

RAMPART

Experimento 1 - Final

Paula J. Alvarado Zabala, Ana C. Fandiño de la Hoz, Miguel Á. Puentes Ramírez, Diego F. Riveros Lancheros, Diego A. Solano Ariza

201313033, 201326407, 2013

ISIS2503 - Arquitectura y Diseño de Software

Universidad de los Andes, Bogotá, Colombia

{pj.alvarado10, ac.fandino10, ma.puentes, d.riveros11, da.solano1}@uniandes.edu.co

Fecha de presentación: Marzo 25 de 2018

# **Contenido**

[**1.** **Escenarios de Prueba** 2](#_Toc509697619)

[**1.1.** **POSTMAN** 2](#_Toc509697620)

[**1.2.** **JMeter** 20](#_Toc509697621)

[**2.** **Diagrama de Despliegue** 21](#_Toc509697622)

[**3.** **Comparación de resultados** 22](#_Toc509697623)

[**4.** **Reflexión Escenarios de Calidad** 24](#_Toc509697624)

[**Desempeño** 24](#_Toc509697625)

[**Escalabilidad** 25](#_Toc509697626)

[**Disponibilidad** 26](#_Toc509697627)

[**Seguridad** 26](#_Toc509697628)

[**5.** **Declaración de esfuerzo de cada miembro** 27](#_Toc509697629)

# **Escenarios de Prueba**

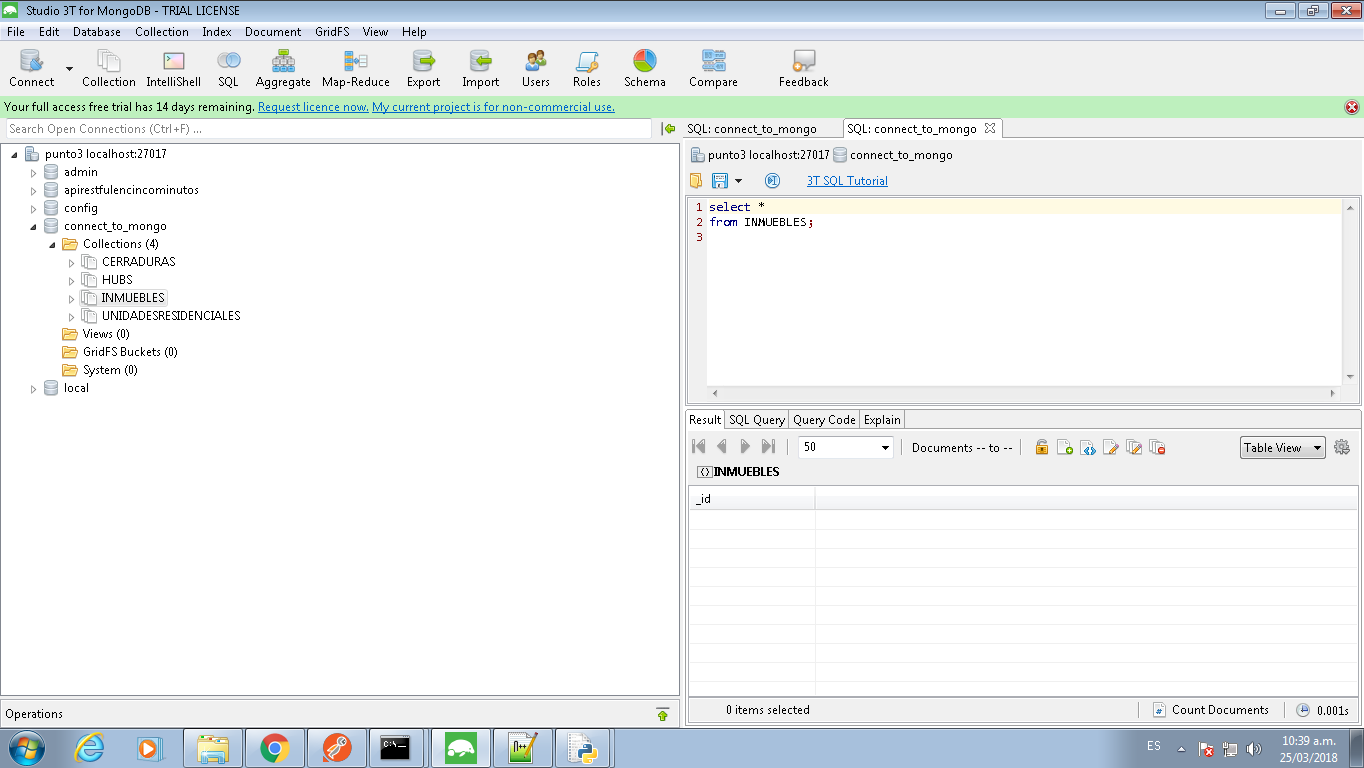
## **POSTMAN**

Adjunto al documento, se encuentra una carpeta zip con las colecciones de Postman utilizadas para poblar la base de datos.

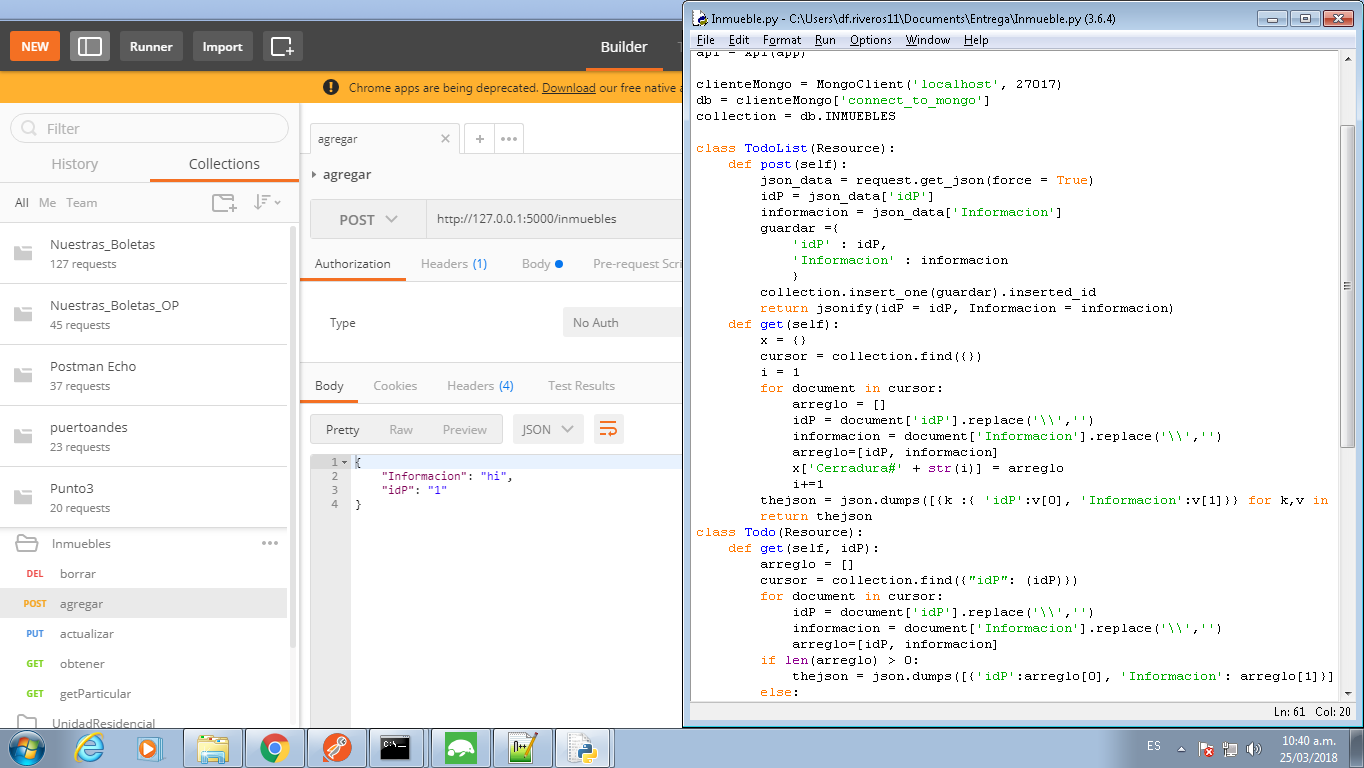
**INMUEBLE**

**Servicio POST:**

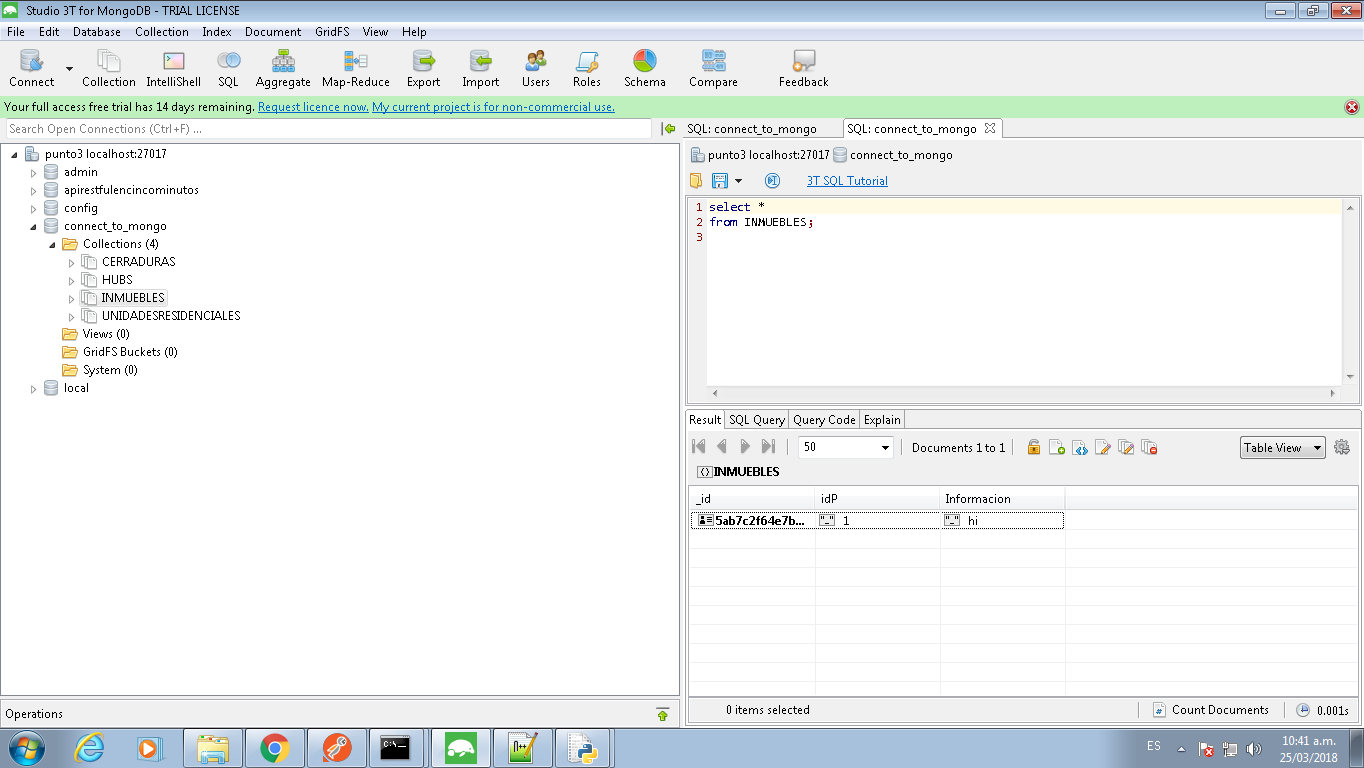
La base de datos vacía:



Postman (http://127.0.0.1:5000/inmuebles):



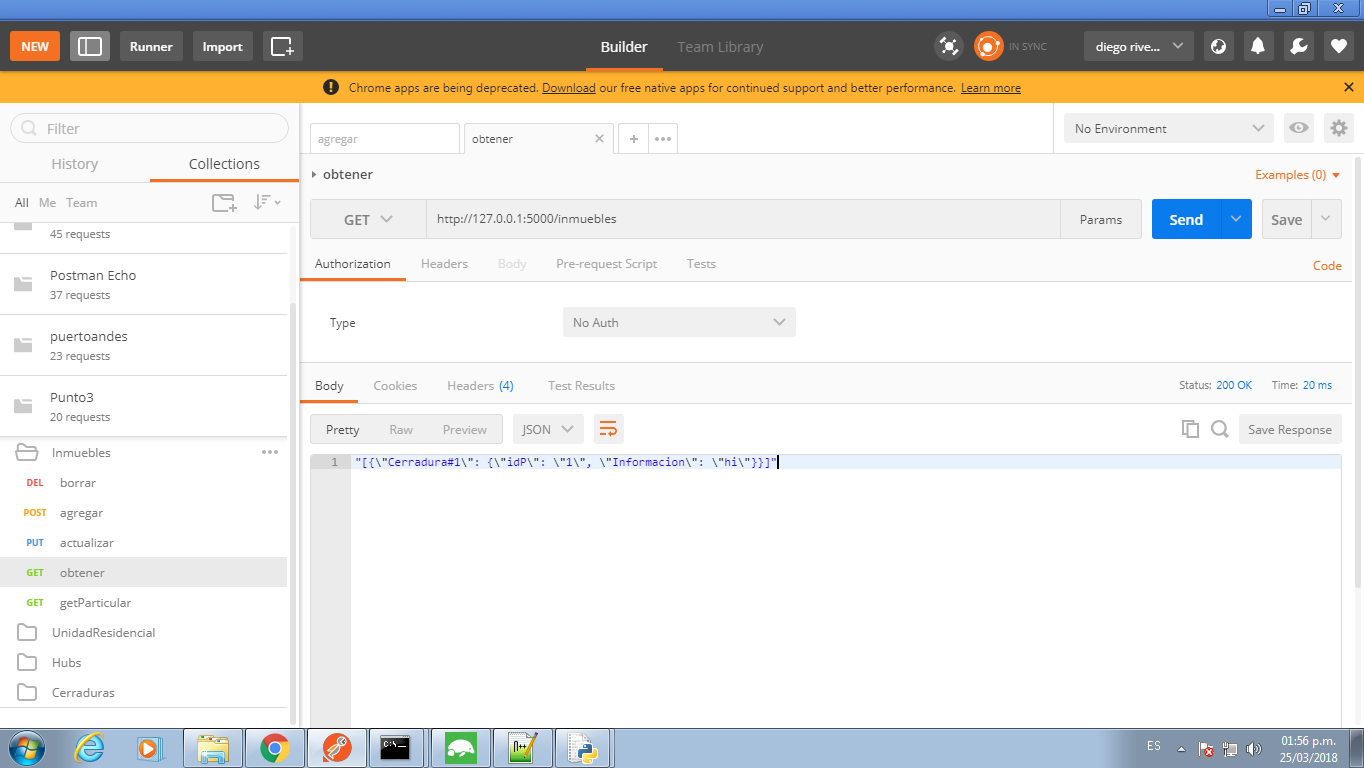
Base datos con el archivo agregado:



**Servicio GET:**

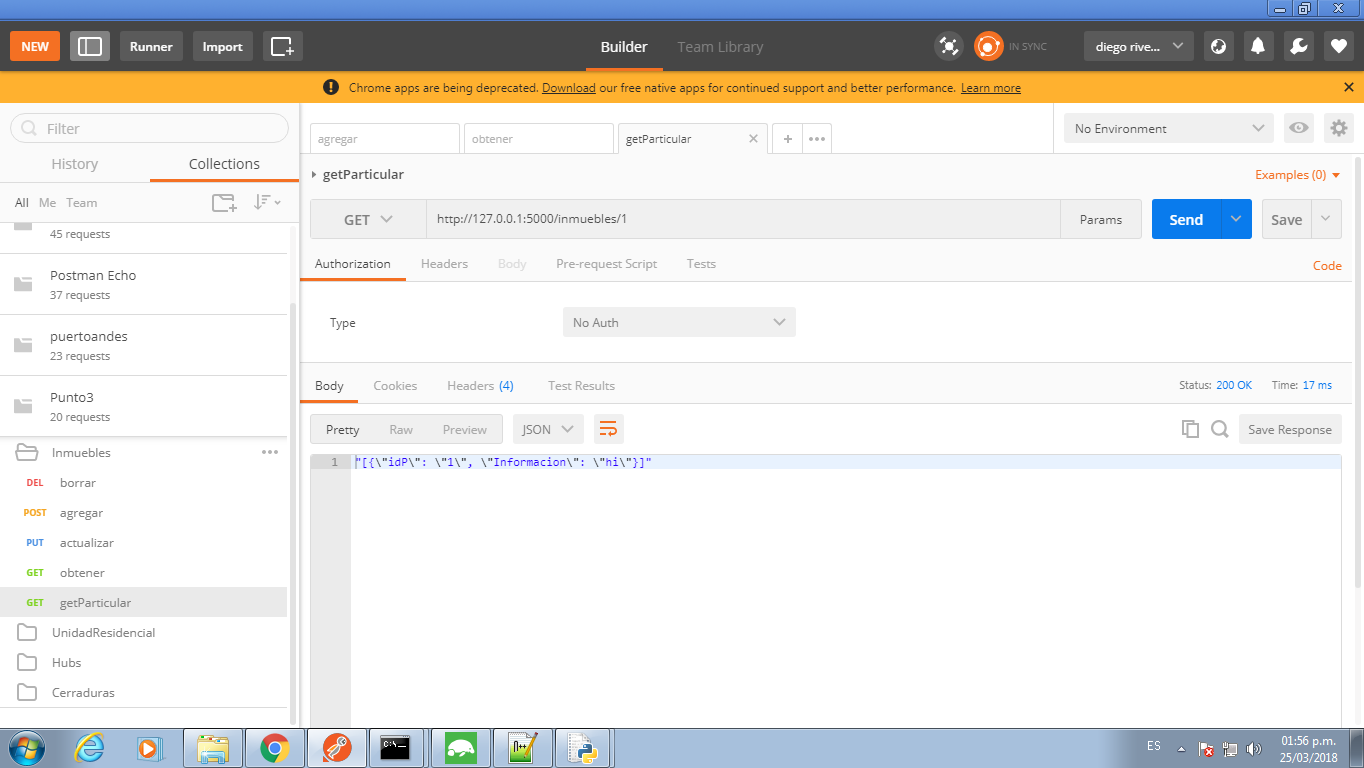
**GETALL:**

<http://127.0.0.1:5000/inmuebles>



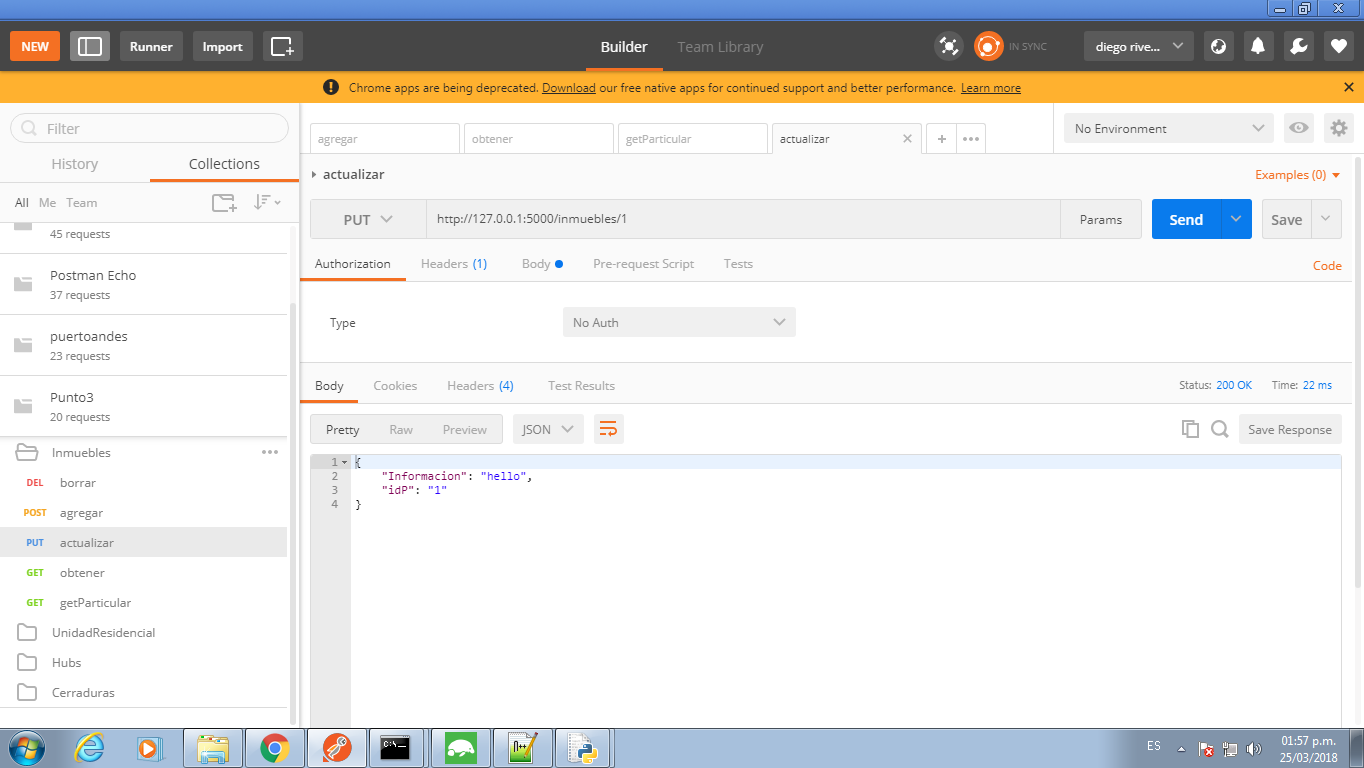
**GETPARTICULAR:**

<http://127.0.0.1:5000/inmuebles/1>

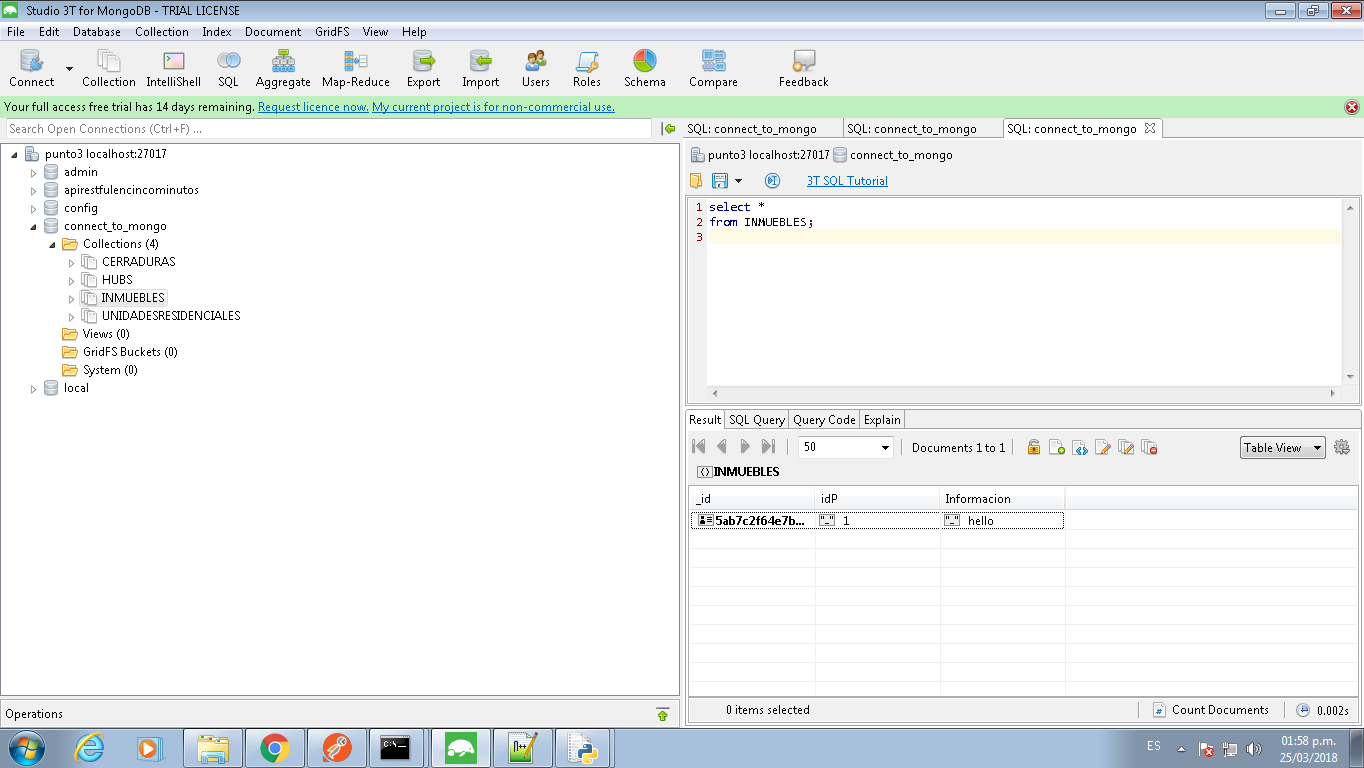


**Servicio PUT:**

<http://127.0.0.1:5000/inmuebles/1>

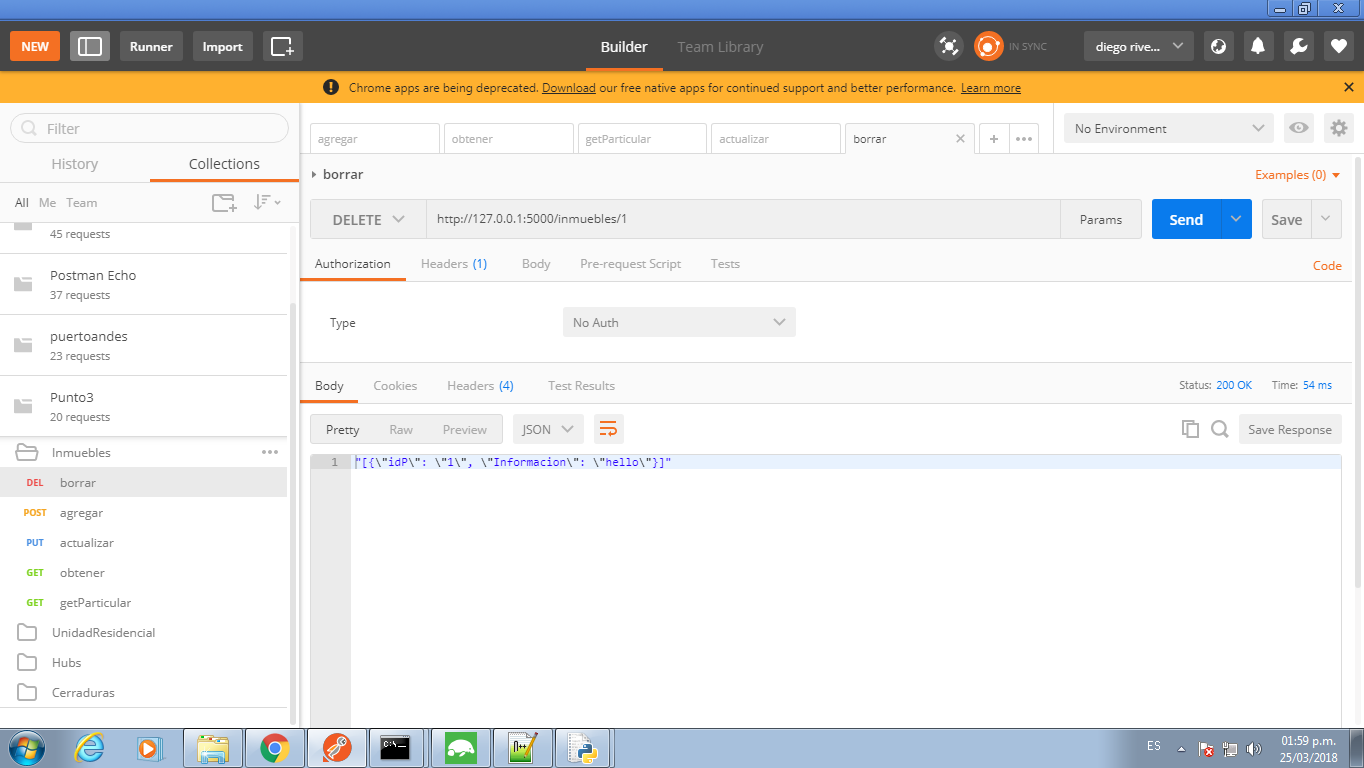


**Base de datos**

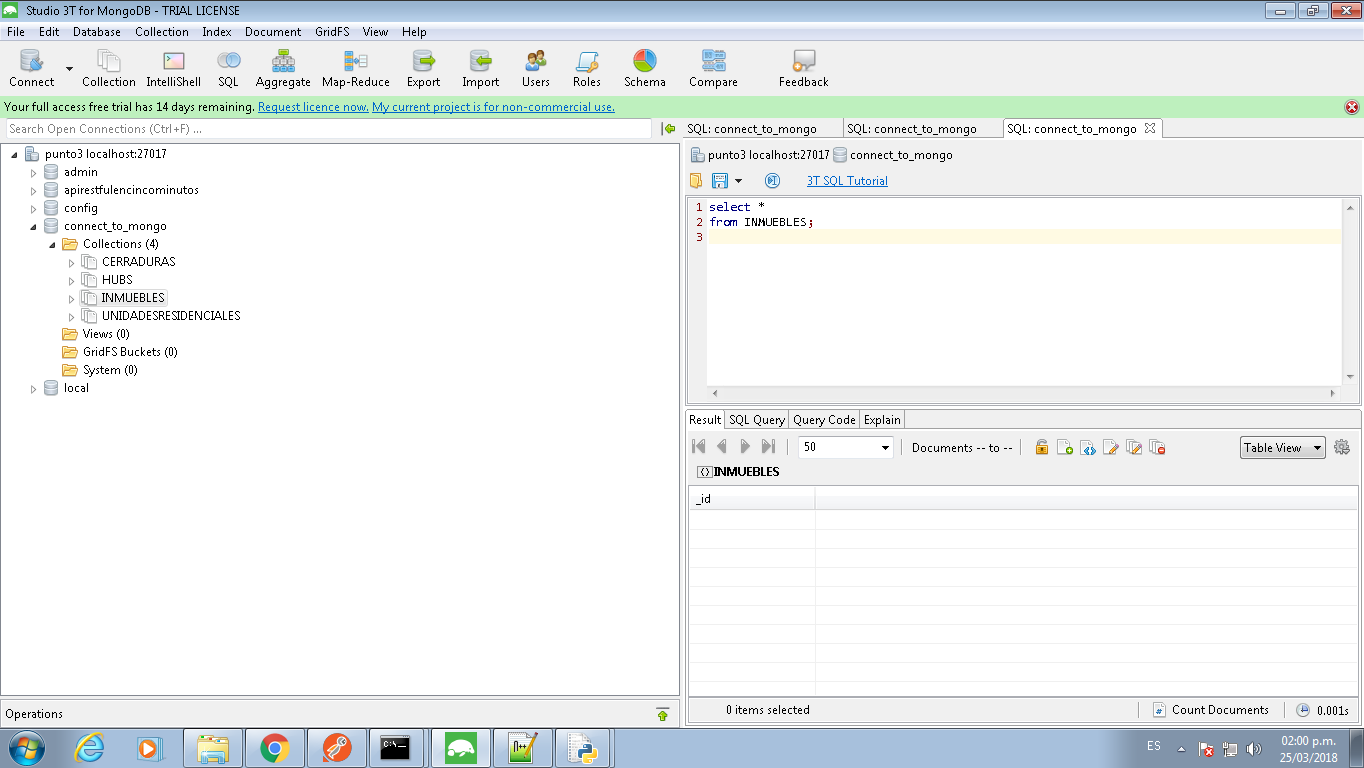


**Servicio DELETE:**

<http://127.0.0.1:5000/inmuebles/1>



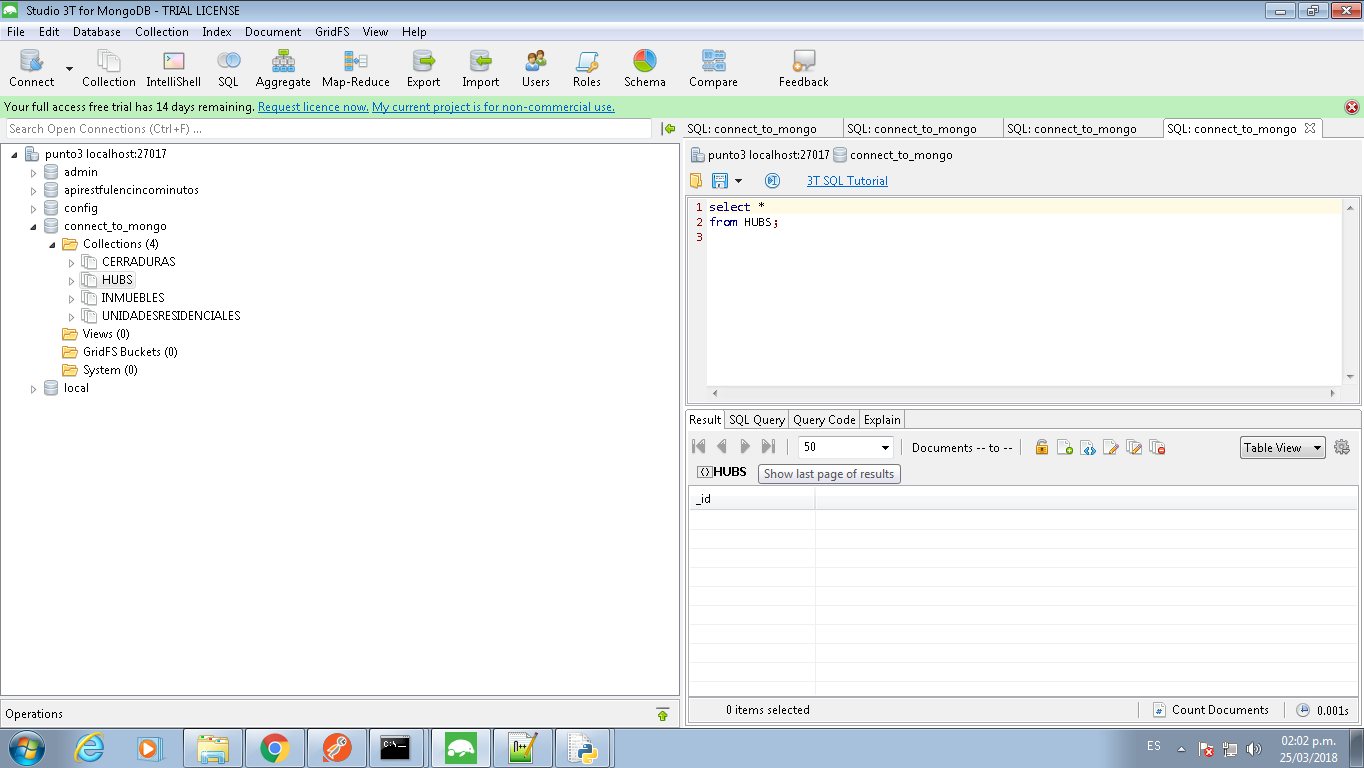
**Base de Datos:**



**HUB**

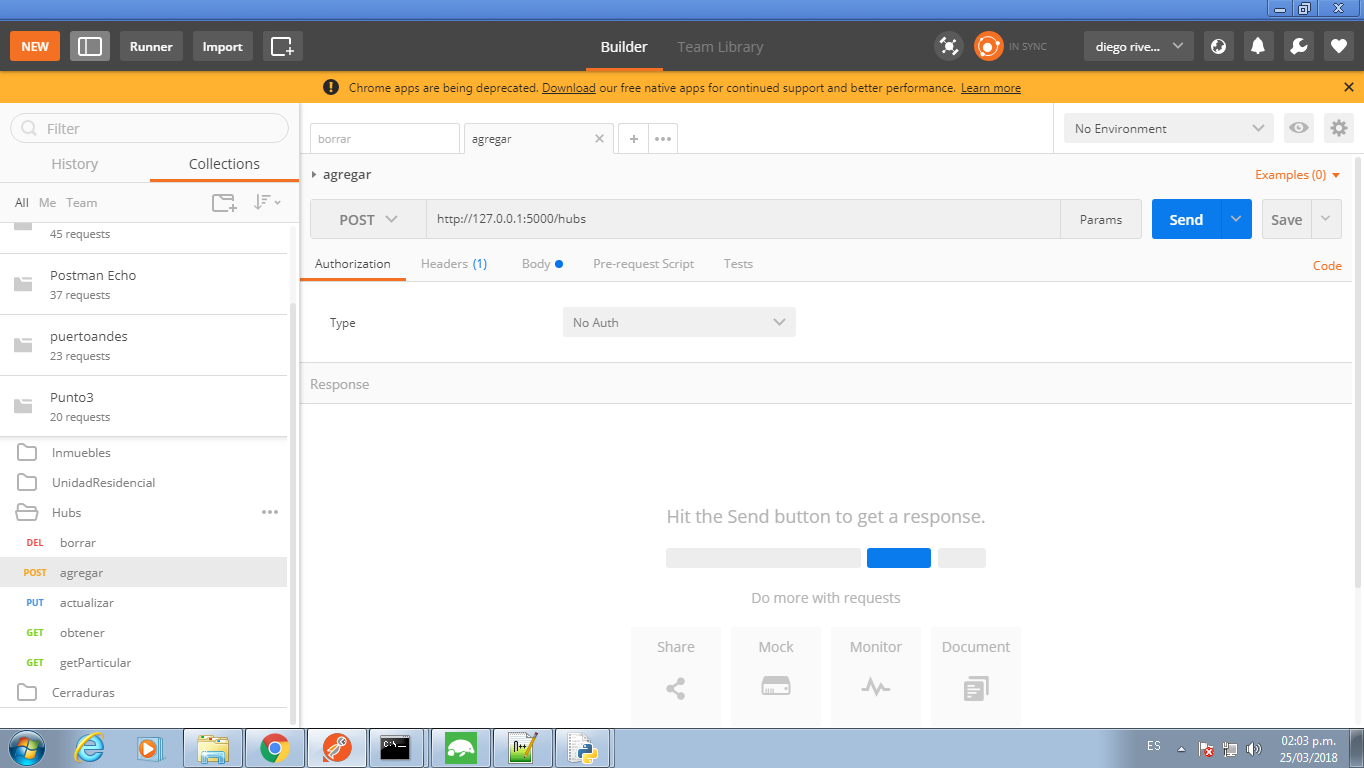
**Servicio POST:**

**Base de datos vacía**

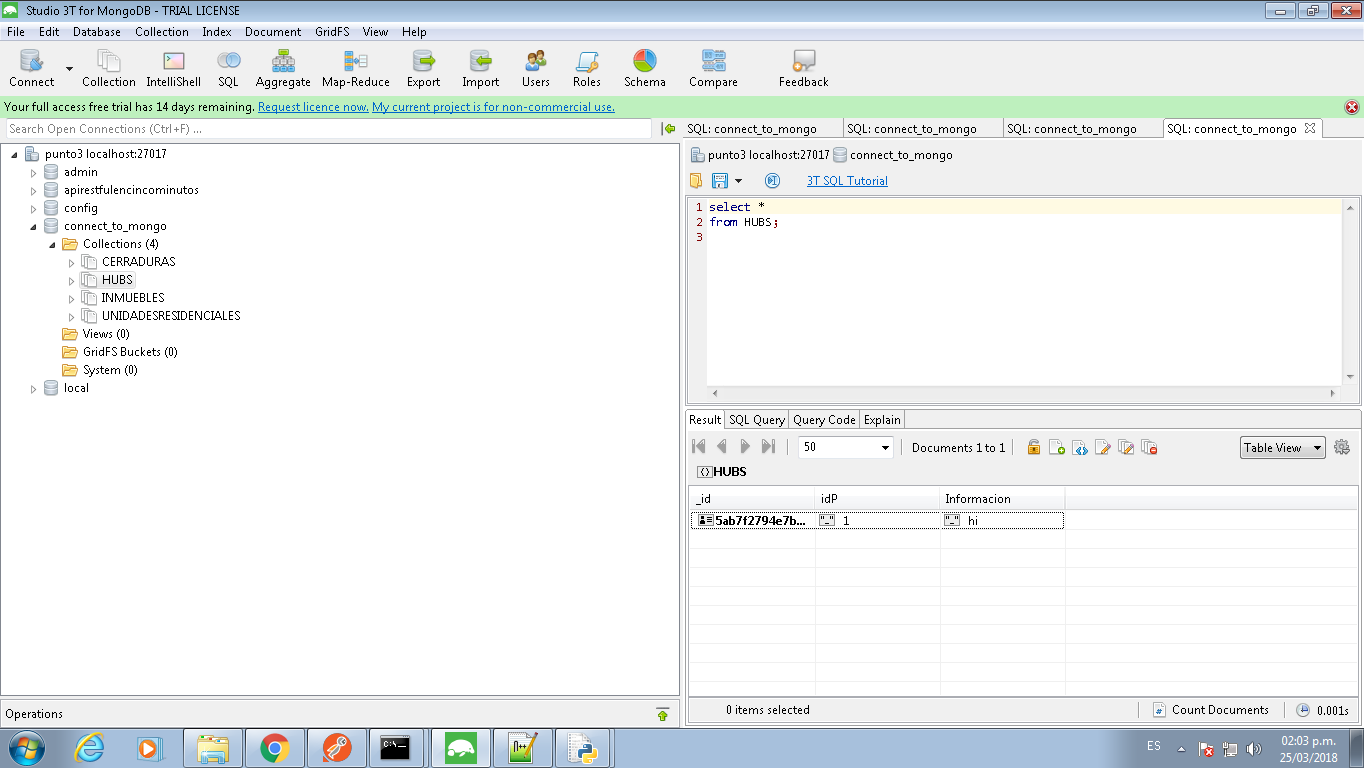


**Postman:**

<http://127.0.0.1:5000/hubs>

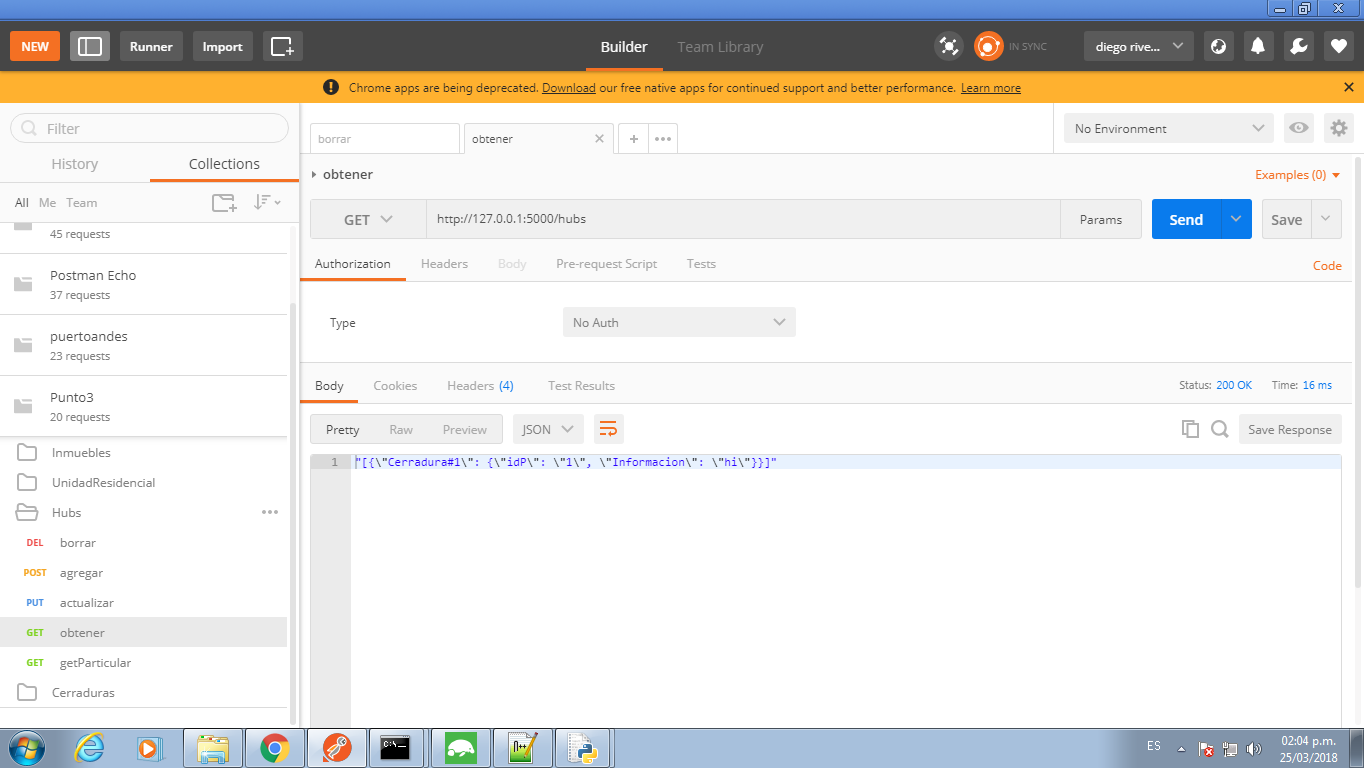


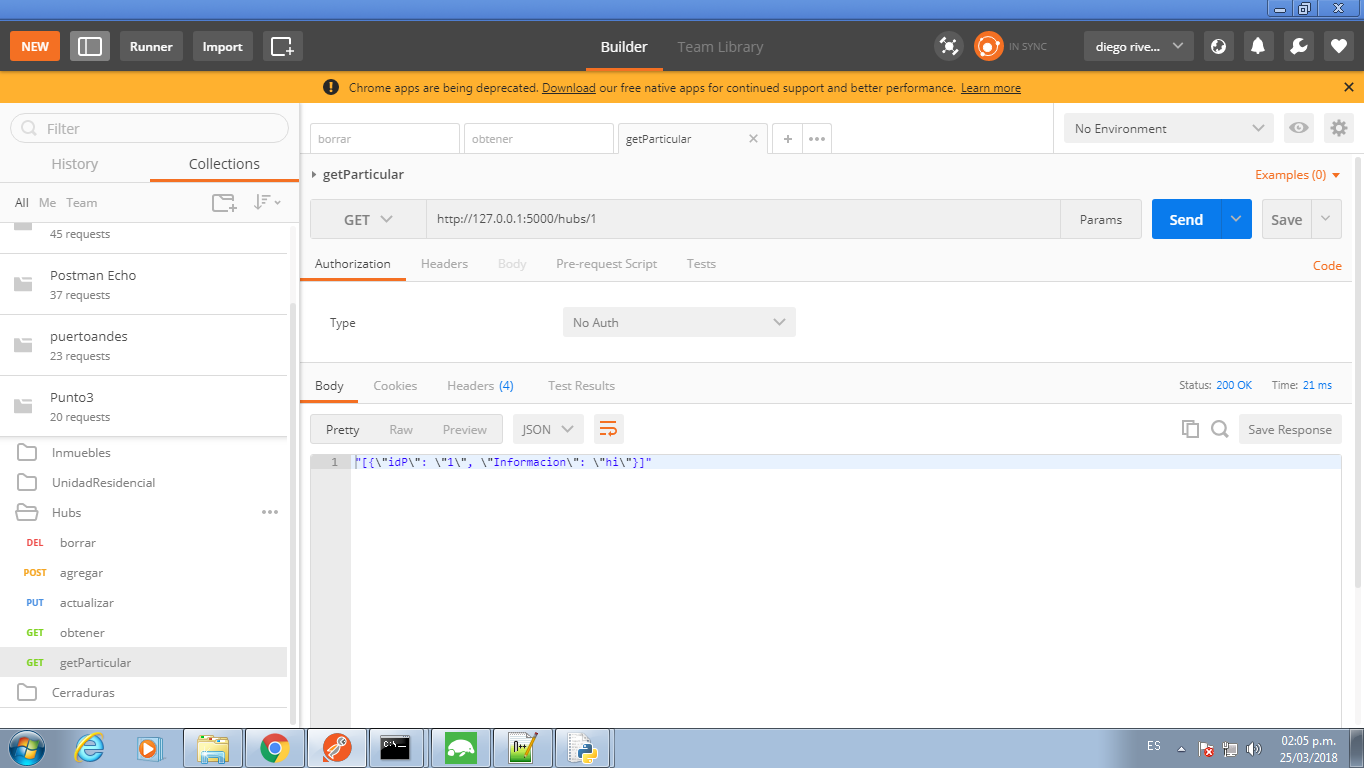
**Base de datos**



**Servicio GETALL:**

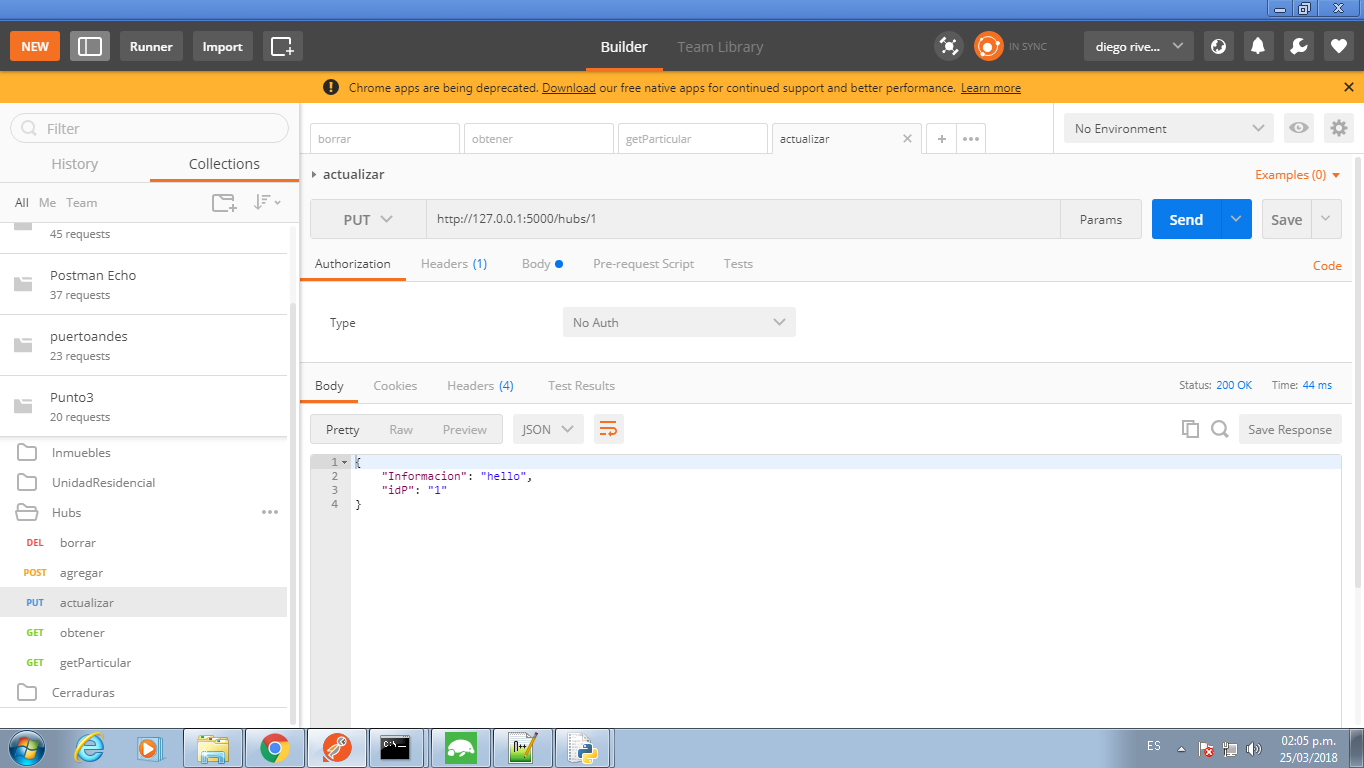
<http://127.0.0.1:5000/hubs>



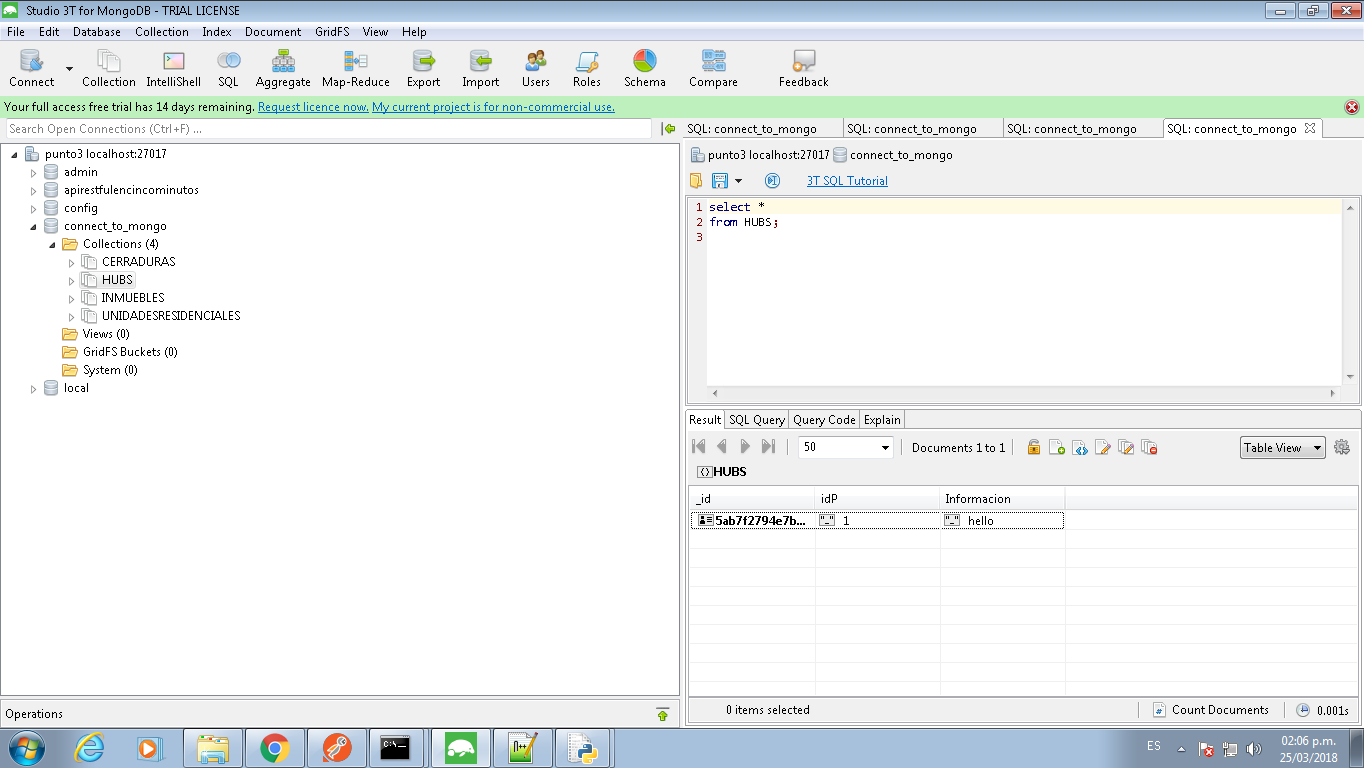
**Servicio GETPARTICULAR:**

**Servicio PUT:**

<http://127.0.0.1:5000/hubs/1>

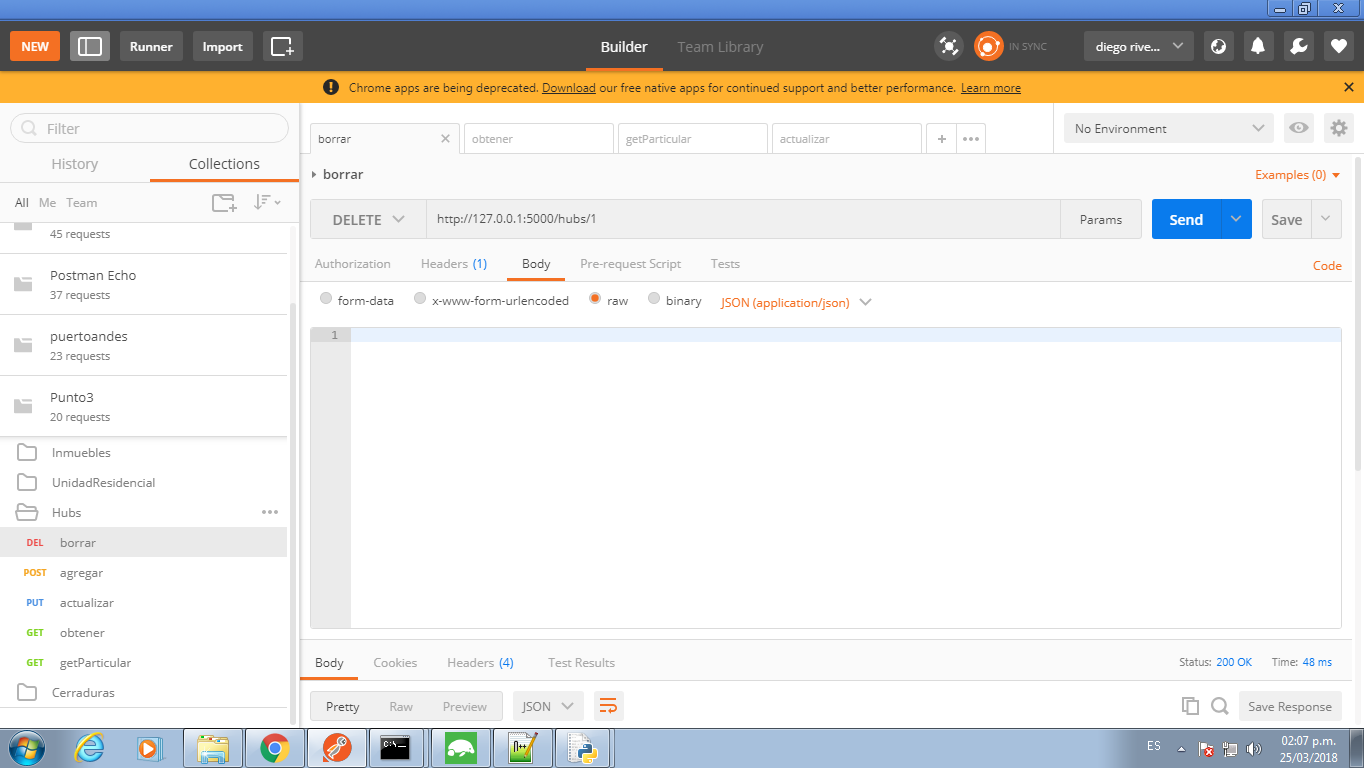


**Base de Datos:**

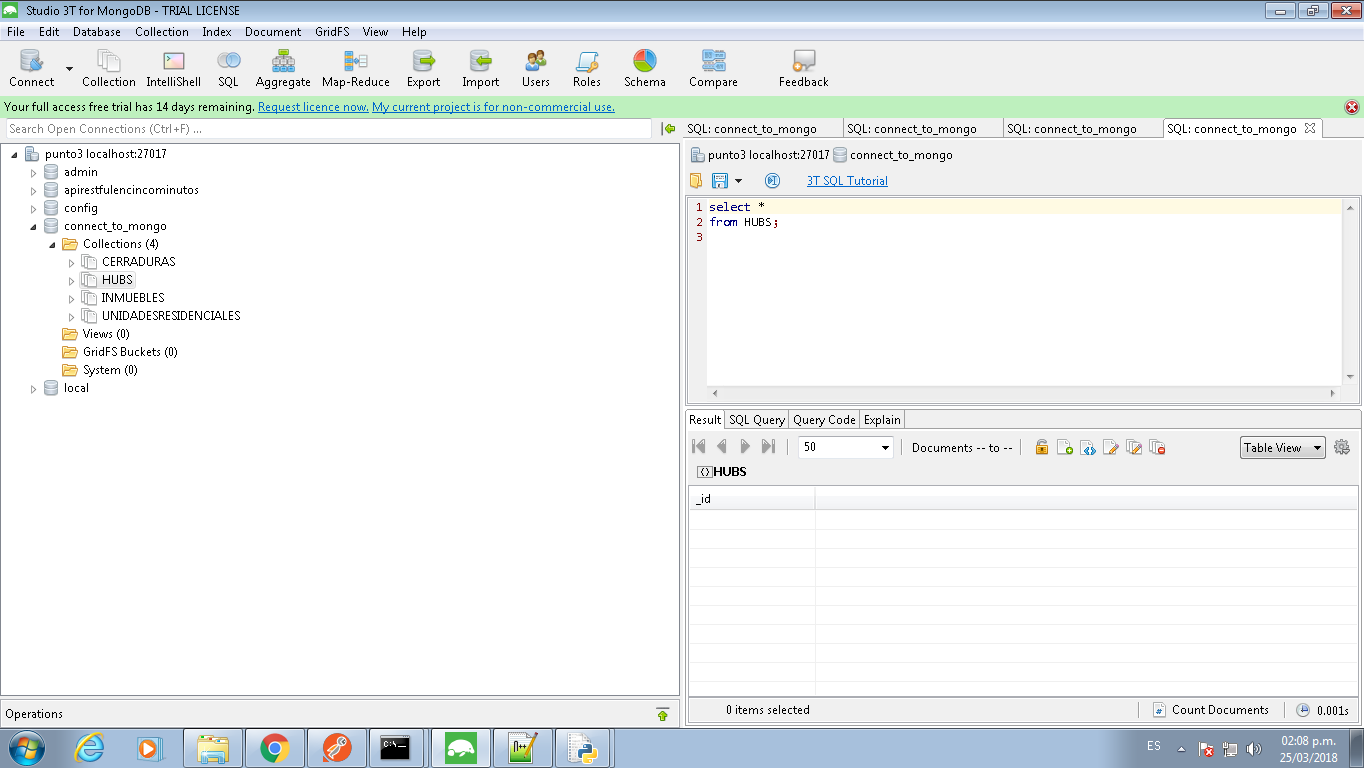


**Servicio DELETE:**

<http://127.0.0.1:5000/hubs/1>



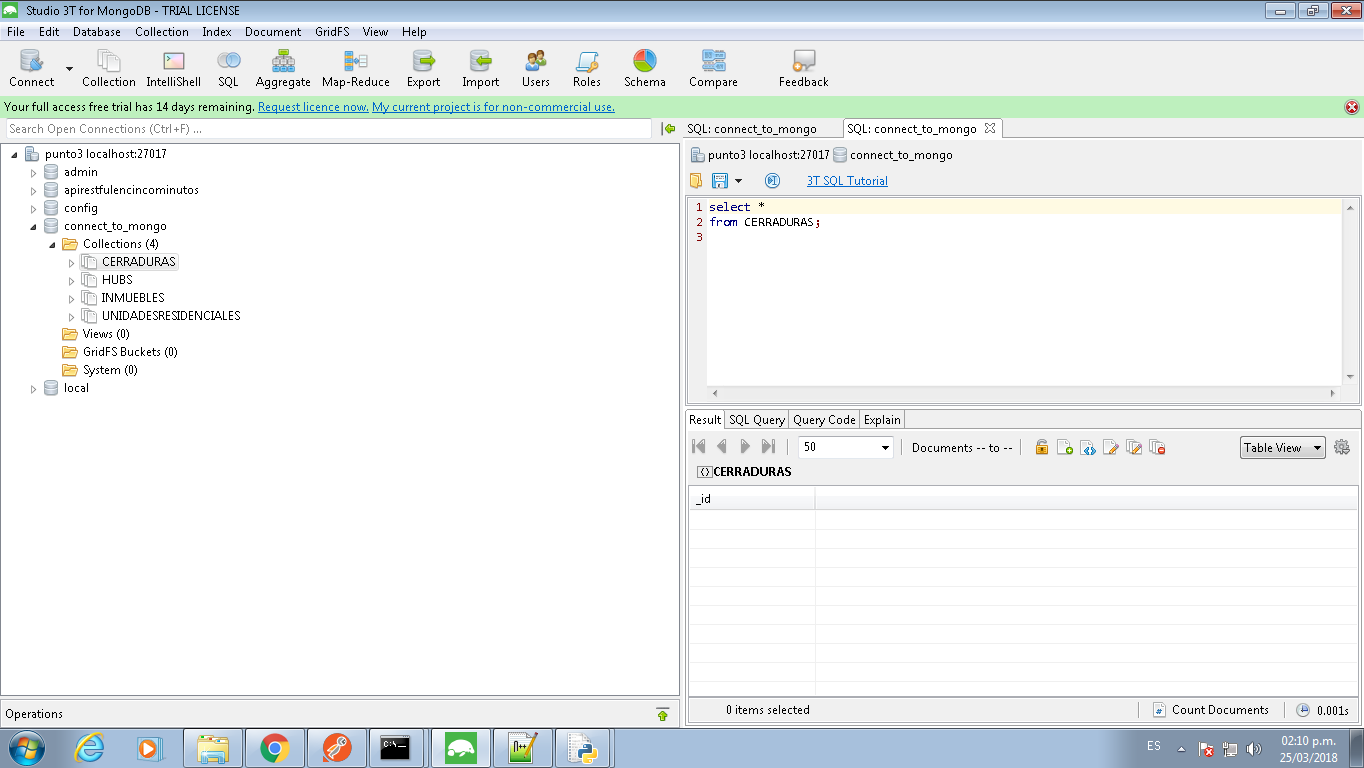
**Base de Datos:**



**CERRADURA**

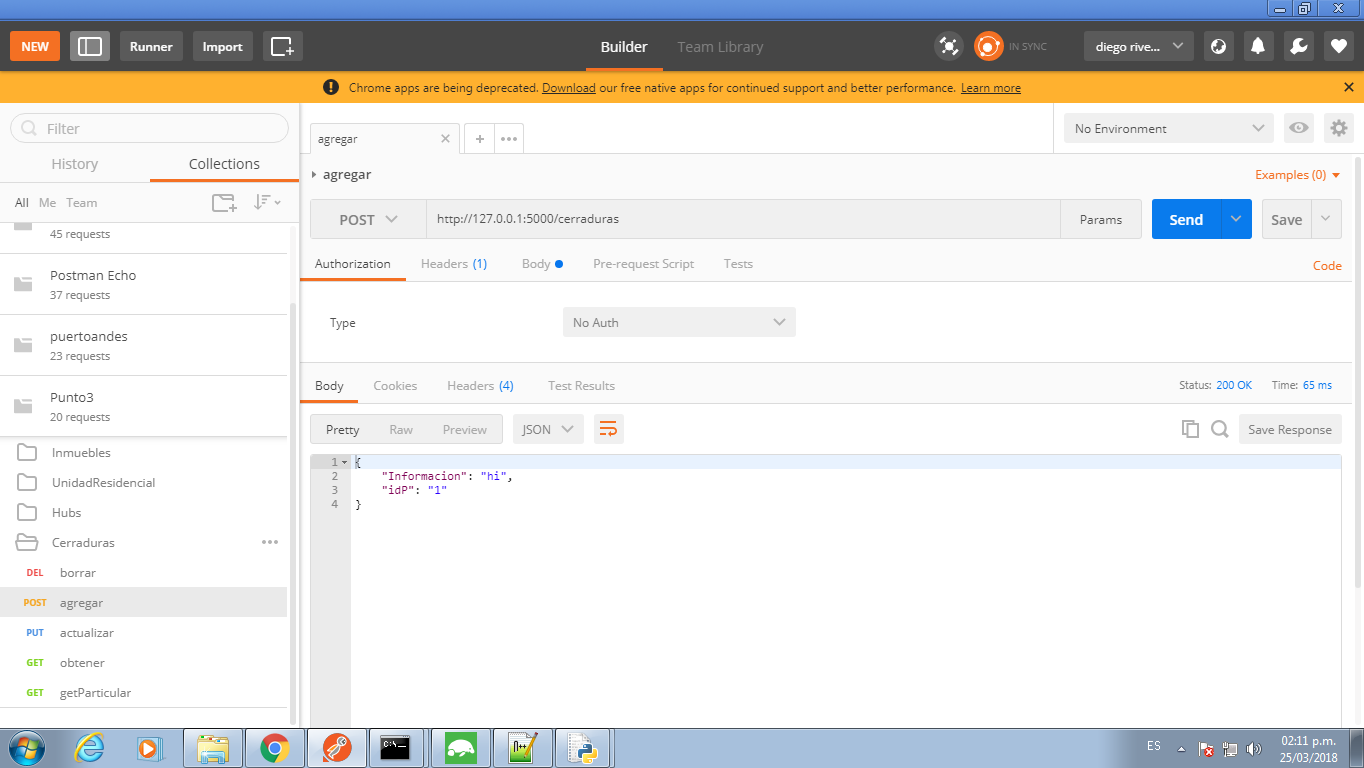
**Servicio POST**

**Base de datos Vacía**

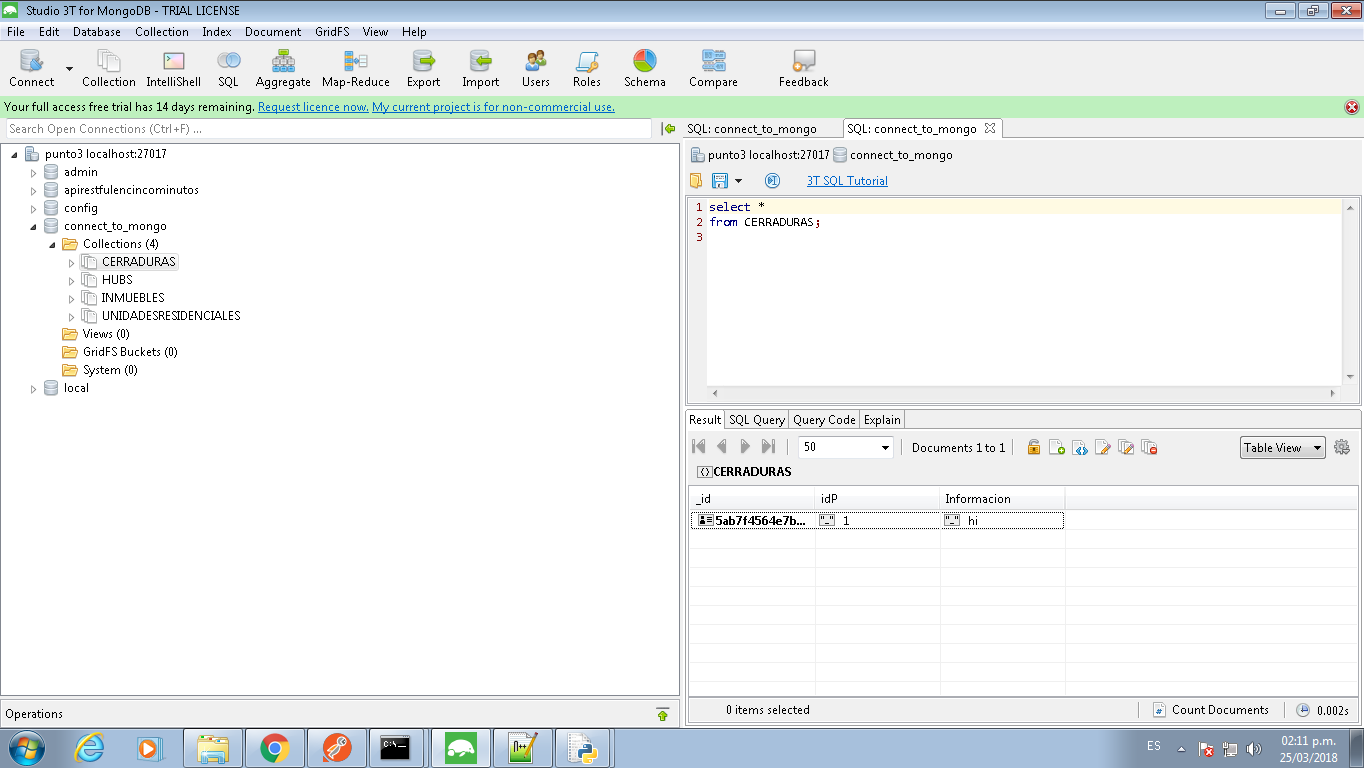


**Postman:**

<http://127.0.0.1:5000/cerraduras>

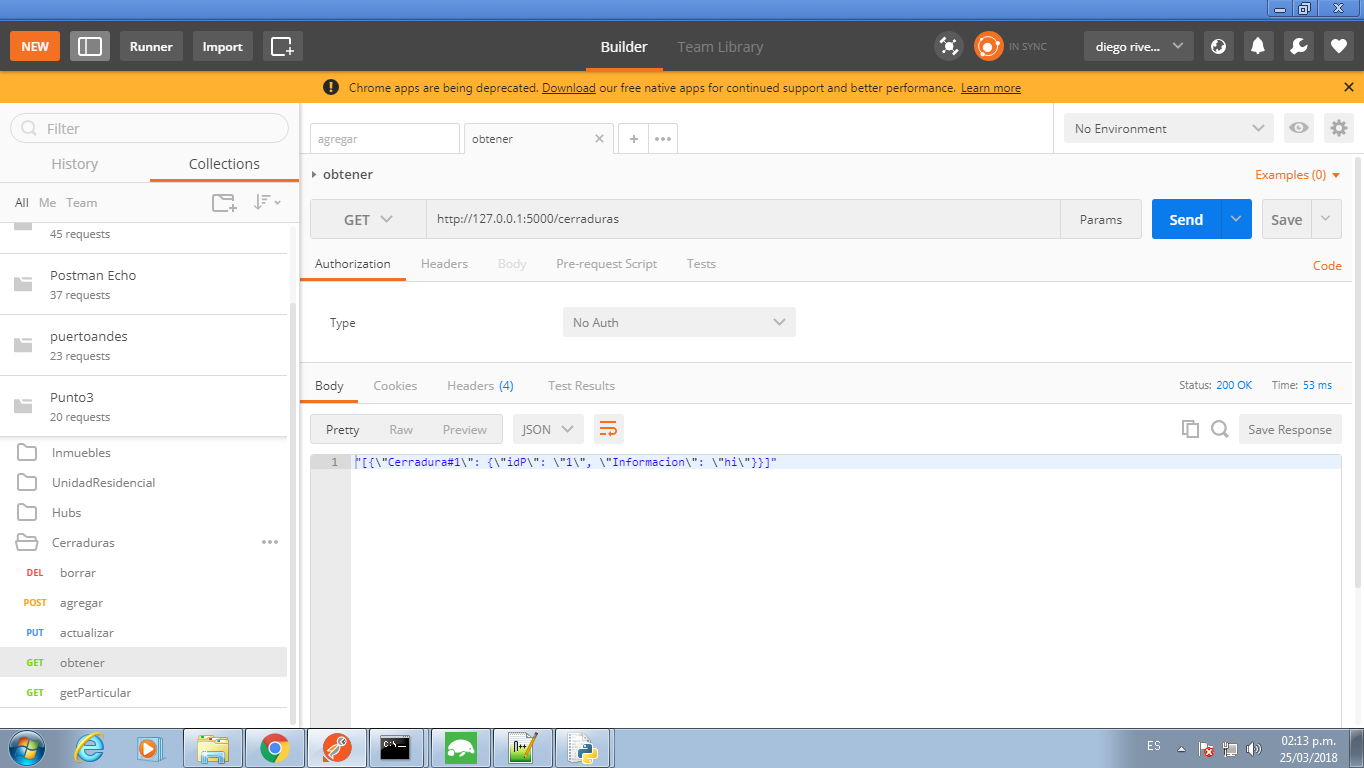


**Base de Datos:**



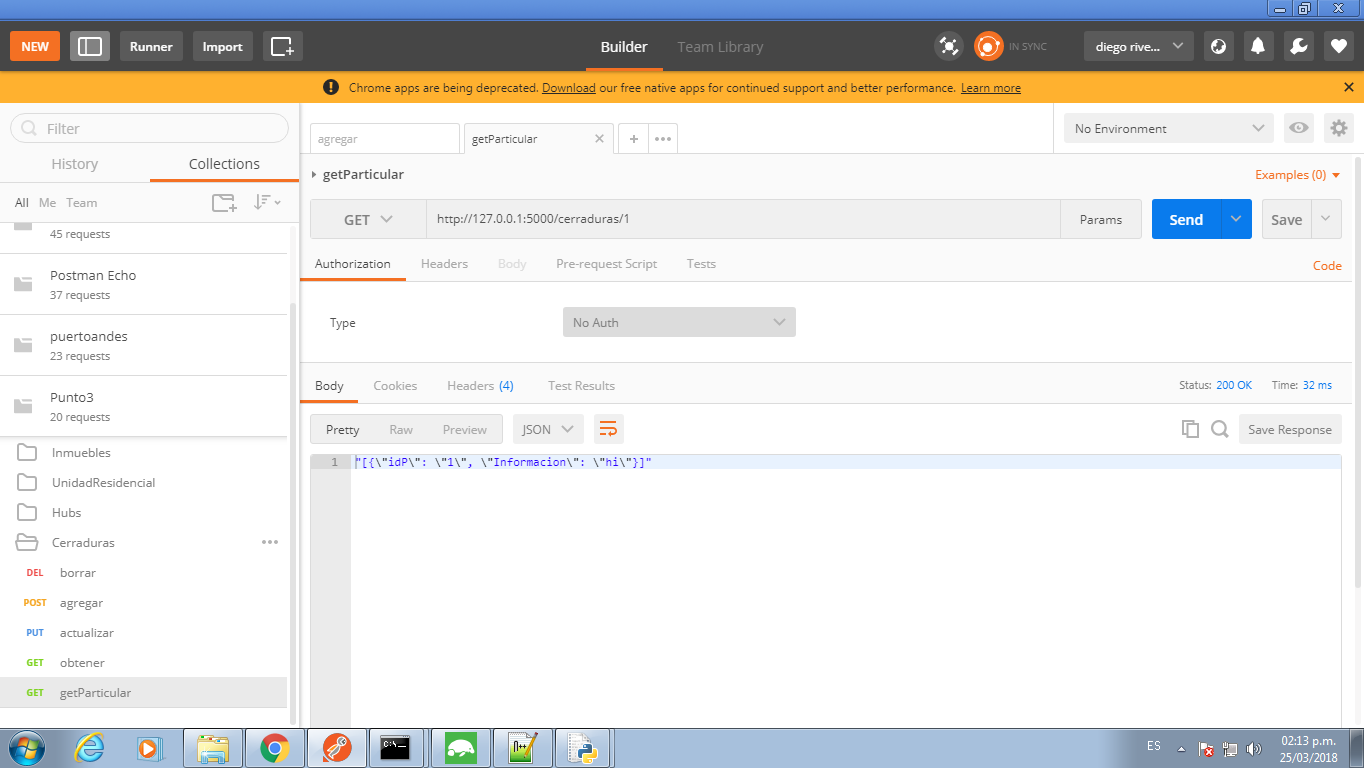
**Servicio GETALL:**

<http://127.0.0.1:5000/cerraduras>



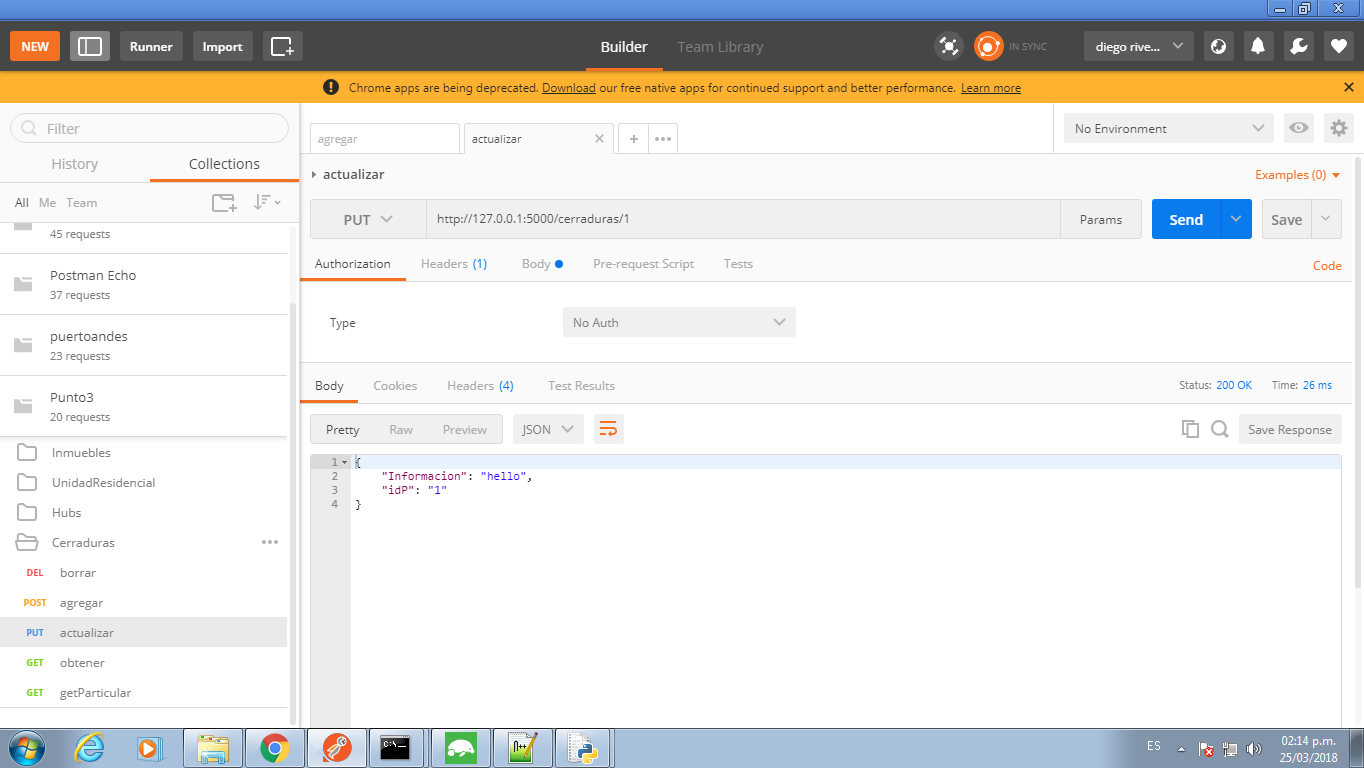
**Servicio GETPARTICULAR:**

<http://127.0.0.1:5000/cerraduras/1>

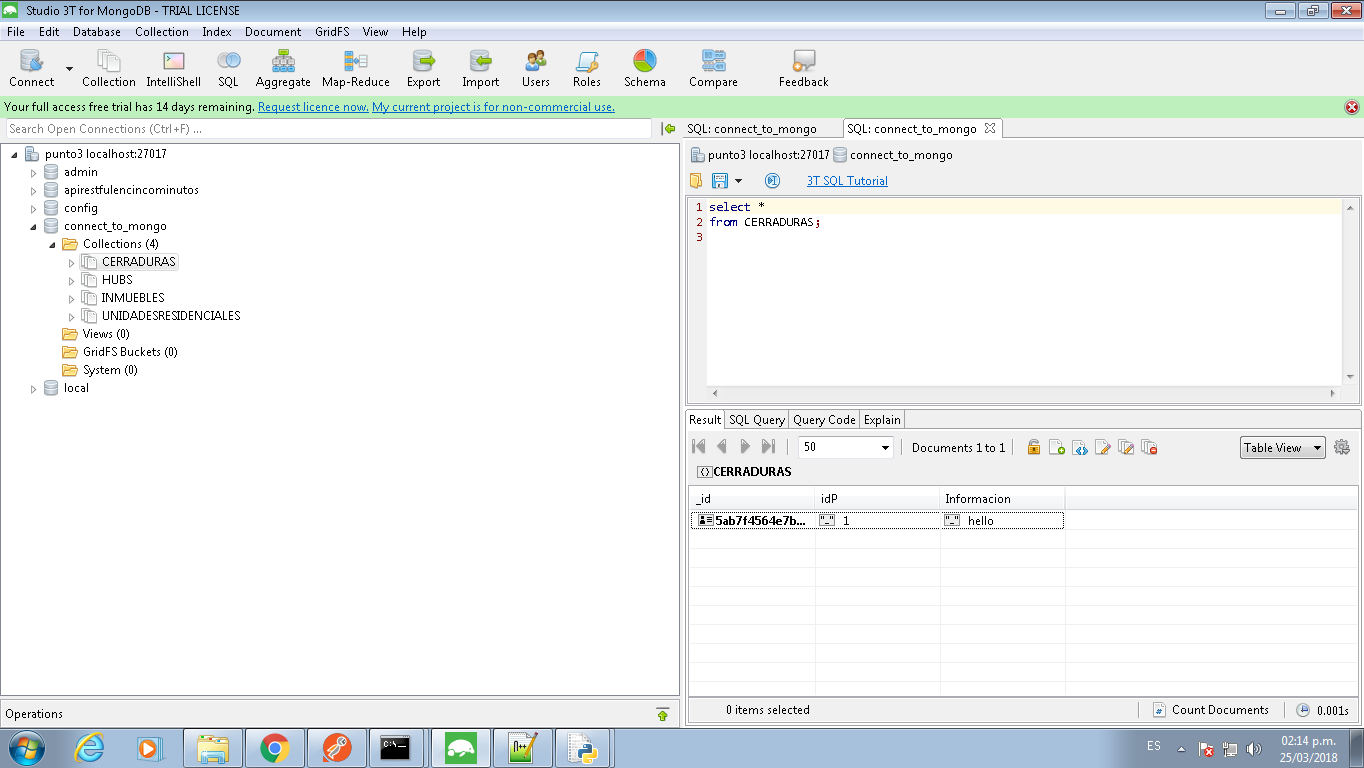


**Servicio PUT:**

<http://127.0.0.1:5000/cerraduras/1>

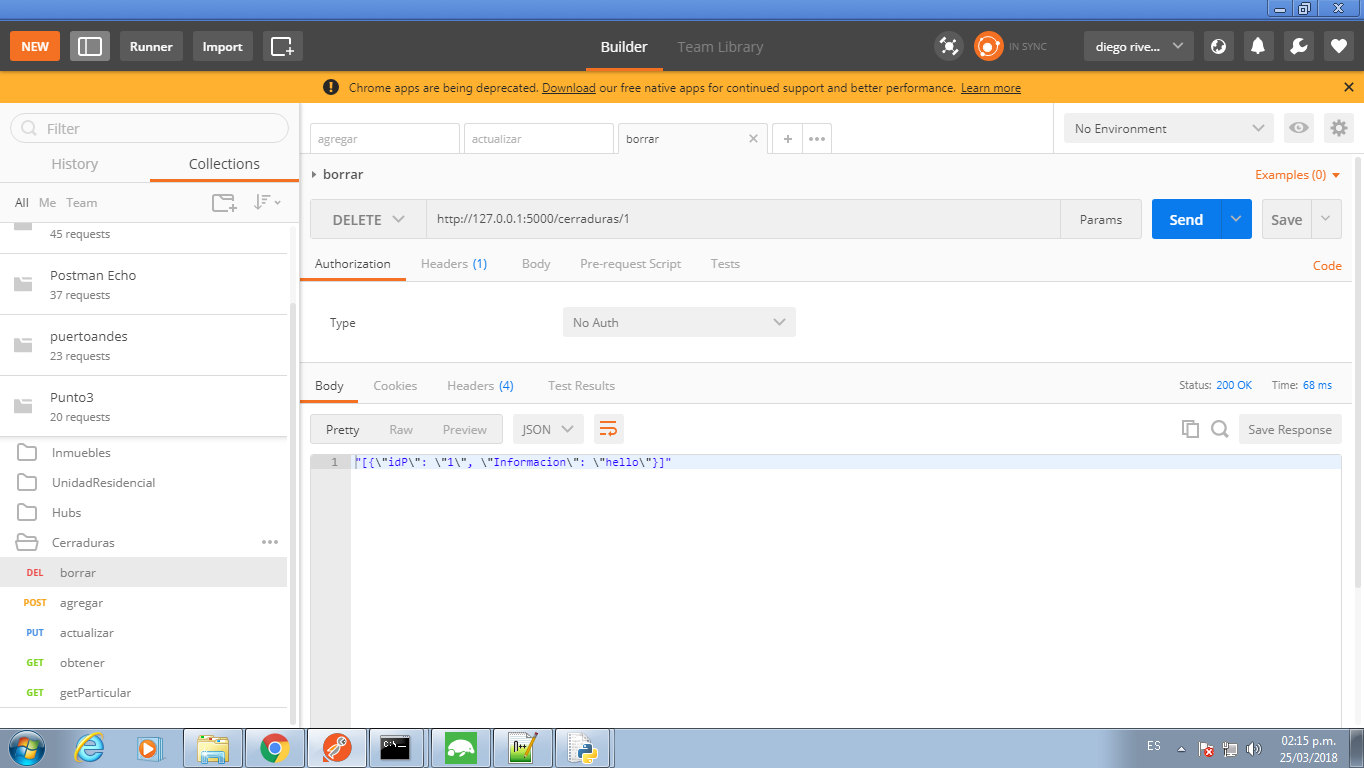


**Base de Datos:**

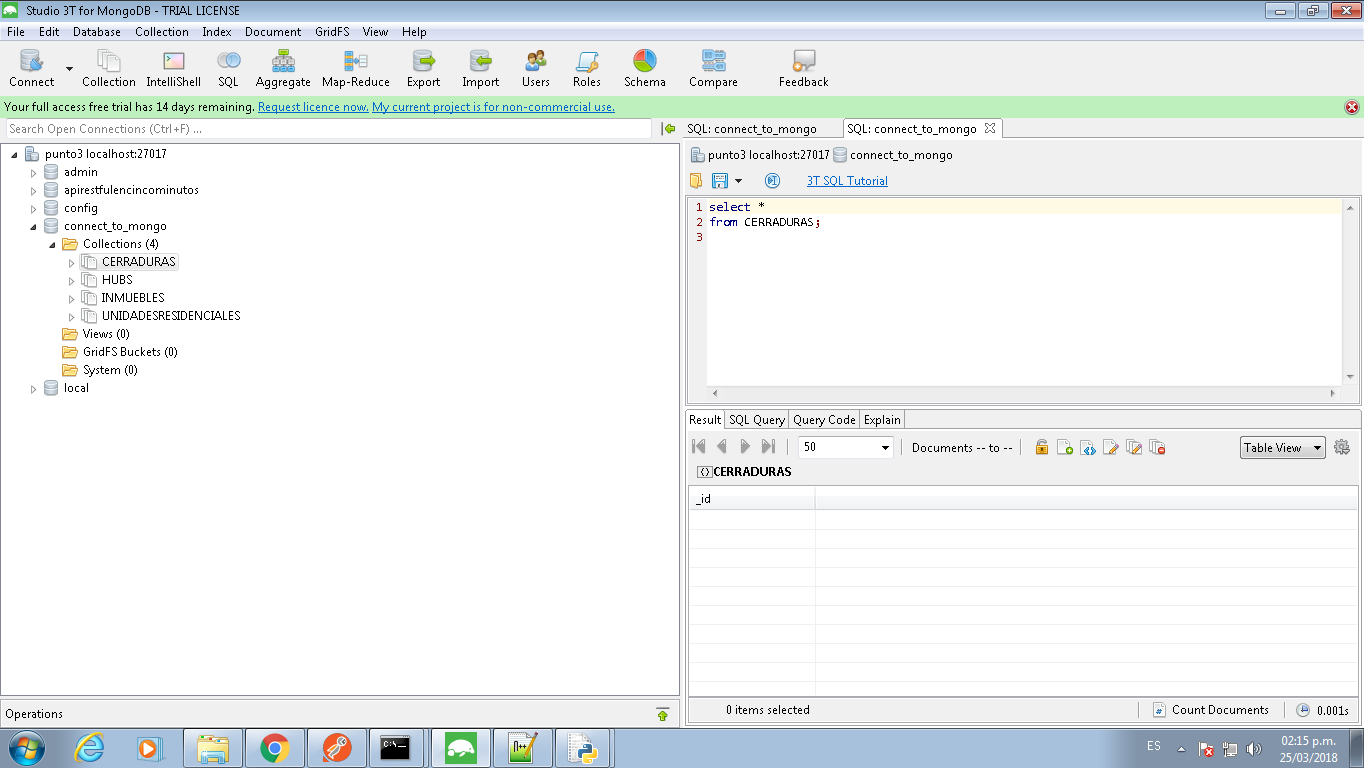


**Servicio DELETE:**

<http://127.0.0.1:5000/cerraduras/1>



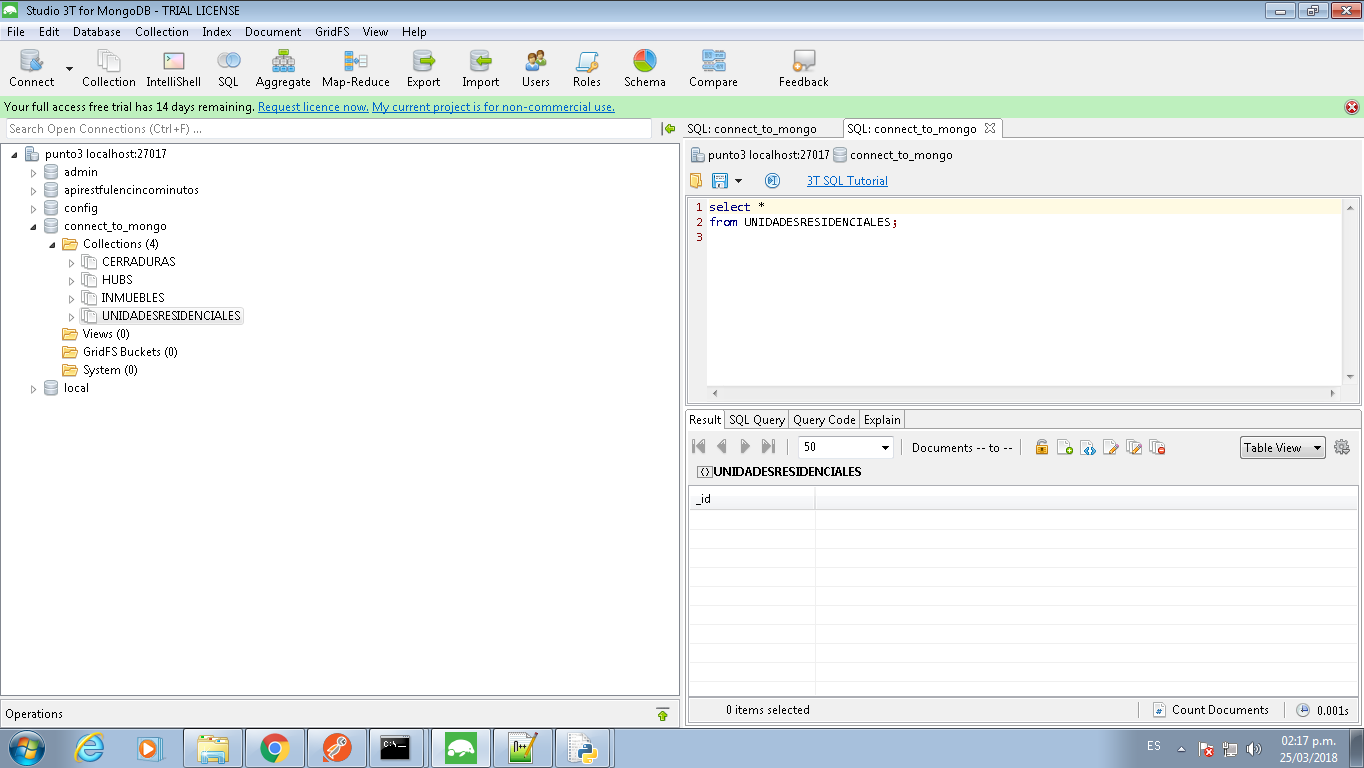
**Base de Datos**



**UNIDADESRESIDENCIALES**

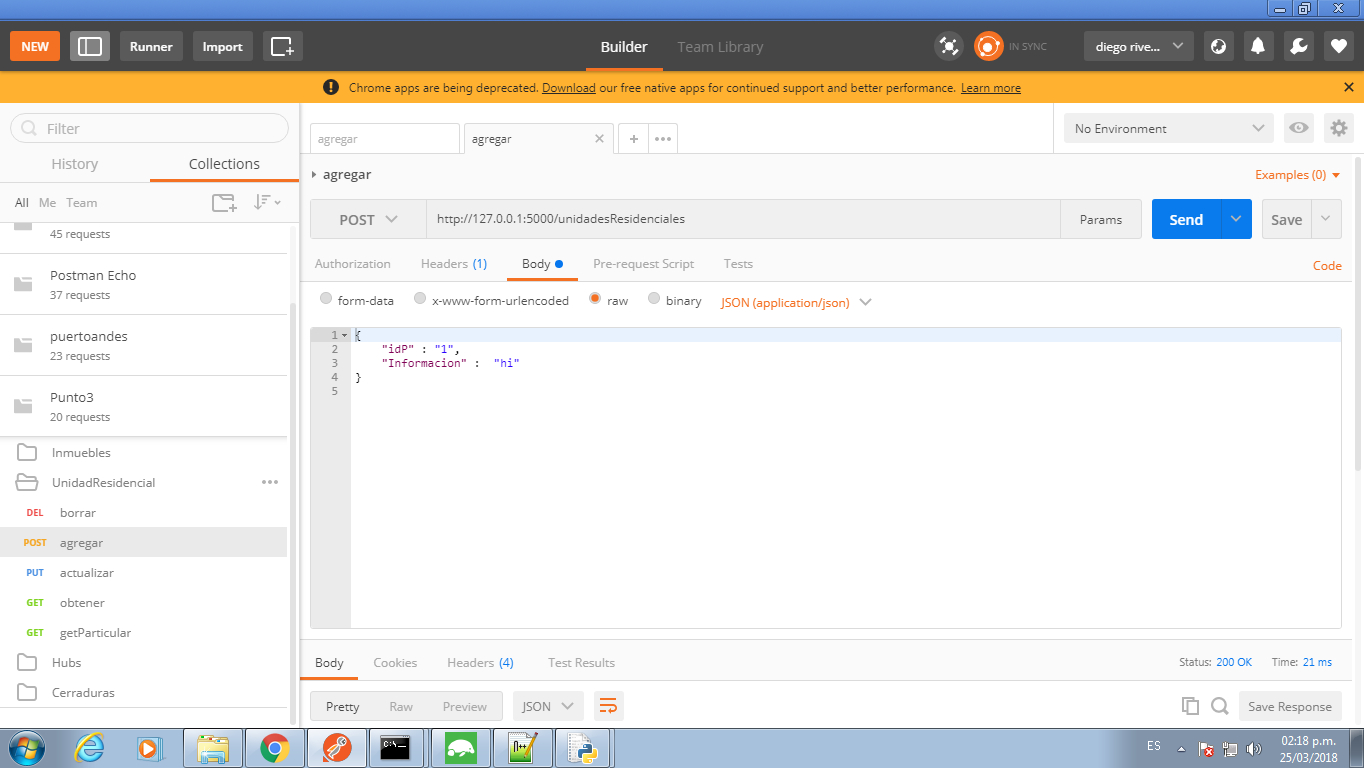
**Servicio POST:**

**Base de Datos Vacía:**

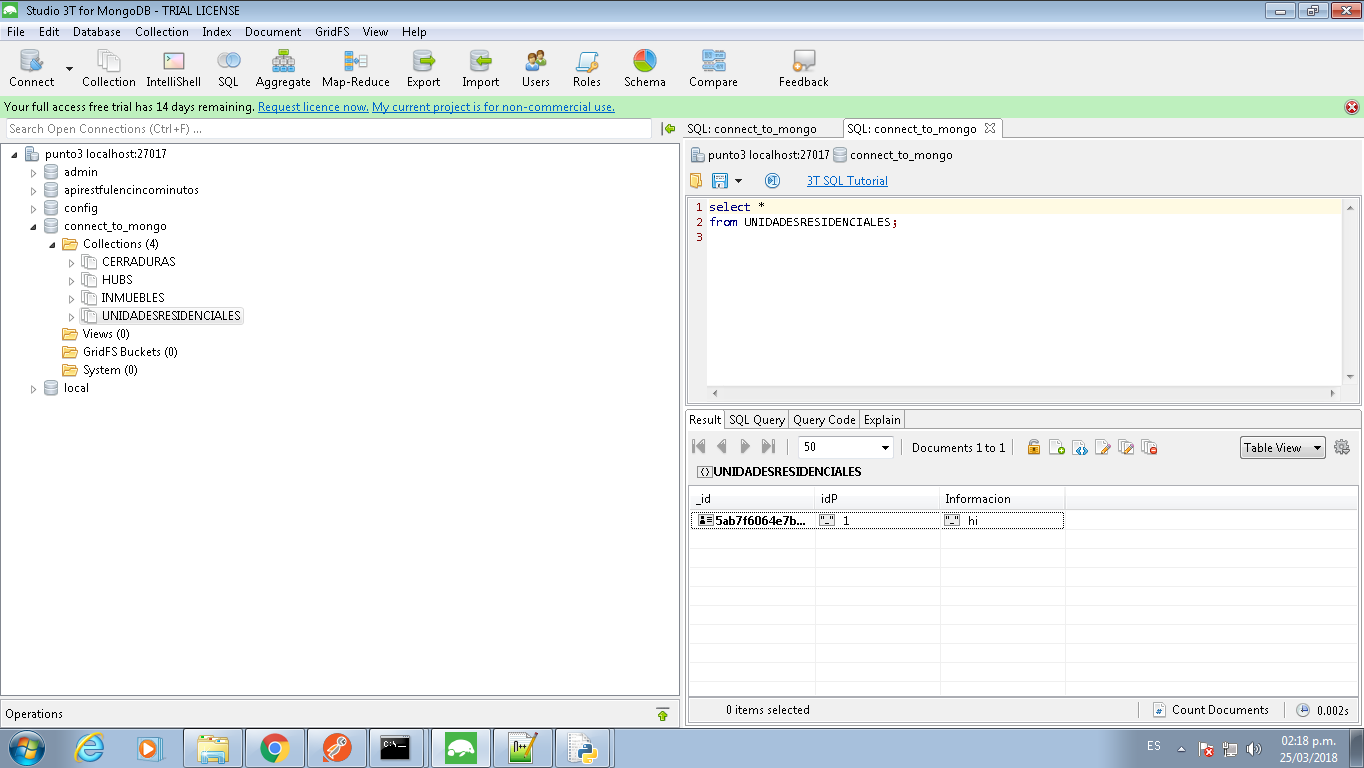


**Postman:**

<http://127.0.0.1:5000/unidadesResidenciales>

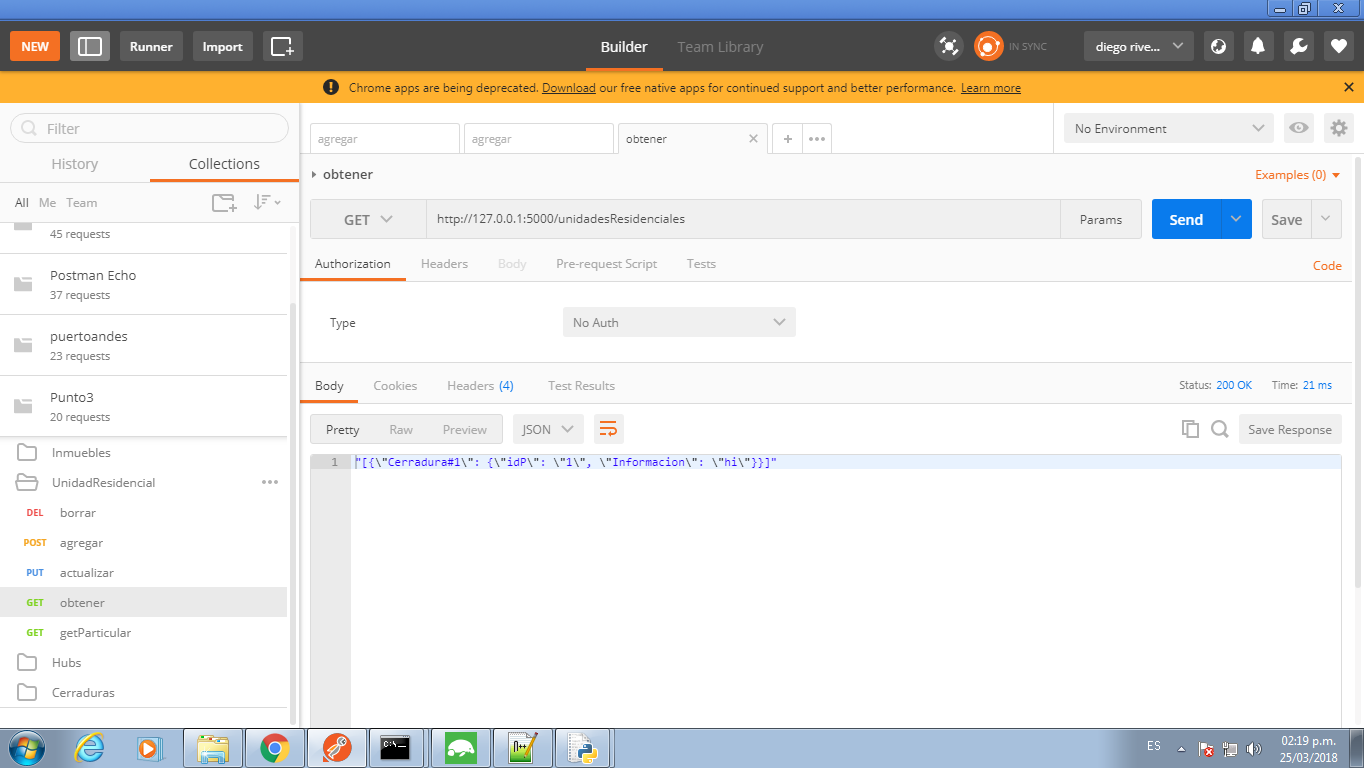


**Base de Datos:**

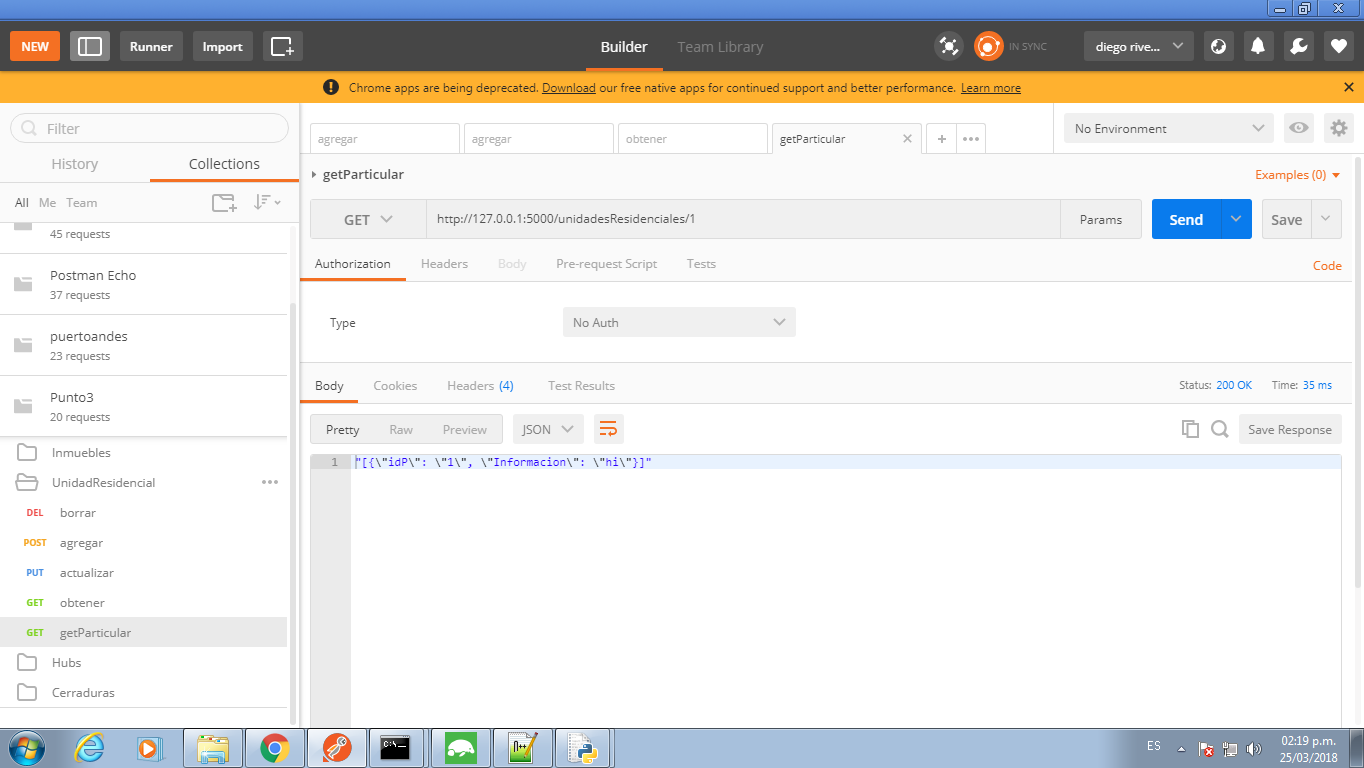


**Servicio GETALL:**

<http://127.0.0.1:5000/unidadesResidenciales>

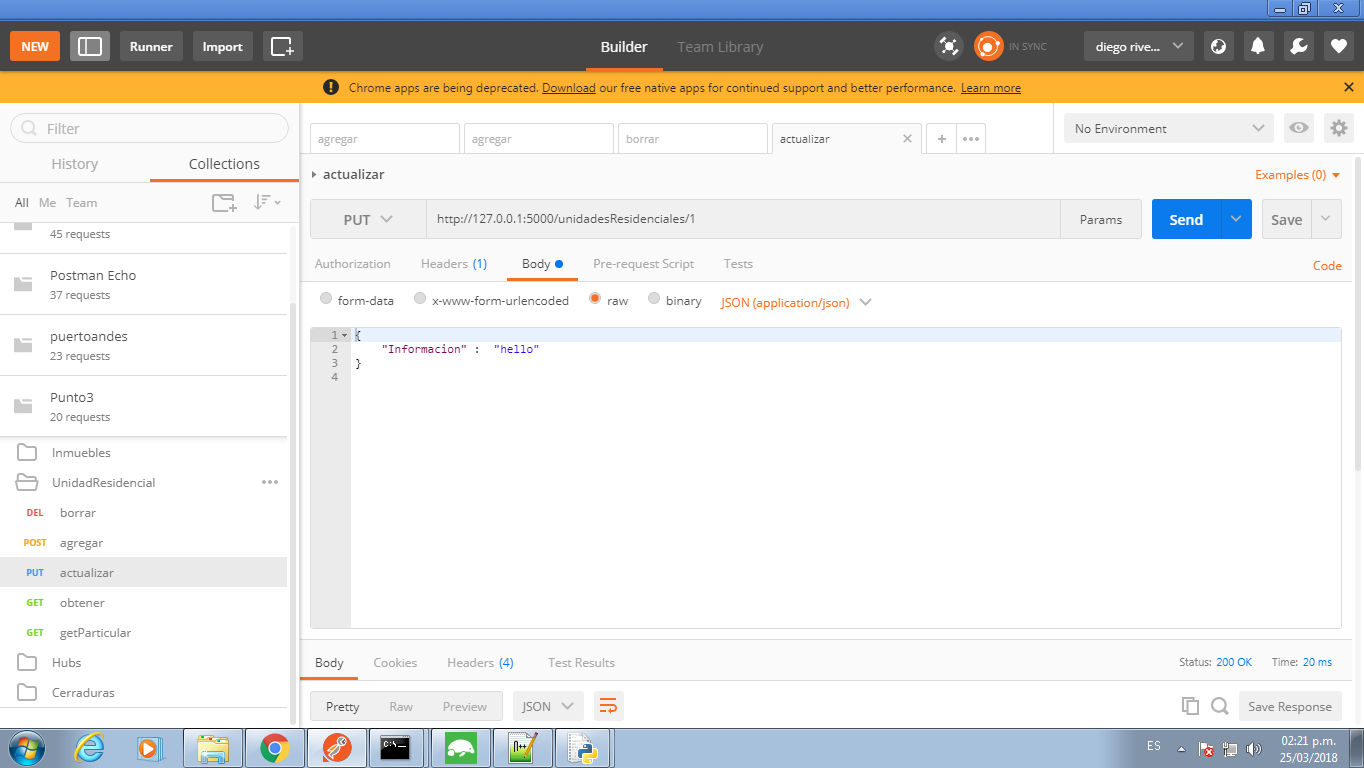


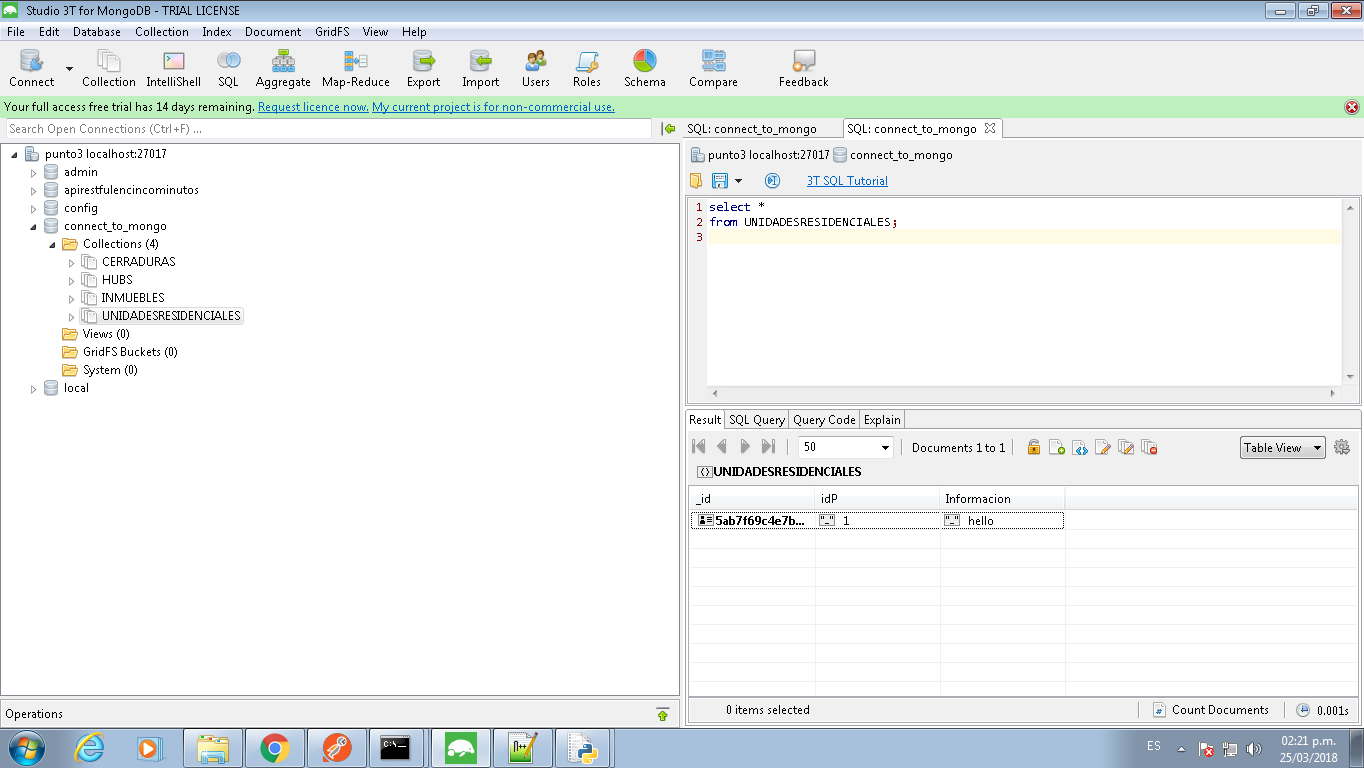
**Servicio GETPARTICULAR:**



**Servicio PUT:**

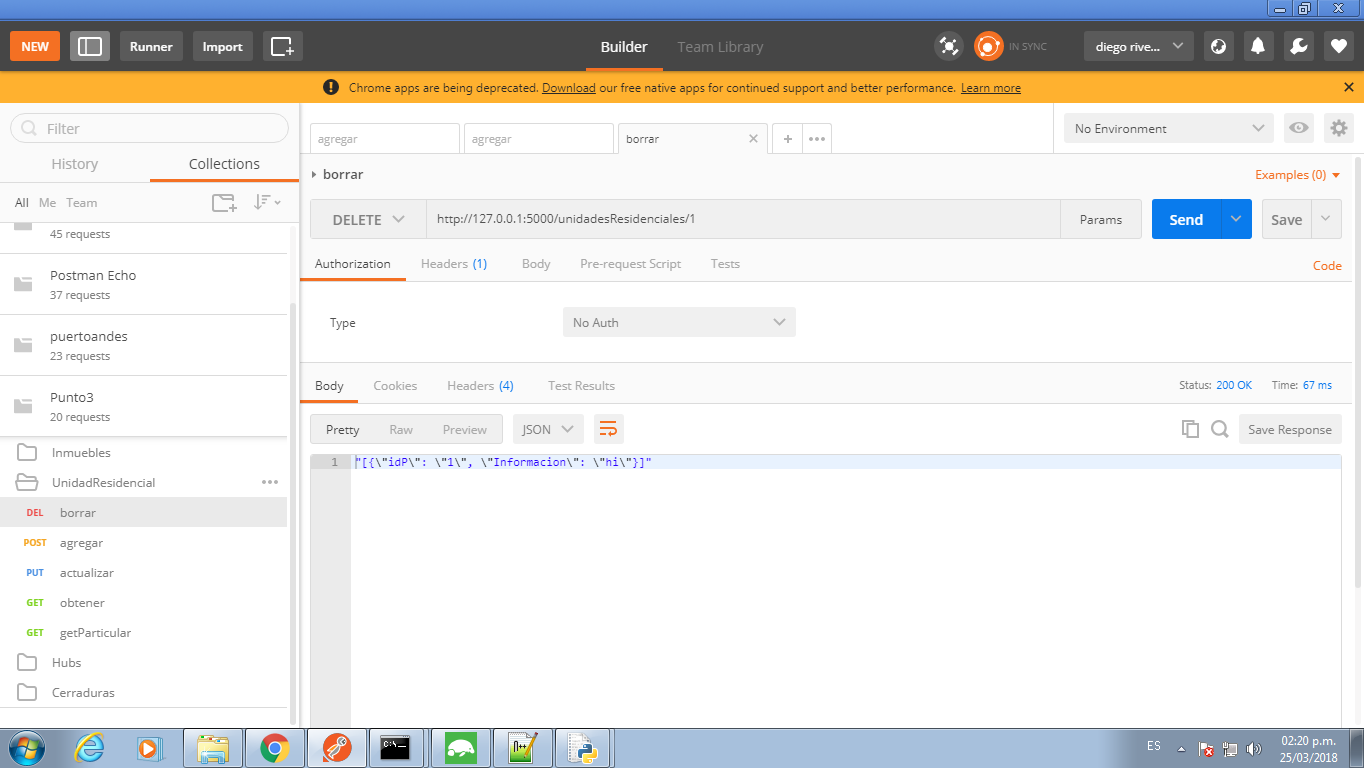
<http://127.0.0.1:5000/unidadesResidenciales/1>



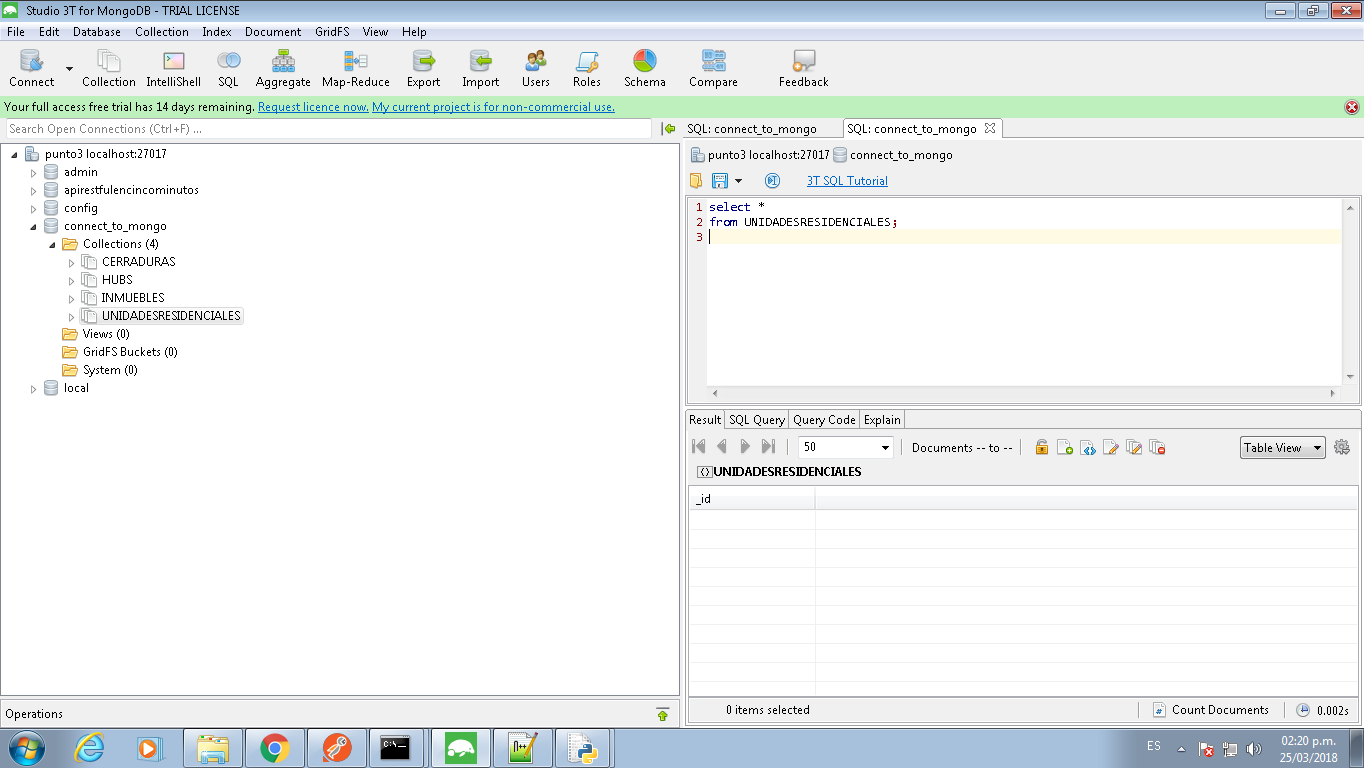
**Base de Datos**

**Servicio DELETE:**

<http://127.0.0.1:5000/unidadesResidenciales/1>



**Base de Datos**



## **JMeter**

Para probar las 300.000 alertas en un minuto con tiempos de respuesta de 1 s y 0% de error, se configuraron las pruebas de carga con los siguientes parámetros:

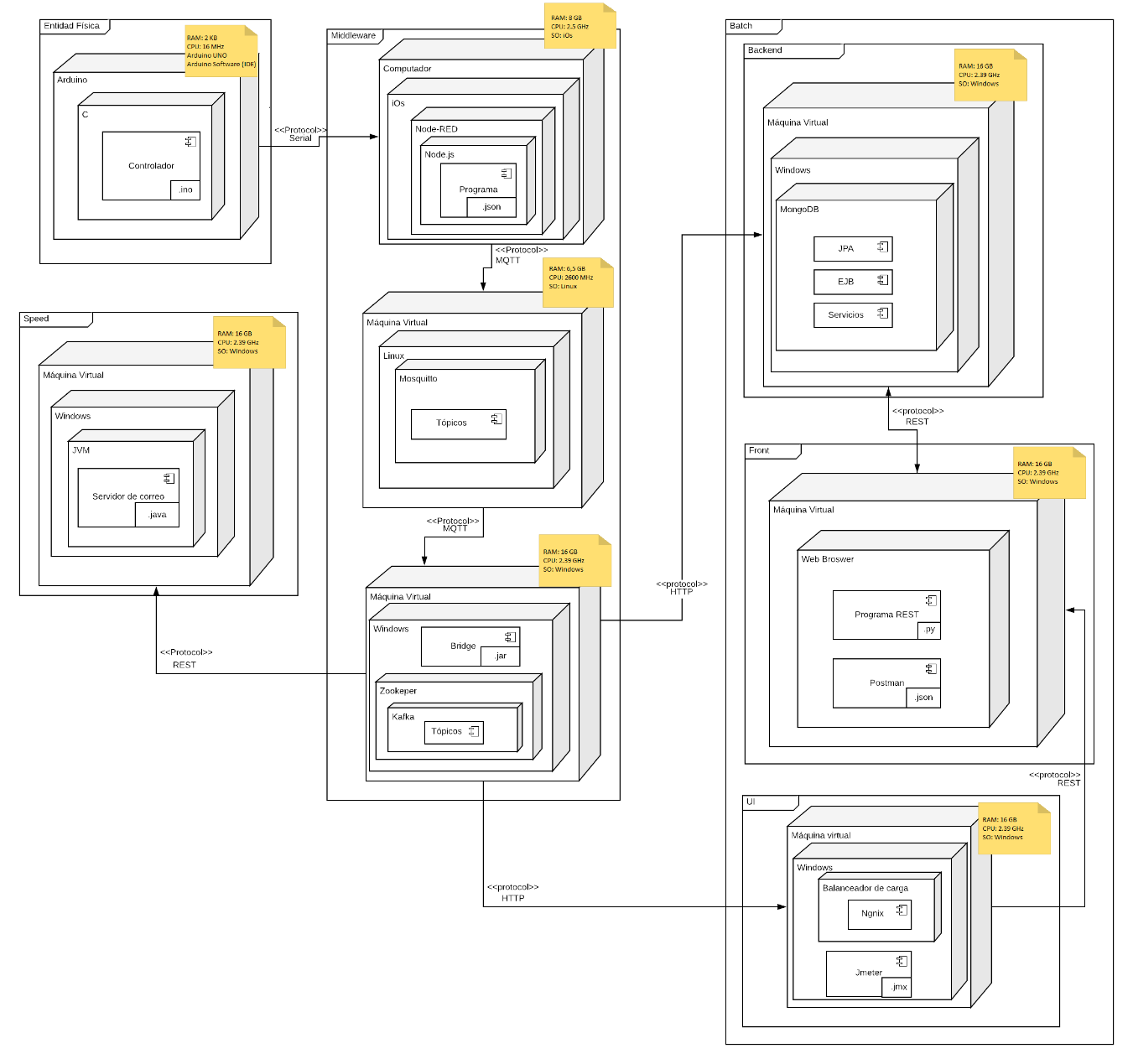
* Número hilos: 300.000
* Ramp-up: 0
* Iteraciones: 1

Antes de estas pruebas, se empezaron con un nùmero menor de threads desde 200 hasta 300.000, así:

100, 500, 1000, 5000, 10.000, 50.000, 100.000, 300.000

Adjunto al documento, se encuentra un archivo .jmx del escenario de prueba utilizado para pruebas de carga.

# **Diagrama de Despliegue**



Adjunto al documento, se encuentra la imagen en formato PNG para una mejor visualización.

# **Comparación de resultados**

Resultados en árbol:



Resultados

100



500



1000



5000



10.000



50.000



100.000



300.000



Resumen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hilos | Latencia | Minimo | Máximo | % error |
| 100 | 60 | 6 | 138 | 0 |
| 500 | 92 | 0 | 584 | 0 |
| 1.000 | 99 | 0 | 594 | 0 |
| 5.000 | 95 | 0 | 1401 | 0 |
| 10.000 | 8 | 0 | 273 | 0 |
| 50.000 | 0 | 0 | 152 | 0 |
| 100.000 | 0 | 0 | 210 | 0 |
| 300.000 | 5 | 0 | 1897 | 0 |

Las pruebas de carga muestran las 300.000 alertas en un minuto con tiempos de respuesta máximos mayores a 1 s pero todavía con 0% de error. Así que en este momento, la arquitectura funciona para los requerimientos no funcionales.

# **Reflexión Escenarios de Calidad**

## **Desempeño**

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC1 | Desempeño | ALTA |
| **Fuente** | | |
| Hubs y cerraduras | | |
| **Estímulo** | | |
| Avisar urgentemente sobre un posible fallo en el sistema y en el servicio. | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Las peticiones de envío de fallos son enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC2 | Desempeño | ALTA |
| **Fuente** | | |
| Hubs y cerraduras | | |
| **Estímulo** | | |
| Enviar los datos retornados por la fuente en tiempo real | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Peticiones enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC3 | Desempeño | ALTA |
| **Fuente** | | |
| Hubs y Cerraduras | | |
| **Estímulo** | | |
| Notificar emergencias por mensaje a propietario o en tablero de control a seguridad privada | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Las peticiones de envío de emergencia son enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC4 | Desempeño | MEDIA |
| **Fuente** | | |
| Usuario | | |
| **Estímulo** | | |
| Ver tableros de control con información periódica de alarmas y fallas | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Las peticiones son enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC5 | Desempeño | MEDIA |
| **Fuente** | | |
| Usuario | | |
| **Estímulo** | | |
| Ingresar o salir del sistema | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Las peticiones son enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

Para mejorar los tiempos de respuesta de estas peticiones, se planea colocar el balanceador de carga, el bridge y el servidor de Kafka y Zookeper en máquinas Linux. También colocar balanceadores de carga tanto para distribuir los mensajes de cada tópico a los servidores de mensajería, así como para repartir las peticiones hechas sobre la base de datos.

## **Escalabilidad**

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC6 | Escalabilidad | MEDIA |
| **Fuente** | | |
| Hubs y cerraduras | | |
| **Estímulo** | | |
| Recibir y soportar cargas de varias peticiones al tiempo | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Manejar envío de alarmas de 200 inmuebles en 1500 unidades residenciales con peticiones enviadas en una ventana de 1 minuto con 0% de error y tiempos de respuesta menor a 1 segundo | | |

Al igual que los escenarios de desempeño, para mejorar los tiempos de respuesta de estas peticiones, se planea colocar el balanceador de carga y el servidor de Kafka y Zookeper en máquinas Linux. La escalabilidad no es crítica pues no hay picos en la aplicación, ya que siempre está funcionando con los mismos 300.000 inmuebles.

## **Disponibilidad**

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC7 | Disponibilidad | ALTA |
| **Fuente** | | |
| Sistema | | |
| **Estímulo** | | |
| Tener en funcionamiento todo el sistema | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| El sistema debe estar disponible los 7 días a la semana, durante las 24 horas. El sistema debe estar disponible el 99,99% del tiempo y tener una tiempo de recuperación de 2 horas. | | |

Para lograr este escenario, se implementará ping en el balanceador de carga para detección de fallas y se crearan varias copias del servidor de mongoDB y de los servidores Kafka y Mosquitto, que se estén actualizando ciertos periodos de tiempo, para lograr una redundancia pasiva.

## **Seguridad**

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Tipo de Atributo** | **Prioridad** |
| EC7 | Seguridad | ALTA |
| **Fuente** | | |
| Usuario | | |
| **Estímulo** | | |
| Solo se permite la visualización de tópicos a usuarios autorizados | | |
| **Ambiente** | | |
| Normal | | |
| **Medida Esperada** | | |
| Solo el usuario YALE puede ver todos los tópicos, un propietario solo puede ver su propiedad, seguridad privada los tópicos de sus unidades residenciales y la administración el histórico de alarmas de su unidad residencial. | | |

Este escenario puede solucionarse con la creación de contraseñas para cada tipo de usuario y la encriptación de los datos y contraseñas, como la implementación de protocolos de seguridad como SSL.

# **Declaración de esfuerzo de cada miembro**

Adjunto al documento, se encuentra un archivo EXCEL con la declaración de esfuerzo para cada integrante.