## INFORME DE PRUEBAS DE LA API "REALESTATE"

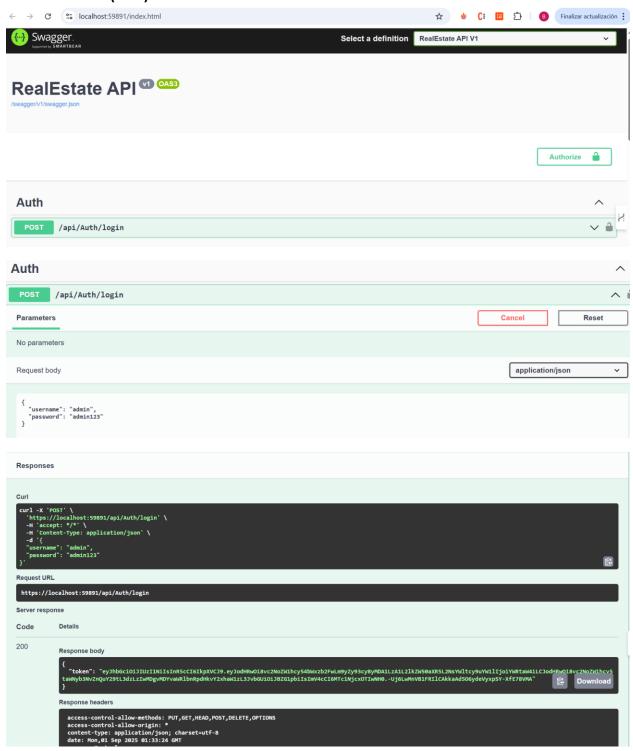
## Análisis de Pruebas Unitarias y de Integración Utilizando Swagger

**Autor: Ing. Brian Alberto Bernal Castillo** 

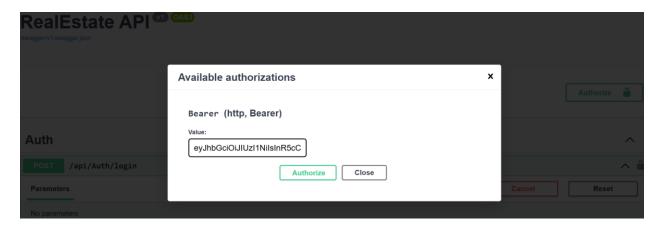
Proyecto: Prueba Técnica Million

Fecha: 1 de septiembre de 2025

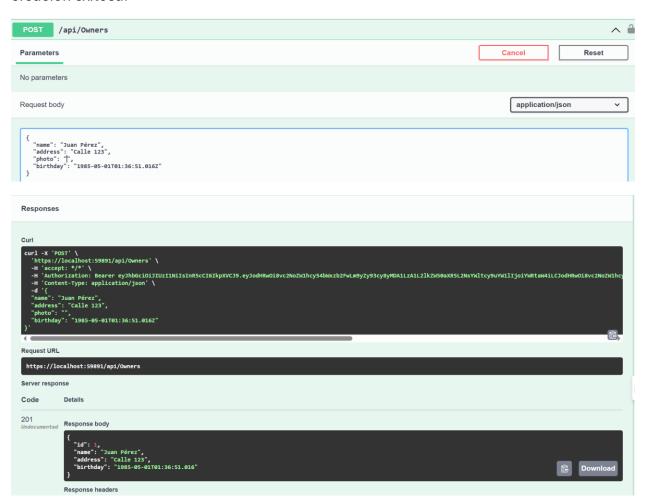
Autenticación (POST /api/Auth/login): Se inicia sesión con las credenciales username: "admin" y password: "admin123". La API responde con un código 200 OK y un token de autenticación (JWT).



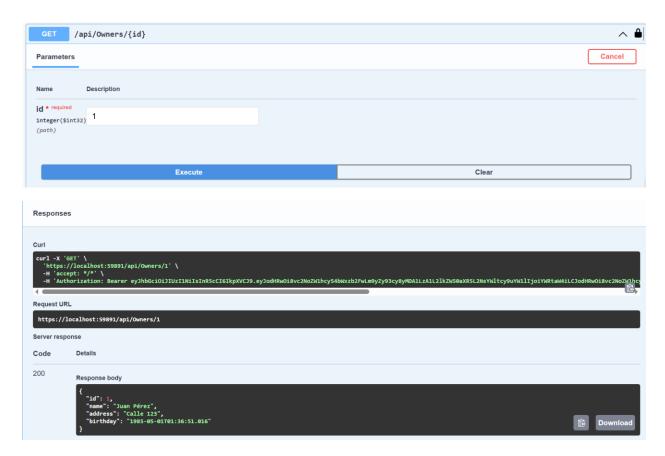
**Autorización (Token Bearer)**: El token recibido se utiliza para autorizar futuras solicitudes a los endpoints protegidos de la API.



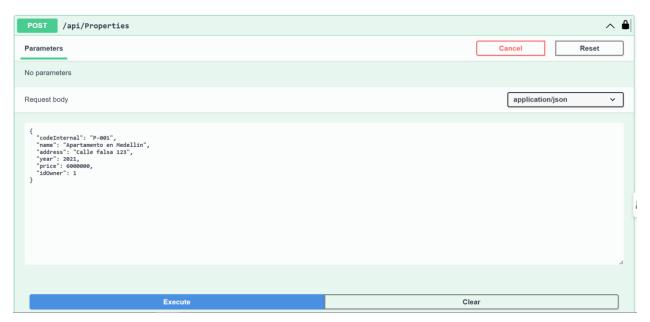
**Creación de Propietario (POST /api/Owners)**: Se envía una solicitud para crear un nuevo propietario ("Juan Pérez"). La API responde con un código 201 Created, confirmando la creación exitosa.



**Obtención de Propietario (GET /api/Owners/{id})**: Se verifica la creación del propietario solicitando sus datos por su id. La API responde con un código 200 OK y devuelve la información del propietario.

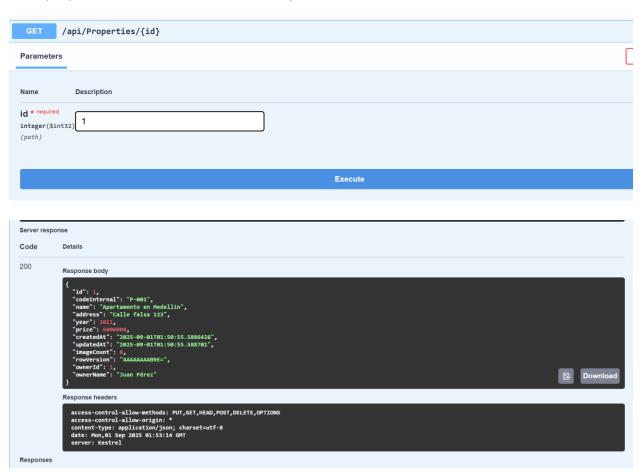


Creación de Propiedad (POST /api/Properties): Se crea una nueva propiedad ("Apartamento en Medellín") asociada al propietario creado en el paso anterior. La API responde con un 201 Created.



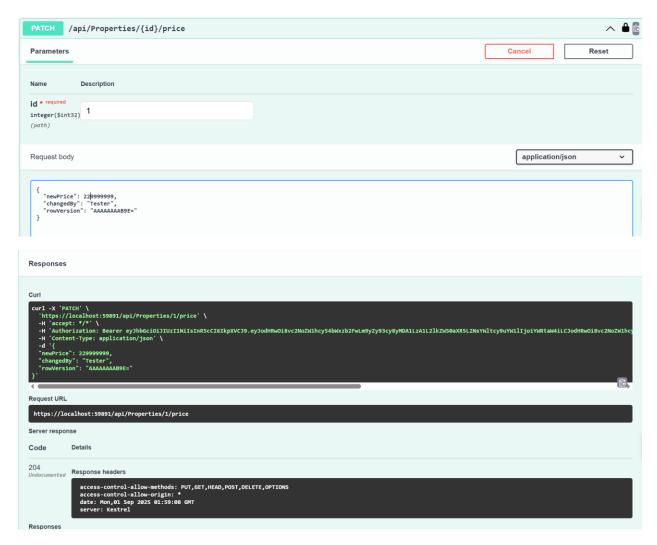


**Obtención de Propiedad (GET /api/Properties/{id}):** Se recuperan los detalles completos de la propiedad recién creada. La API responde con un 200 OK.



Actualización de Precio (PATCH /api/Properties/{id}/price): Se actualiza el precio de la propiedad. Se envía el nuevo valor y el rowversion para manejar la concurrencia. La API

responde con un 204 No Content, indicando que la operación se completó sin devolver datos.



**Subida de Imagen (POST /api/Properties/{id}/images)**: Se muestra el proceso de subir un archivo de imagen (Estructura.PNG) a la propiedad. La solicitud utiliza multipart/form-data

## para enviar el archivo.

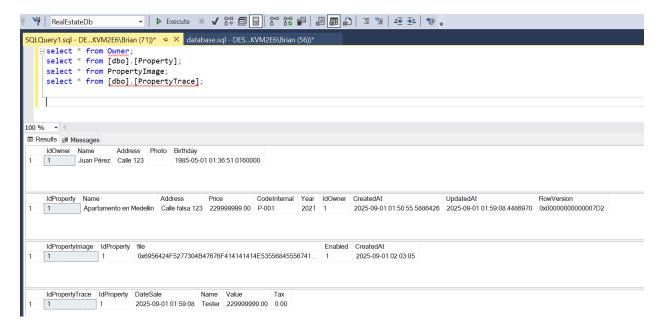
POST /api/Properties/{id}/images		^ •
Parameters		Cancel Reset
Name Description		
id * required integer(\$int32)  (path)		
Request body		multipart/form-data >
file * required string(\$binary)  Seleccionar archivo Estructura.PNG		
enabled boolean true	Send empty value	
Execute		

**Owner**: Almacena información sobre los propietarios de inmuebles (IdOwner, Name, Address, Photo, Birthday).

**Property**: Contiene los detalles de las propiedades (IdProperty, Name, Address, Price, Year, IdOwner).

**PropertyImage**: Relaciona imágenes (file) con una propiedad específica (IdProperty). El tipo de dato VARBINARY se utiliza para almacenar la imagen.

**PropertyTrace**: Registra el historial de cambios en las propiedades, como las ventas.



## **Pruebas Unitarias**

Se muestra el resultado exitoso de las pruebas unitarias, con todas las pruebas marcadas con un check verde. Esto confirma que los componentes del código funcionan como se esperaba de forma aislada. Las pruebas clave incluyen:

- ChangePrice\_Stores\_History\_And...: Verifica que los cambios de precio de una propiedad se registran en el historial.
- **Create\_And\_GetById\_Works\_Wi...**: Confirma que se puede crear una propiedad y luego recuperarla correctamente usando su identificador.
- **List\_With\_Filters\_Returns\_Pagin...**: Valida que la funcionalidad de listado de propiedades con filtros y paginación opera correctamente.

