

# **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)**.

[illegible]

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



[illegible]

POST

/api/Properties

⌵ 🔒

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "codeInternal": "P-001",
  "name": "Apartamento en Medellín",
  "address": "Calle falsa 123",
  "year": 2021,
  "price": 6000000,
  "idOwner": 1
}
```

Execute

Clear

Server response

Code

Details

201

Undocumented

Response body

```
{
  "id": 1,
  "codeInternal": "P-001",
  "name": "Apartamento en Medellin",
  "address": "Calle falsa 123",
  "year": 2021,
  "price": 6000000,
  "createdAt": "2025-09-01T01:50:55.5886426",
  "updatedAt": "2025-09-01T01:50:55.588701",
  "imageCount": 0,
  "rowVersion": "AAAAAAAAB9E=",
  "ownerId": 1,
  "ownerName": "Juan Pérez"
}
```

Download

Response headers

```
access-control-allow-methods: PUT,GET,HEAD,POST,DELETE,OPTIONS
access-control-allow-origin: *
content-type: application/json; charset=utf-8
date: Mon, 01 Sep 2025 01:50:54 GMT
location: https://localhost:59891/api/Properties/1
server: Kestrel
```

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

GET

/api/Properties/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	1
(path)	

Execute

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "codeInternal": "P-001",
  "name": "Apartamento en Medellin",
  "address": "Calle falsa 123",
  "year": 2021,
  "price": 6000000,
  "createdAt": "2025-09-01T01:50:55.5886426",
  "updatedAt": "2025-09-01T01:50:55.588701",
  "imageCount": 0,
  "rowVersion": "AAAAAAAAB9E=",
  "ownerId": 1,
  "ownerName": "Juan Pérez"
}
```

Download

Response headers

```
access-control-allow-methods: PUT,GET,HEAD,POST,DELETE,OPTIONS
access-control-allow-origin: *
content-type: application/json; charset=utf-8
date: Mon, 01 Sep 2025 01:53:14 GMT
server: Kestrel
```

**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.

[illegible]

**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

Name	Description
id * required	
integer(\$int32)	1
(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.

```
SQLQuery1.sql - DE...KVM2E6(Brian (71))* database.sql - DES...KVM2E6(Brian (56))*
select * from Owner;
select * from [dbo].[Property];
select * from PropertyImage;
select * from [dbo].[PropertyTrace];
```

100 %

Results Messages

IdOwner	Name	Address	Photo	Birthday
1	Juan Pérez	Calle 123		1985-05-01 01:36:51.0160000

IdProperty	Name	Address	Price	CodeInternal	Year	IdOwner	CreatedAt	UpdatedAt	RowVersion
1	Apartamento en Medellin	Calle falsa 123	229999999.00	P-001	2021	1	2025-09-01 01:50:55.5886426	2025-09-01 01:59:08.4486970	0x000000000000007D2

IdPropertyImage	IdProperty	file	Enabled	CreatedAt
1	1	0x6956424F5277304B47676F4141414E53556845556741...	1	2025-09-01 02:03:05

IdPropertyTrace	IdProperty	DateSale	Name	Value	Tax
1	1	2025-09-01 01:59:08	Tester	229999999.00	0.00

## 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\_GetByld\_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.

Test	Duration	Tra
▲ ✓ RealEstate.Tests (3)	831 ms	
▲ ✓ RealEstate.Tests (3)	831 ms	
▲ ✓ PropertyServiceTests (3)	831 ms	
✓ ChangePrice_Stores_History_An...	736 ms	
✓ Create_And_GetByld_Works_Wi...	34 ms	
✓ List_With_Filters_Returns_Pagin...	61 ms	