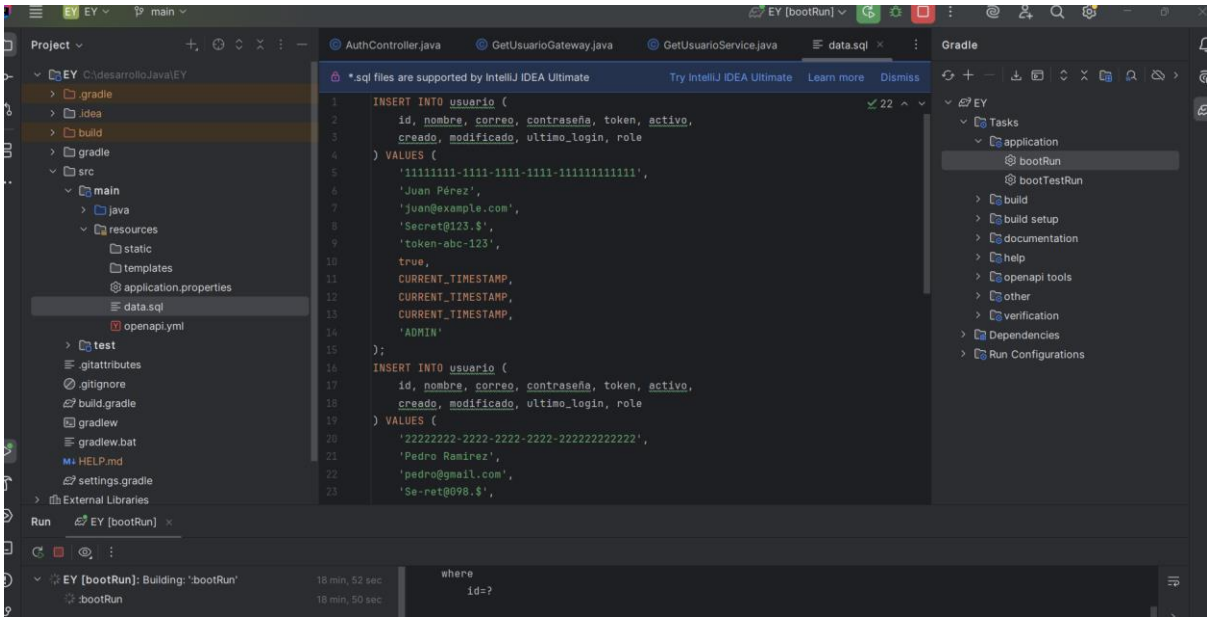
| Código | Correo | Contraseña | Rol |
|---|---|---|---|
| 1 | juan@example.com | Secret@123.$ | ADMIN |
| 2 | pedro@gmail.com | Se-ret@098.$ | ADMIN |
| 3 | elias.cornejo@coagra.cl | Eli@sC123.$ | USER |

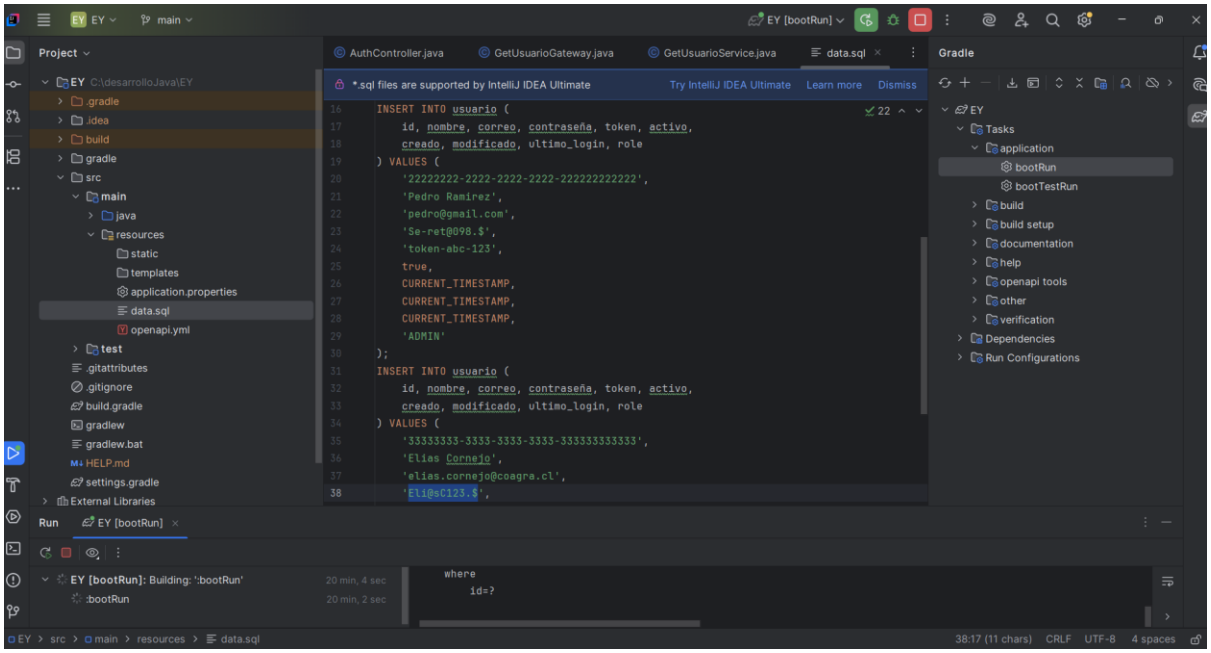
## Roles

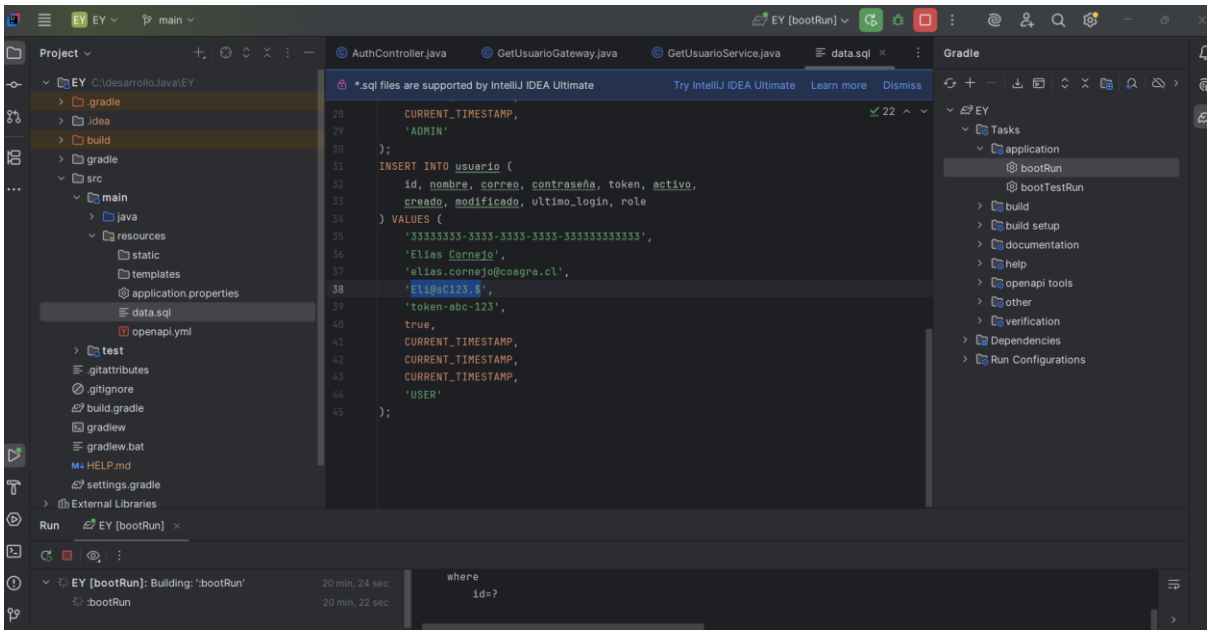
Los roles USER corresponden a usuarios à sólo puede gestionar lectura de los datos, es decir, sólo tiene Acceso a los Endpoint GET

Los roles ADMIN corresponden a usuario à pueden gestionar todo los endpoint, es decir, POST, PUT, PATCH, GET, DEL

### Data Pre-Cargada

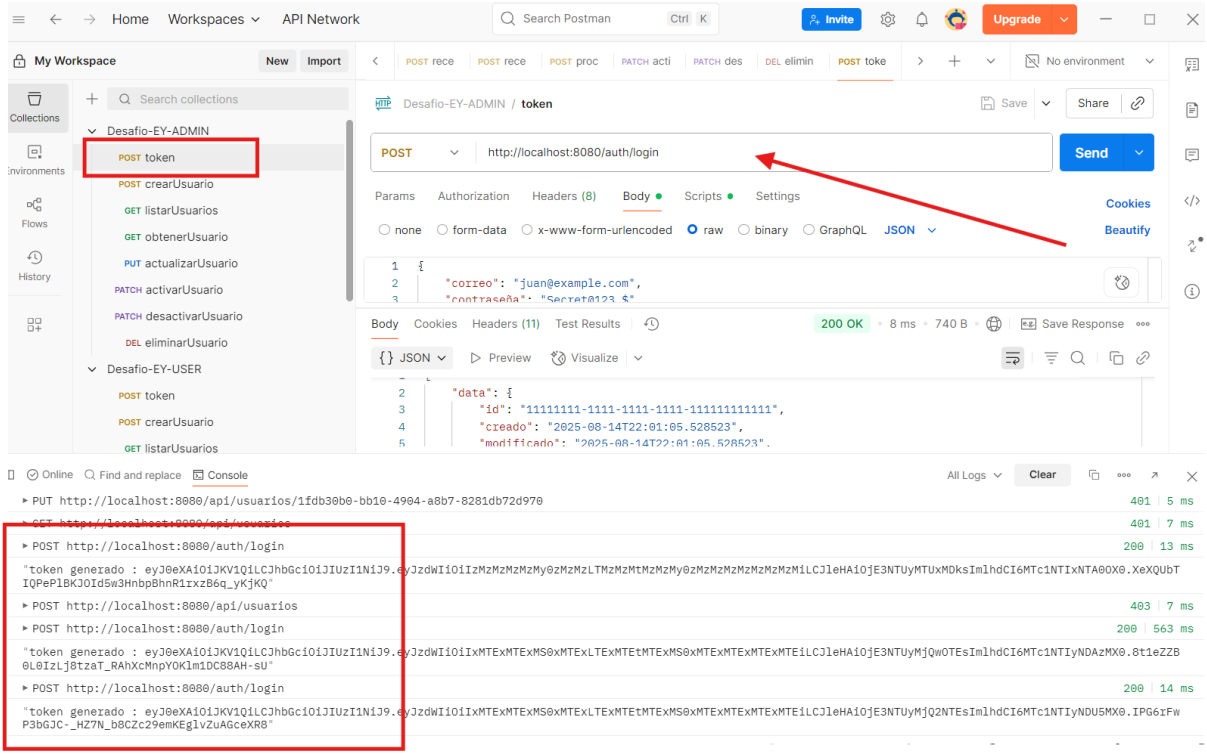




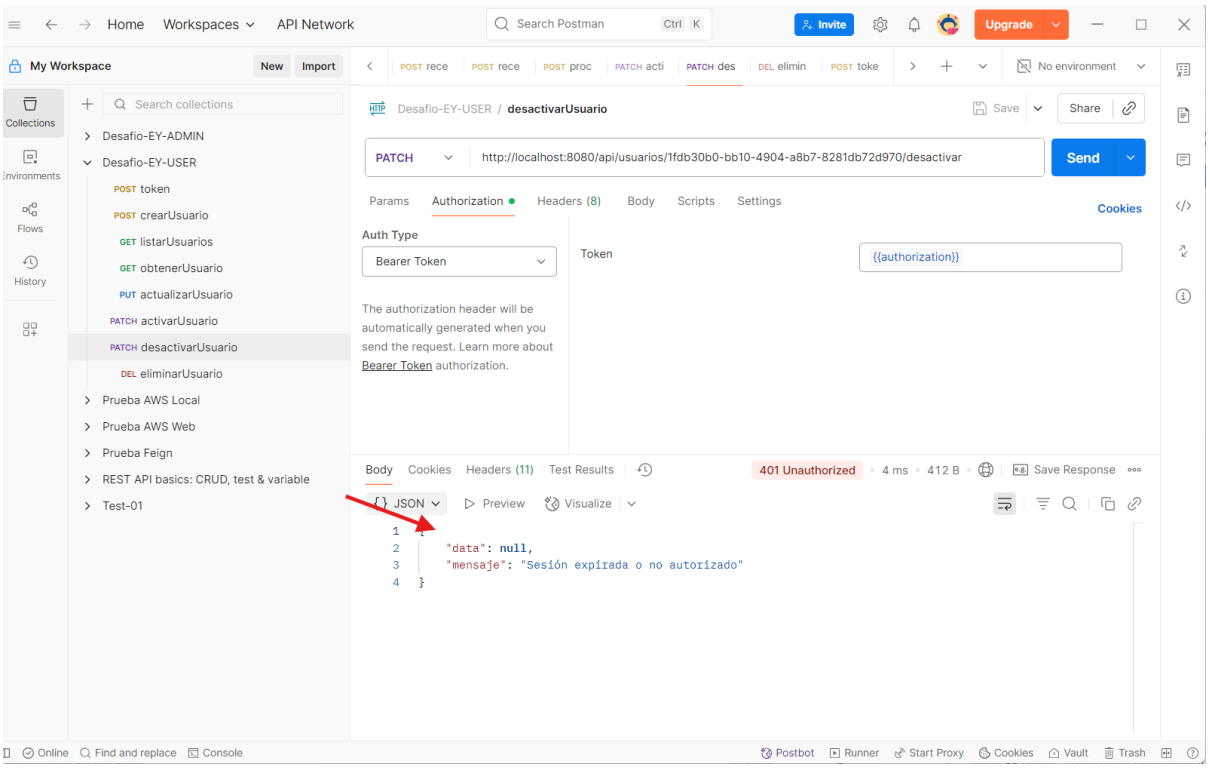




## Token

El token dura aproximadamente 1 minuto, está utilizando postman y pasa el minuto deberá gestionar nuevamente el token para volver a utilizar los endpoint




SI pasa más del minuto expira la sesión y está obligado a solicitar nuevamente el token para acceder a los endpoint



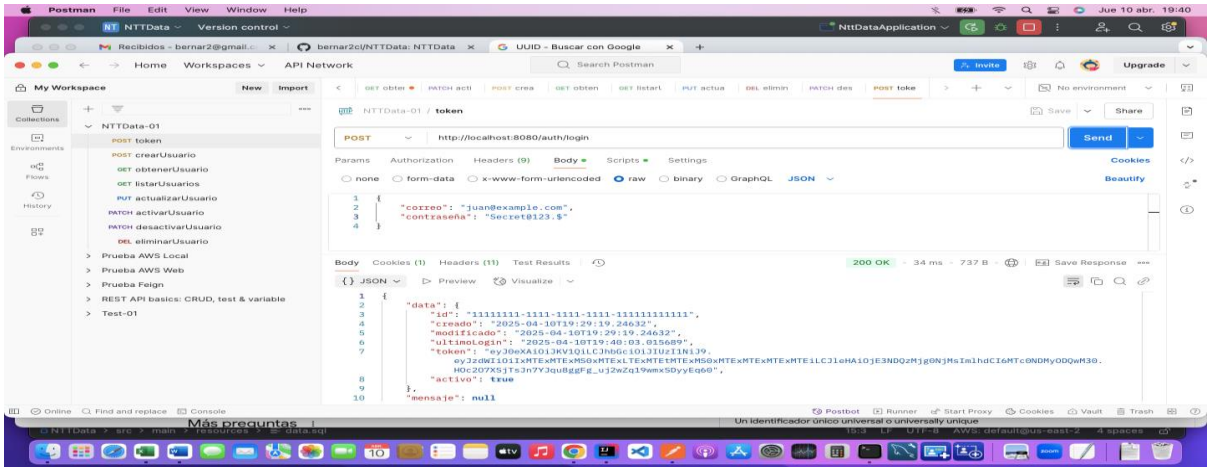

# Rol ADMIN

## POST - Token

Genero token para acceder a crear usuarios

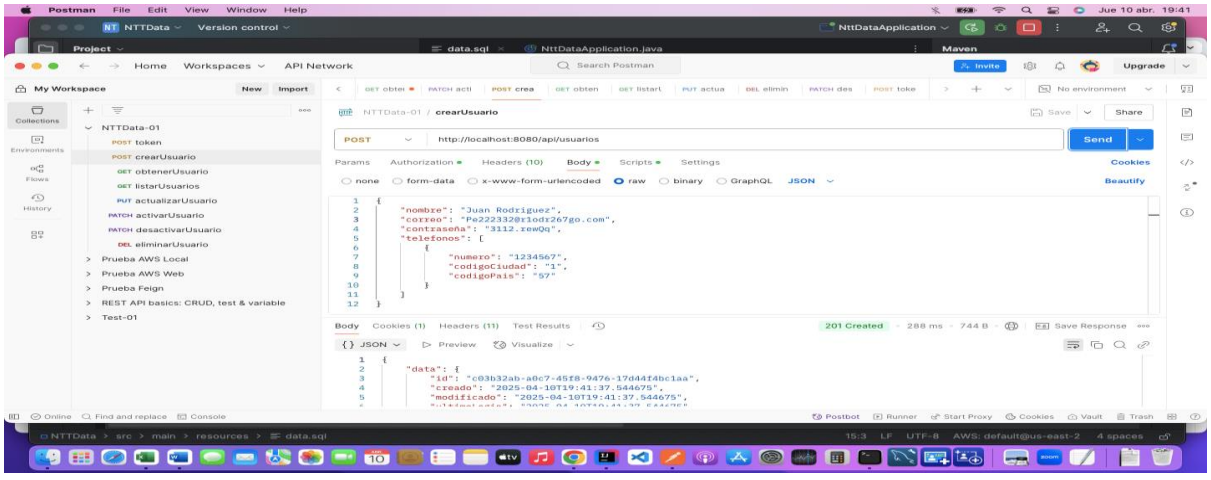
| Código | Correo | Contraseña | Rol |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | juan@example.com | Secret@123.$ | ADMIN |
| 2 | pedro@gmail.com | Se-ret@098.$ | ADMIN |
| 3 | elias.cornejo@coagra.cl | Eli@sC123.$ | USER |



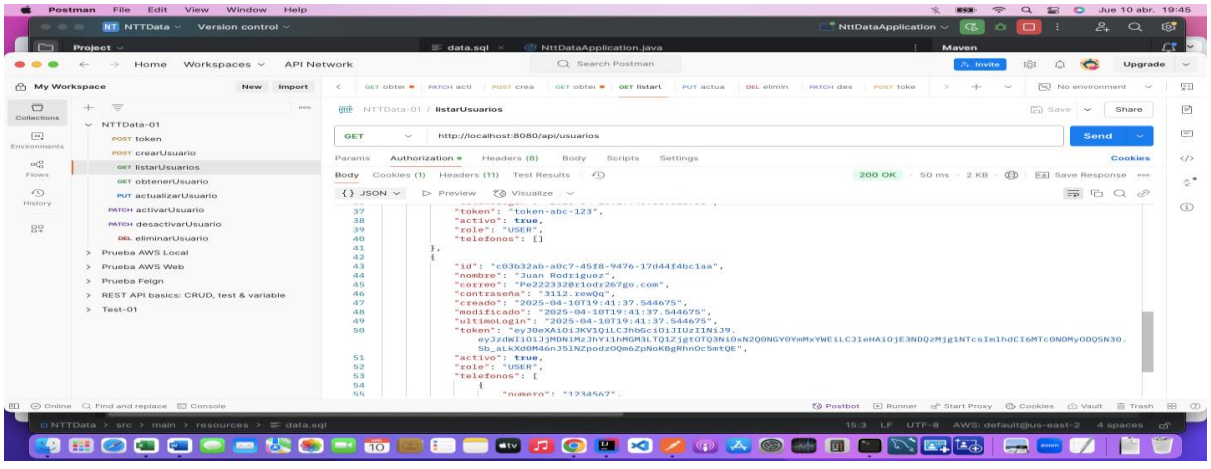
## POST - crearUsuario

Crear un nuevo usuario



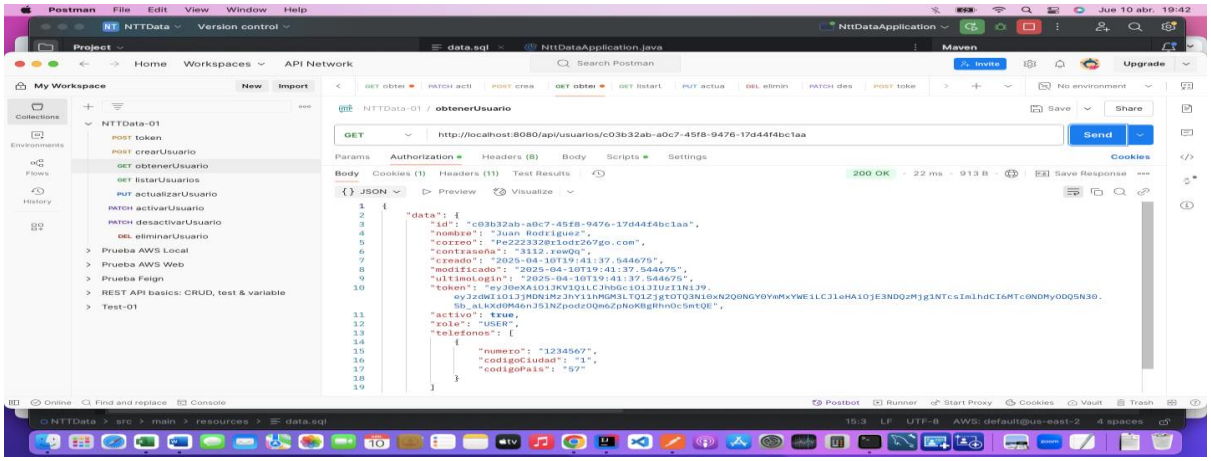
## GET - listarUsuarios

Recupera los usuarios registrados y saco el UUID



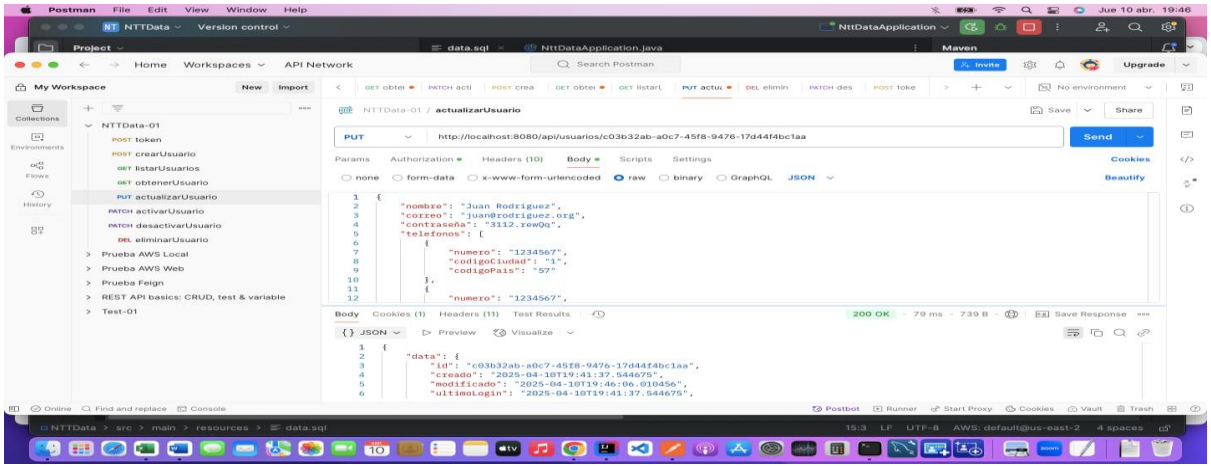
## GET - obtenerUsuario

Buscar el usuario por el UUID

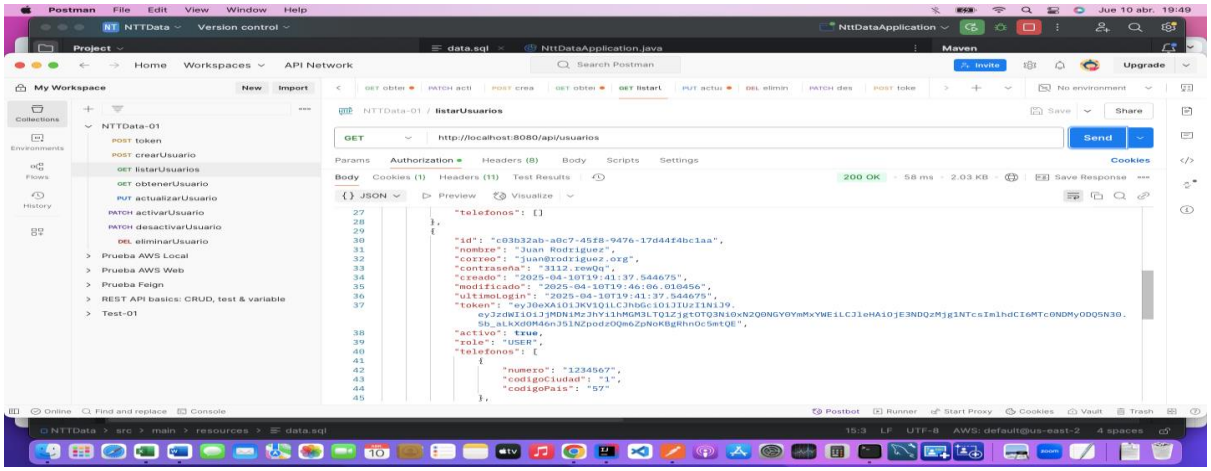


## PUT - actualizarUsuario

Se actualiza la tarea por UUID

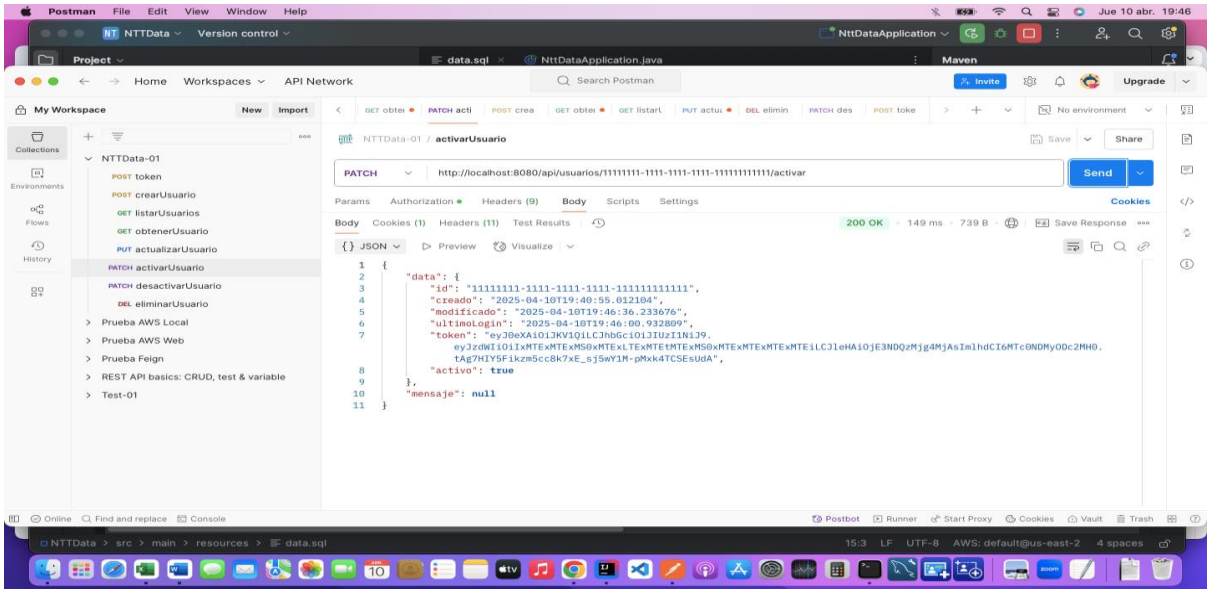


Se revisa la modificación



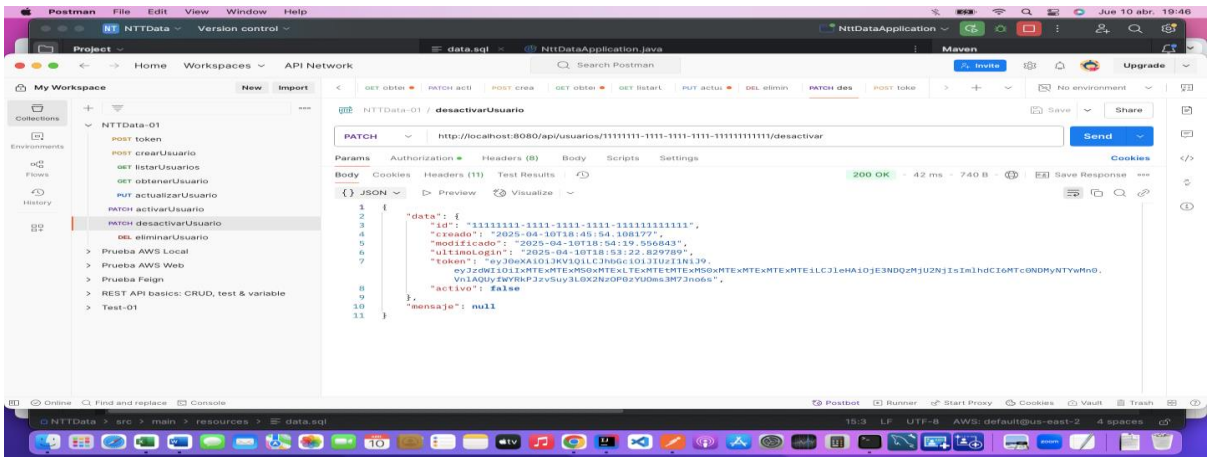
## PATCH - activarUsuario

Se actualiza el usuario por UUID



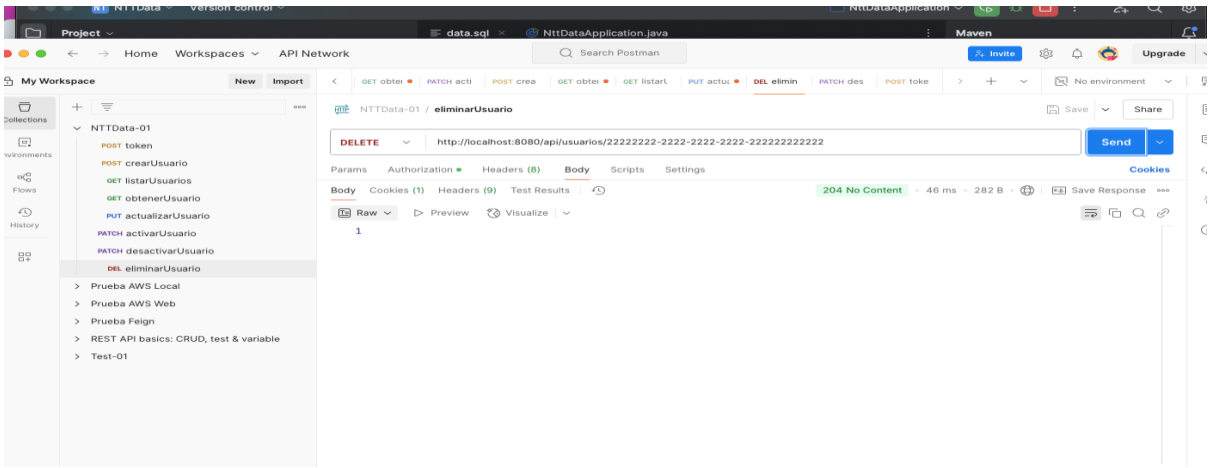

## PATCH - desactivarUsuario

Se actualiza el usuario por UUID

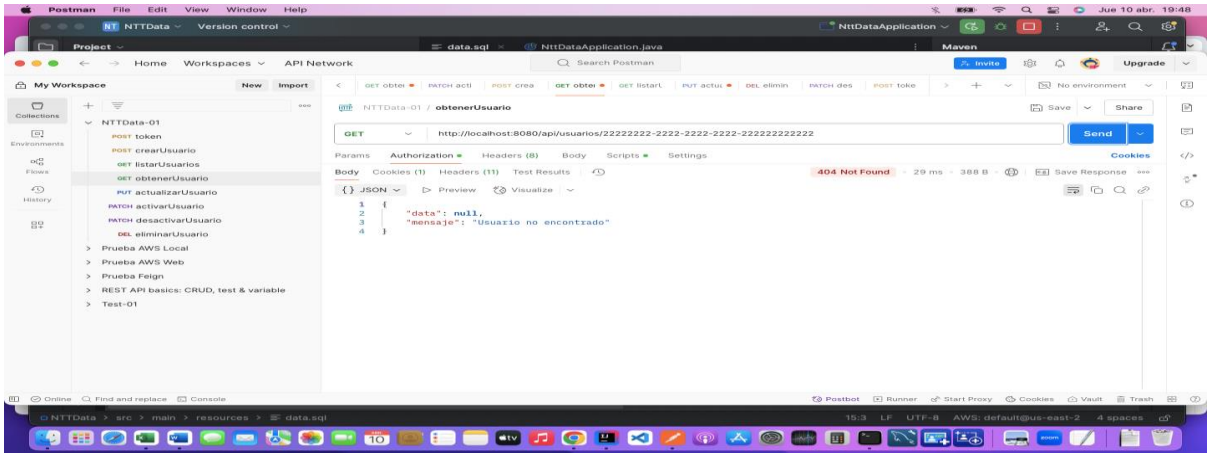



## DELETE - eliminarUsuario

Eliminar el usuario por el UUID




Revisar que si está eliminado

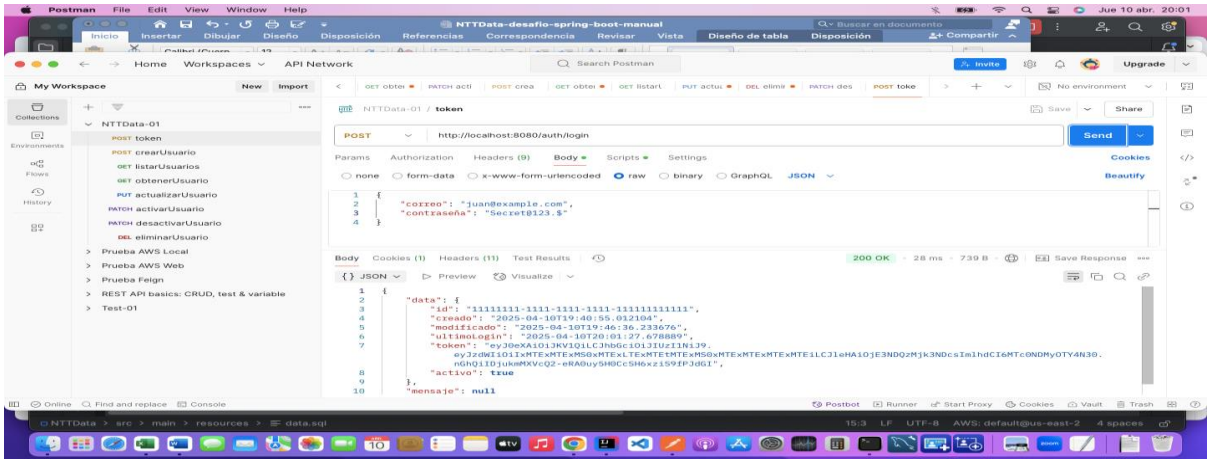



## Rol ADMIN – CREAR UN USUARIO Y SACAR TOKEN

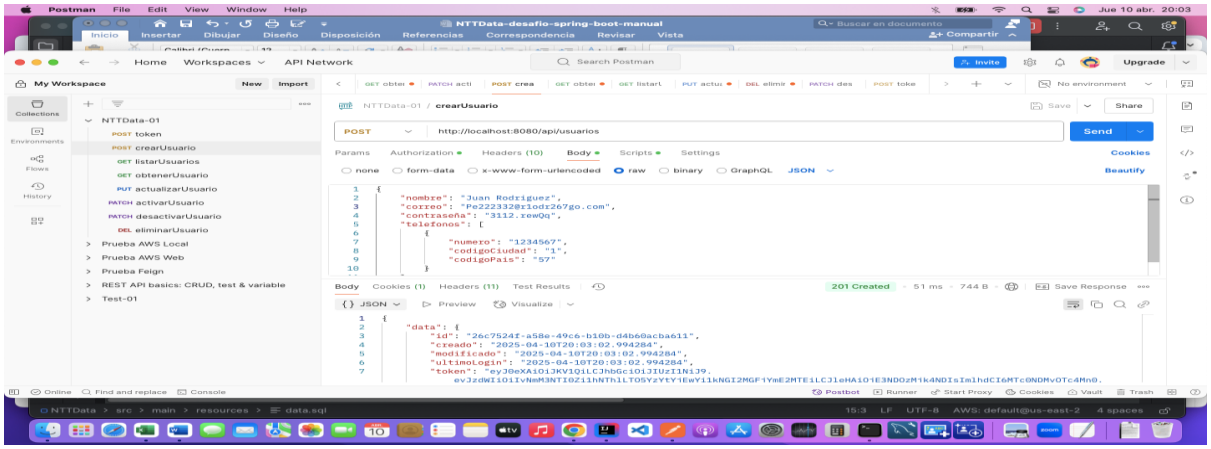
### POST - Token

Genero token para acceder a crear usuarios

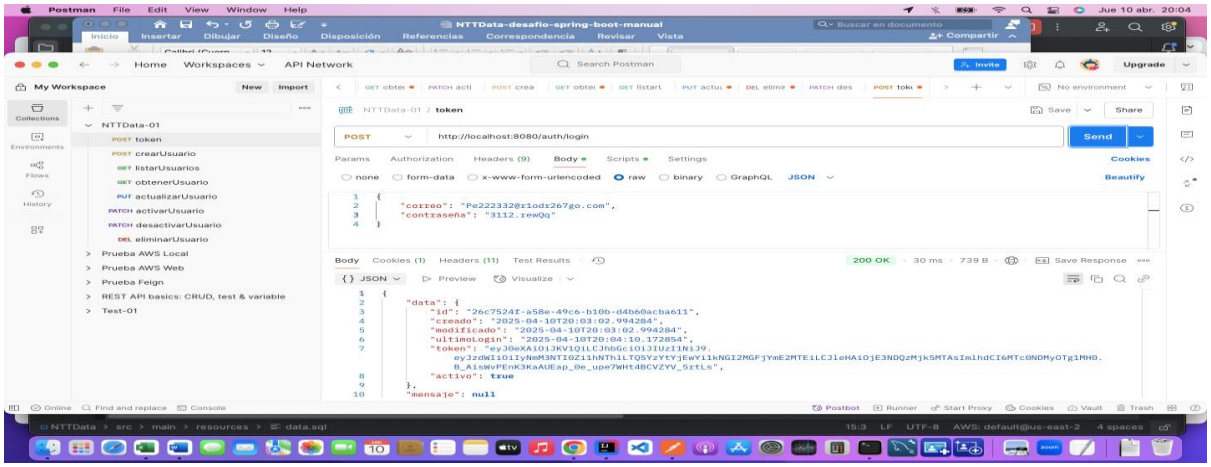
| Código | Correo | Contraseña | Rol |
|---|---|---|---|
| 1 | juan@example.com | Secret@123.$ | ADMIN |
| 2 | pedro@gmail.com | Se-ret@098.$ | ADMIN |
| 3 | elias.cornejo@coagra.cl | Eli@sC123.$ | USER |




Creo el usuario

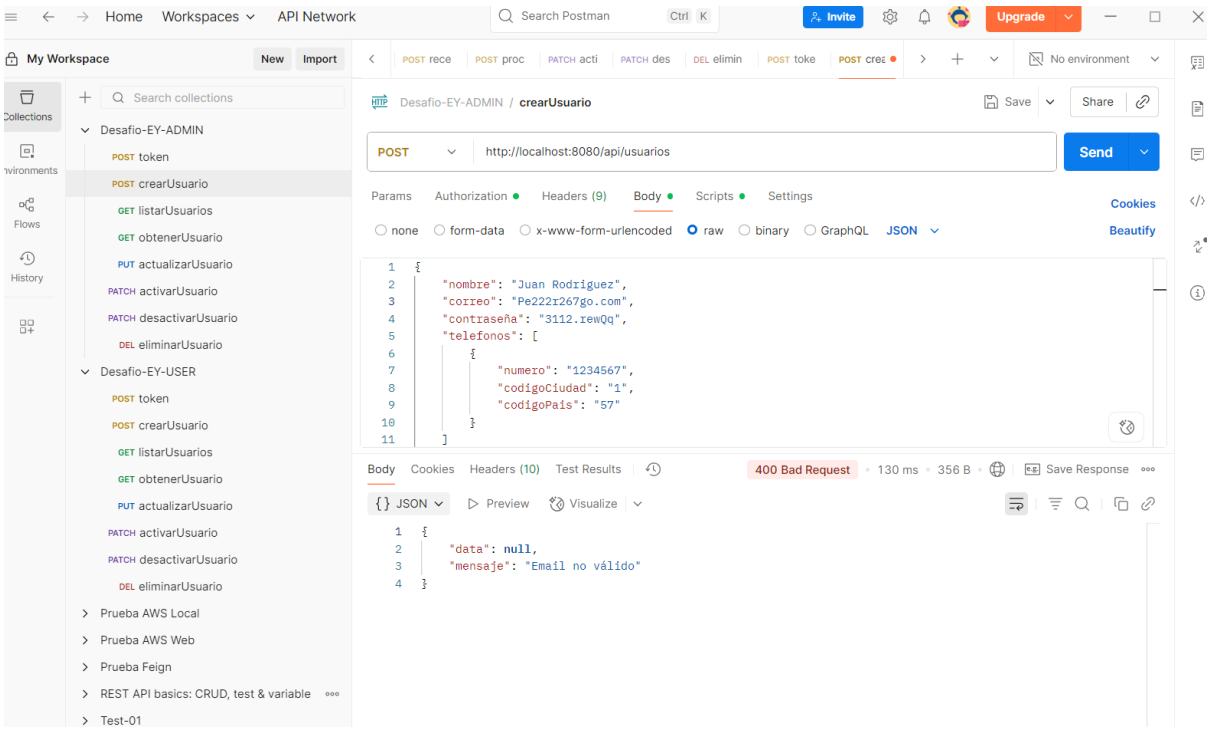


Se inicia sesión con el nuevo usuario



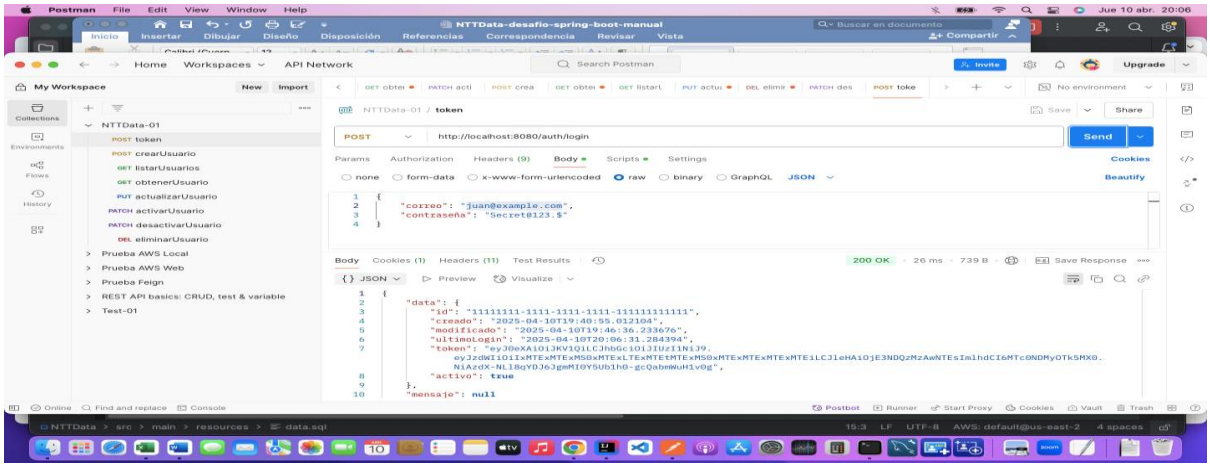
*Validar correo*

No tiene formato correo




*Validar constraseña*

Iniciar sesión



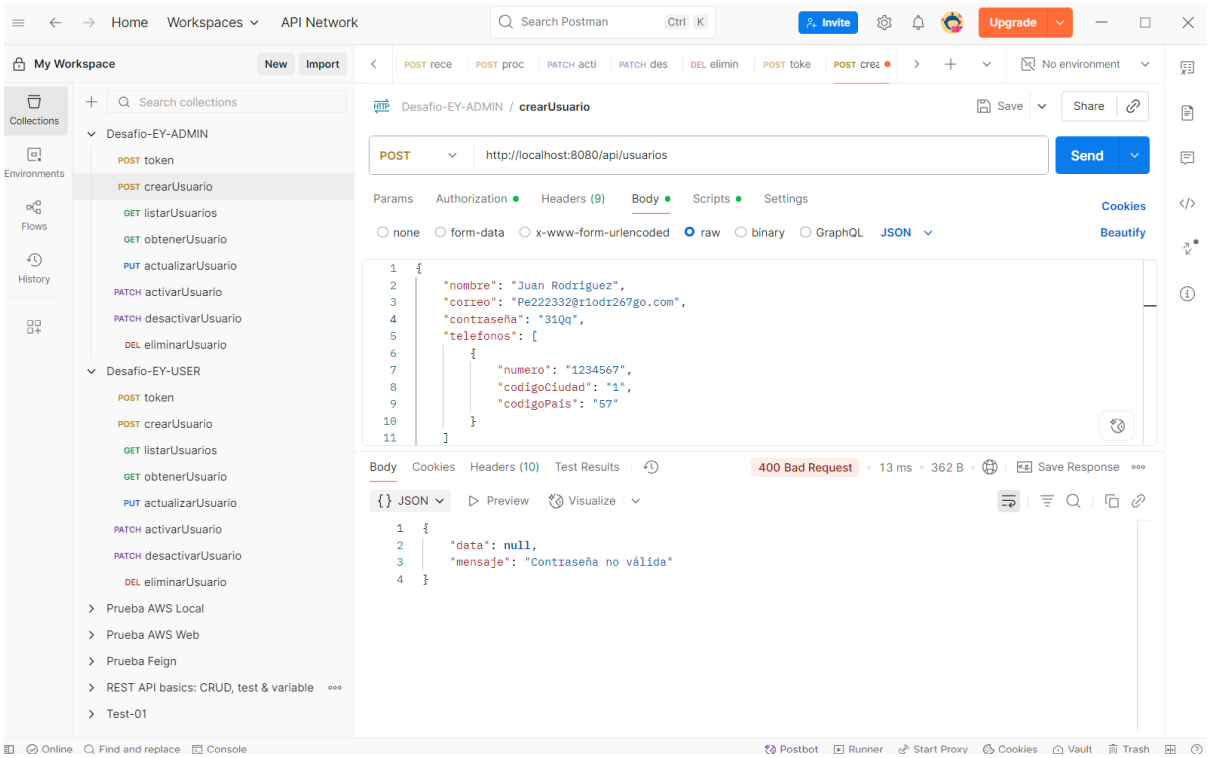
Contraseña no adapta. Se sugiere cambiar una más fuerte.

Una contraseña válida debe tener al menos:

• Una letra mayúscula (A-Z)

• Un número (0-9)

• Seis caracteres o más



## Rol USER

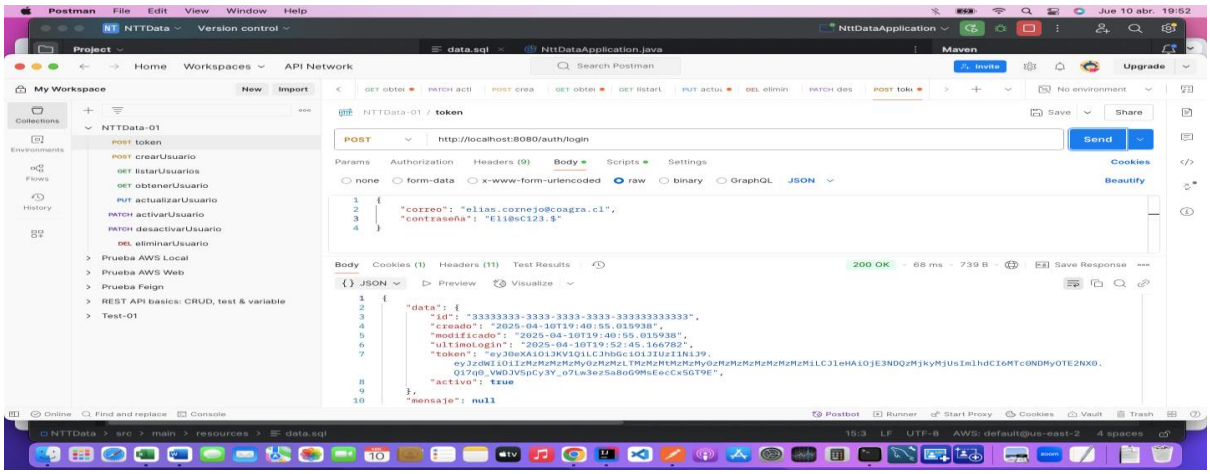
### POST - Token

Genero token para acceder a crear tareas

| Código | Correo | Contraseña | Rol |
|---|---|---|---|
| 1 | juan@example.com | Secret@123.$ | ADMIN |
| 2 | pedro@gmail.com | Se-ret@098.$ | ADMIN |

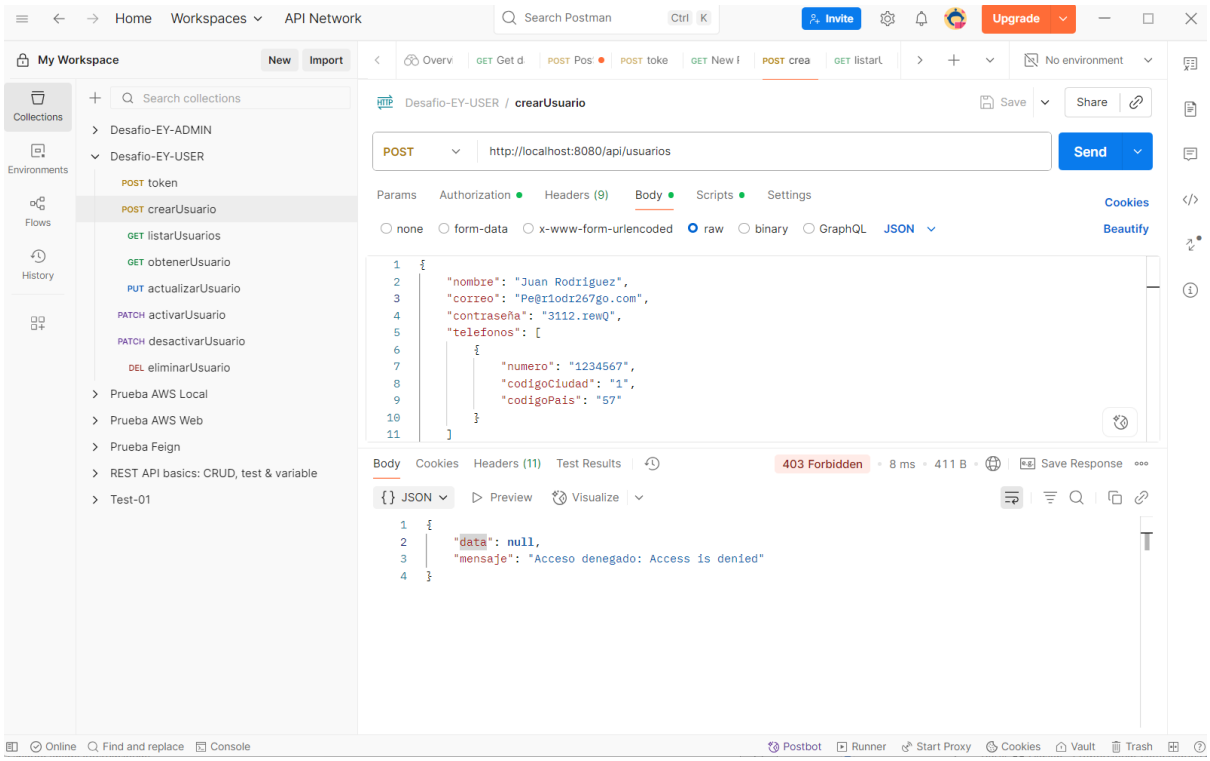
| | | | |
|---|---|---|---|
| 3 | elias.cornejo@coagra.cl | Eli@sC123.$ | USER |

Iniciar sesion



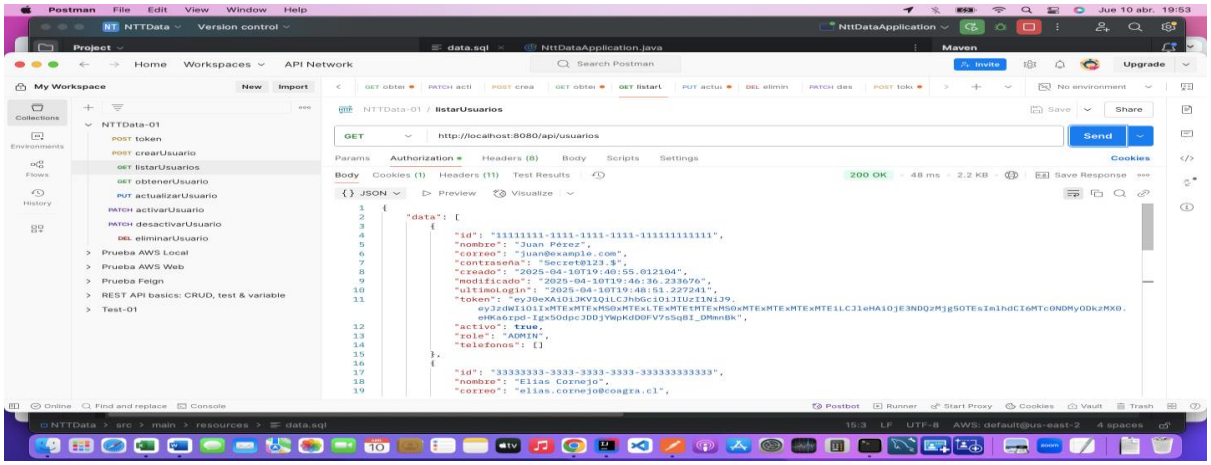

## POST - crearUsuario

No tienen autorización



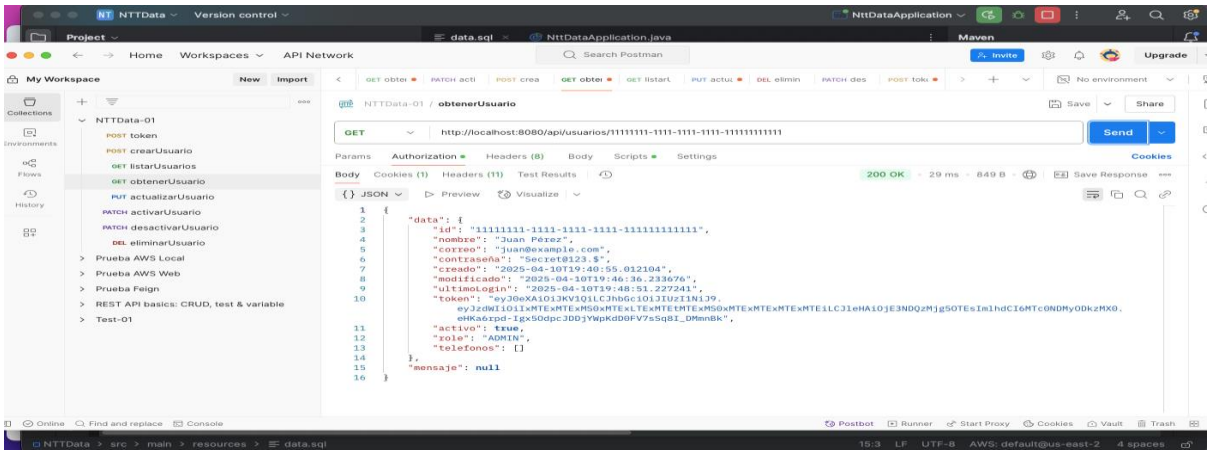

## GET - listarUsuarios

Recupera los usuarios registrados



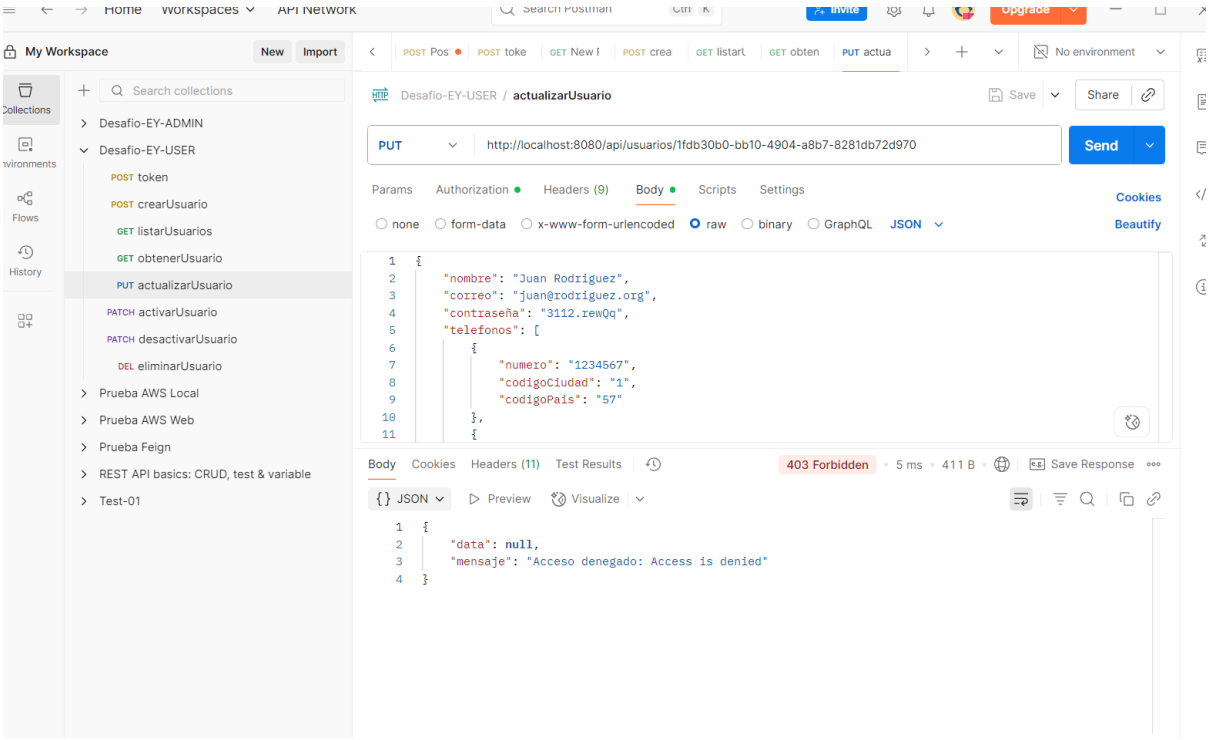
## GET - obtenerUsuario

Buscar el usuario por el UUID



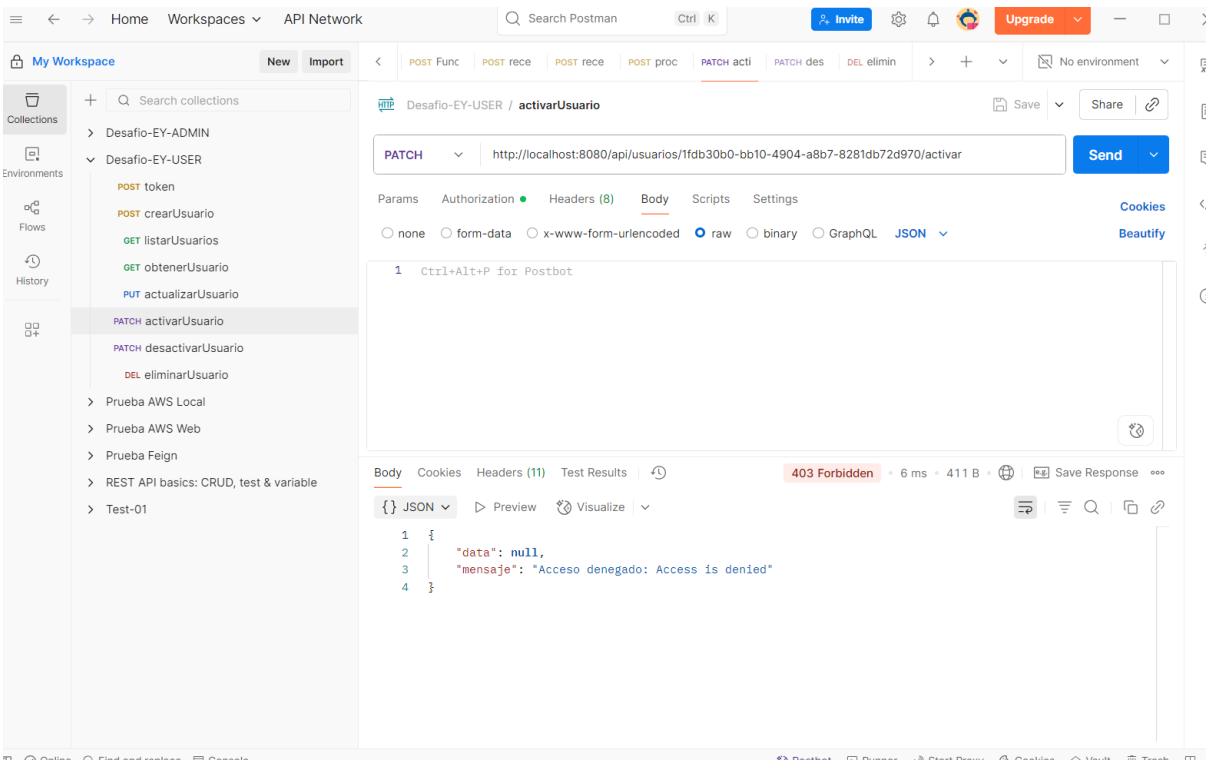
## PUT - actualizarUsuario

No tienen autorización



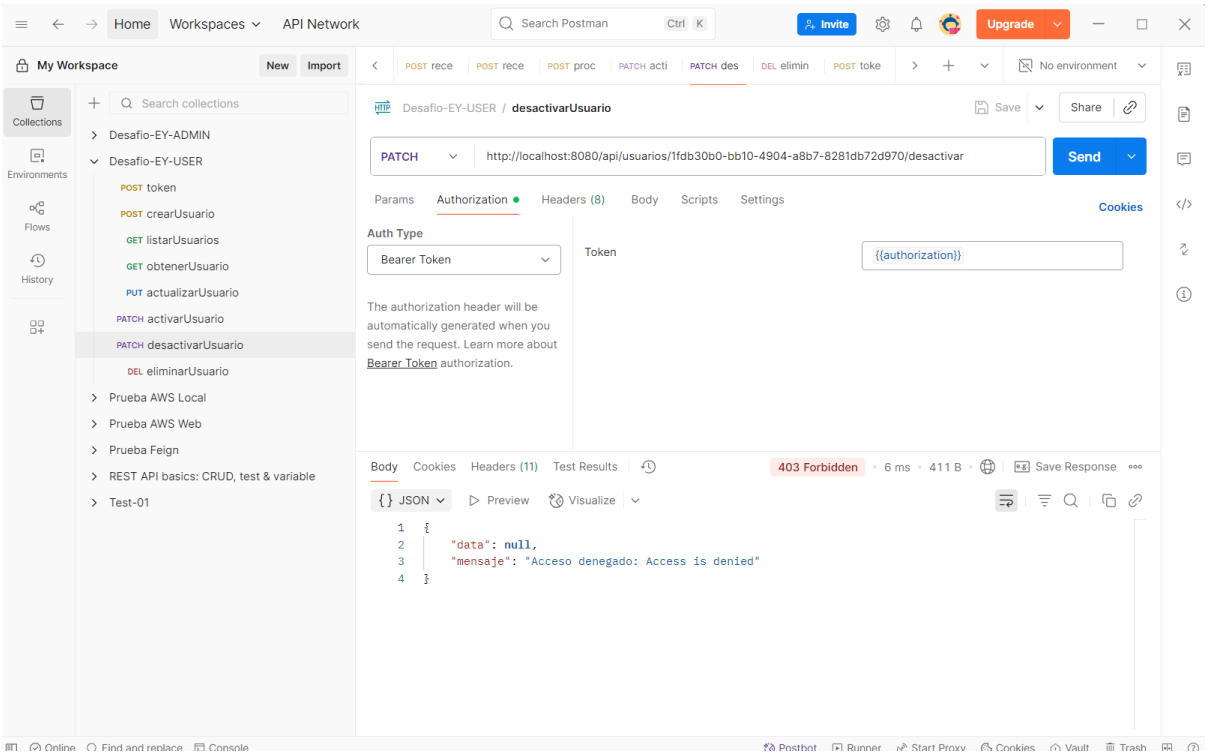

## PATCH - activarUsuario

No tienen autorización




## PATCH - desactivarUsuario

No tienen autorización




# DELETE - eliminarUsuario

## No tiene autorización