

Programación Python para Big Data - Tarea lección 5

Kevin Martínez García

6 de julio de 2022

Índice

1. Introducción	2
2. Creación de las tablas con Robo3T	2
3. Ejercicio 1	3
4. Ejercicio 2	3
5. Ejercicio 3	4
6. Ejercicio 4	4
7. Ejercicio 5	4

1. Introducción

En la actividad relacionada a la lección 4 se nos solicitó hacer uso de *docker* y *MongoDB* para crear y realizar diferentes operaciones sobre una tabla en una base de datos basada en documentos. A continuación explicaremos cada uno de los ejercicios solicitados junto con capturas de pantalla para poder verificar los resultados.

2. Creación de las tablas con Robo3T

Para empezar, se nos solicitó usar Robo3T para crear una base de datos *actividad* que contuviese la tabla *notas*. Para proceder, en primer lugar debemos lanzar un Docker que ejecute una base de datos MongoDB, para ello, utilizaré el mismo proceso que ya se explicó en la memoria de la Lección 2. Una vez ejecutado el Docker, usamos Robo3T para conectarnos a la base de datos como aparece en la Figura 1.

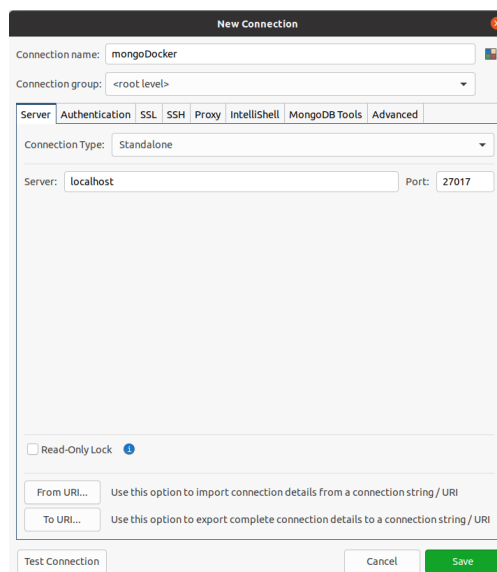


Figura 1: Conexión a MongoDB

Seguidamente, procedemos a generar una nueva base de datos “actividad” junto con la colección “notas”. En la Figura 2 pueden verse ambos elementos ya creados.

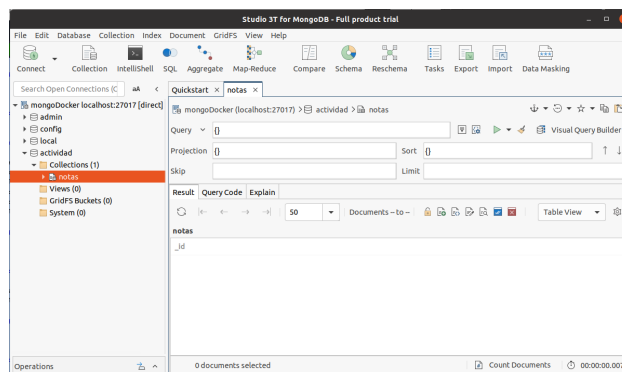
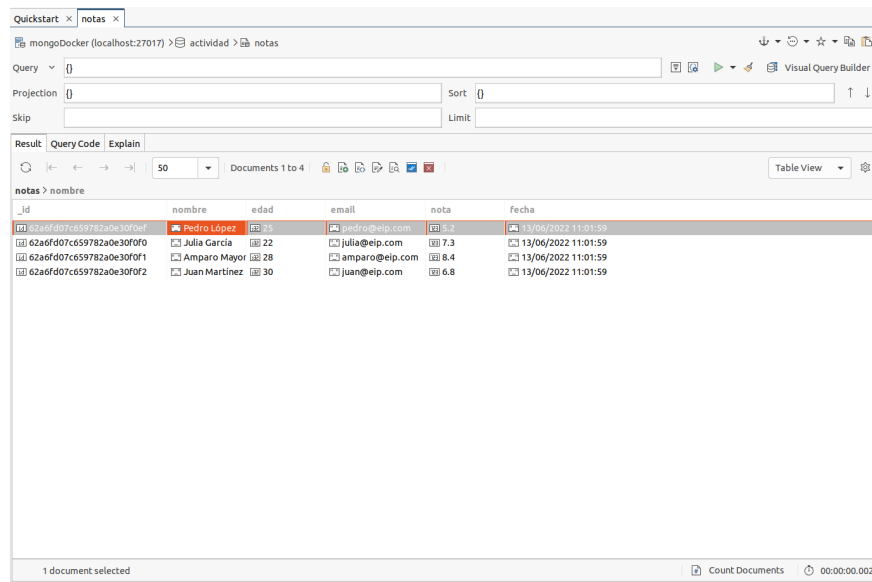


Figura 2: Base de datos actividad y colección notas ya creadas

Una vez hecho esto, ya se pudo proceder con las diferentes actividades solicitadas.

3. Ejercicio 1

Tanto para esta actividad como para todas las que siguen, hicimos uso de la librería `pymongo` descrita en sesiones de teoría para poder operar sobre los diferentes datos. Como primer ejercicio, se nos solicitó introducir en la colección `notas` los datos de una serie de alumnos. Los resultados del proceso son los que pueden verse en la Figura 3 a continuación.



_id	nombre	edad	email	nota	fecha
62aef407c559782a0e30f0f0	Pedro López	22	julia@elp.com	7.3	13/06/2022 11:01:59
62aef407c559782a0e30f0f1	Amparo Mayor	28	amparo@elp.com	8.4	13/06/2022 11:01:59
62aef407c559782a0e30f0f2	Juan Martínez	30	juan@elp.com	6.8	13/06/2022 11:01:59

Figura 3: Datos insertados en la colección `notas`

4. Ejercicio 2

En el segundo ejercicio, se nos solicitó modificar las notas de los alumnos Juan Martínez y Amparo Mayoral. Para ello, recuperamos todos los datos de la colección con `db.notas.find({})` e hicimos una búsqueda de estos dos alumnos con el campo `nombre`. Una vez encontrados modificamos el campo `notas` y actualizamos la colección mediante `update_one`. Los resultados de la actualización pueden verse en la Figura 4. Puede verse como la nota de Amparo Mayoral ha quedado como un 9.3, y la de Juan Martínez como un 7.2.

_id	nombre	edad	email	nota	fecha
62a706c70c03d15253b2db40	Pedro López	25	pedro@eip.com	5.2	13/06/2022 11:43:35
62a706c70c03d15253b2db41	Julia García	22	julia@eip.com	7.3	13/06/2022 11:43:35
62a706c70c03d15253b2db42	Amparo Mayoral	28	amparo@eip.com	9.3	13/06/2022 11:43:35
62a706c70c03d15253b2db43	Juan Martínez	30	juan@eip.com	7.2	13/06/2022 11:43:35

Figura 4: Datos actualizados

5. Ejercicio 3

Para este tercer ejercicio, se nos solicitó realizar una lectura de los datos de la colección. Para ello usamos `db.notas.find({})` para obtener todas las entradas, y un bucle `for` para mostrarlas por terminal. El resultado el proceso es el que aparece en la Figura 5.

```
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion5$ python3 script.py
{'_id': ObjectId('62a706c70c03d15253b2db40'), 'nombre': 'Pedro López', 'edad': 25, 'email': 'pedro@eip.com', 'nota': 5.2, 'fecha': '13/06/2022 11:43:35'}
{'_id': ObjectId('62a706c70c03d15253b2db41'), 'nombre': 'Julia García', 'edad': 22, 'email': 'julia@eip.com', 'nota': 7.3, 'fecha': '13/06/2022 11:43:35'}
{'_id': ObjectId('62a706c70c03d15253b2db42'), 'nombre': 'Amparo Mayoral', 'edad': 28, 'email': 'amparo@eip.com', 'nota': 9.3, 'fecha': '13/06/2022 11:43:35'}
{'_id': ObjectId('62a706c70c03d15253b2db43'), 'nombre': 'Juan Martínez', 'edad': 30, 'email': 'juan@eip.com', 'nota': 7.2, 'fecha': '13/06/2022 11:43:35'}
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion5$
```

Figura 5: Datos en la colección notas

6. Ejercicio 4

En el ejercicio 4 se nos pedía mostrar los datos de aquellos alumnos con una nota entre 7 y 7.5. Para ello, hicimos uso de la instrucción `datos = db.notas.find({"nota": {"$gte": 7, "$lte": 7.5}})`. Con esta instrucción recuperamos aquellas entradas cuyo campo notas sea mayor o igual a 7 y menor o igual que 7.5. Los resultados de la consulta pueden verse en la Figura 6.

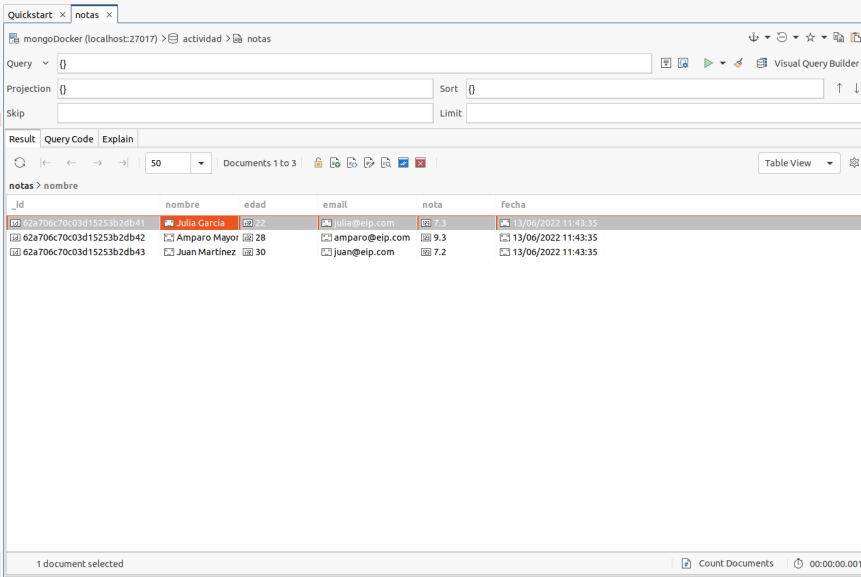
```
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion5$ python3 script.py
{'_id': ObjectId('62a706c70c03d15253b2db41'), 'nombre': 'Julia García', 'edad': 22, 'email': 'julia@eip.com', 'nota': 7.3, 'fecha': '13/06/2022 11:43:35'}
{'_id': ObjectId('62a706c70c03d15253b2db43'), 'nombre': 'Juan Martínez', 'edad': 30, 'email': 'juan@eip.com', 'nota': 7.2, 'fecha': '13/06/2022 11:43:35'}
(base) kevin@kevin-ubuntu:~/Documentos/MasterPython/9.-ProgramacionBigData/Leccion5$
```

Figura 6: Alumnos con notas entre 7 y 7.5

7. Ejercicio 5

Para terminar, se nos pidió borrar los datos del alumno Pedro López previamente insertado en la base de datos. Para ello hicimos uso en primer lugar de la instrucción `db.user.find({"nombre": "Pedro López"})` para obtener los datos de dicho alumno. Seguidamente, obtuvimos su `_id` y usamos la instrucción

`delete_one()` para borrarlo. En la Figura 7 puede verse una captura de la base de datos en la que dicho alumno ya no se encuentra presente.



_id	nombre	edad	email	nota	fecha
62a706c70c03d15253b2db41	Pedro López	25	pedrolopez@elp.com	6	13/06/2022 11:43:35
62a706c70c03d15253b2db42	Amparo Mayor	28	amparo@elp.com	9.3	13/06/2022 11:43:35
62a706c70c03d15253b2db43	Juan Martínez	30	juan@elp.com	7.2	13/06/2022 11:43:35

Figura 7: Colección notas sin el alumno Pedro López