

IMPORTANTE – ESTE MANUAL ESTÁ REALIZADO EN UN SO WINDOWS 10

## Instrucciones puesta en marcha docker con MongoDB

Para el fichero docker-compose.yml se puede descargar desde mi github en:

<https://raw.githubusercontent.com/jbermejog/pyhton-mongodb/master/docker-compose.yml>

Para iniciar el contenedor ejecutaremos:

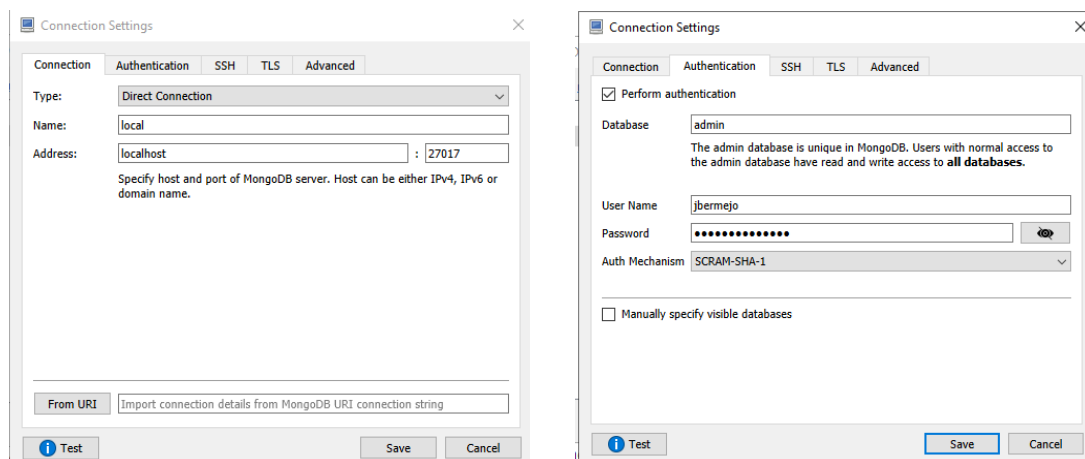
```
λ docker-compose up -d
Starting mongodb ... done
```

Cuando pone “done” significa que se está ejecutando y ya tenemos disponible la conexión con el servidor por el puerto 27017.

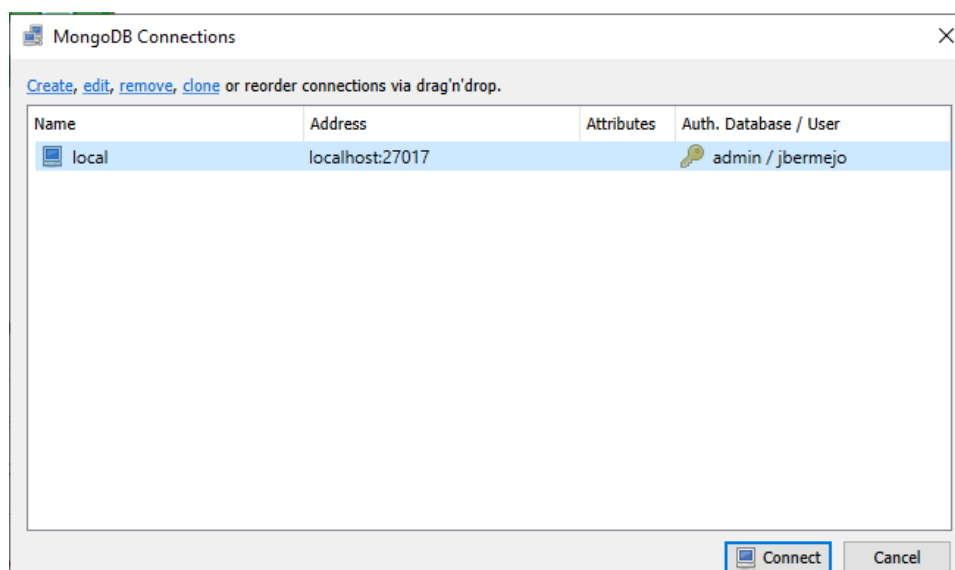
## En Robo3T

Procedemos a realizar las tareas solicitadas en la actividad.

Para ello ejecutamos Robo3T y creamos una nueva conexión con los parámetros definidos de usuario y password del docker-composer.



The image shows two screenshots of the Robo3T Connection Settings dialog box. The left screenshot shows the 'Connection' tab with 'Direct Connection' selected, Name 'local', and Address 'localhost:27017'. The right screenshot shows the 'Authentication' tab with 'Perform authentication' checked, Database 'admin', User Name 'jbermejo', and Password masked with dots. The Auth Mechanism is set to 'SCRAM-SHA-1'.



The image shows the Robo3T MongoDB Connections window. It displays a table with the following data:

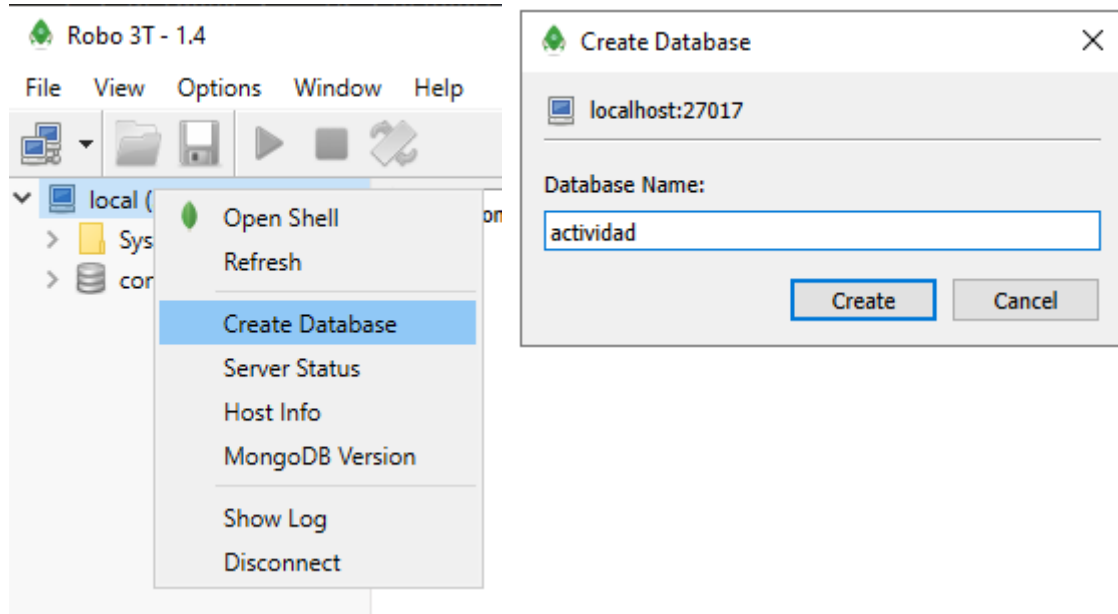
Name	Address	Attributes	Auth. Database / User
local	localhost:27017		admin / jbermejo

At the bottom right, there is a 'Connect' button and a 'Cancel' button.

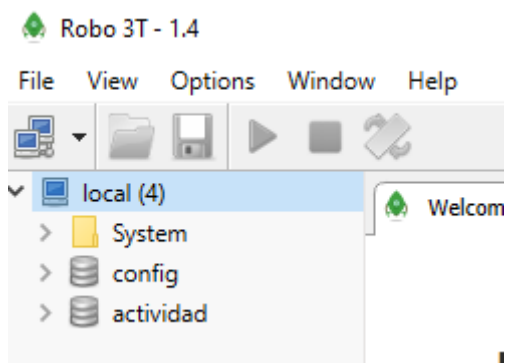
Le damos a guardar y conectamos para realizar las tareas solicitadas

### Creación de DB con nombre “actividad”

Una vez conectado, hacemos clic en con el botón derecho sobre el icono de la conexión y seleccionamos “**Create Database**”, tal y como se nos pide en esta actividad, escribimos el nombre “**actividad**” en “**Database Name:**” y pulsamos en el botón “**Create**”

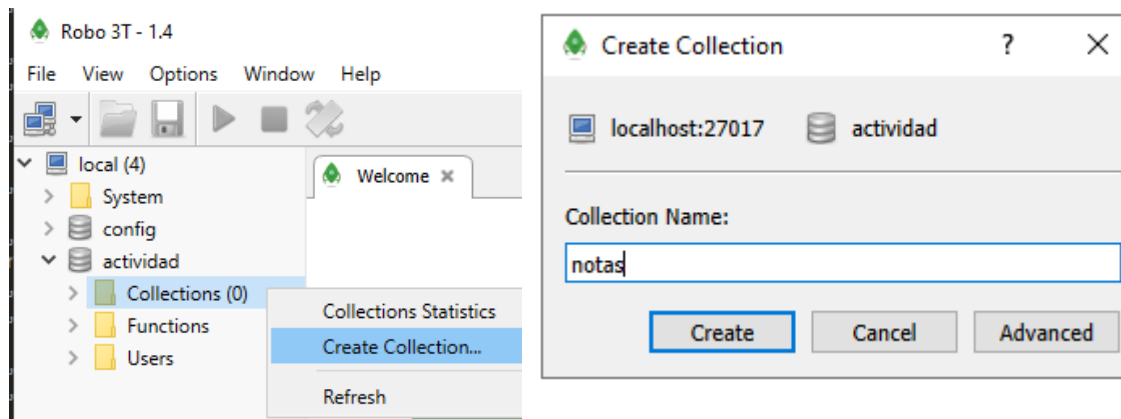


Esto crea la base de datos “actividad” tal y como se muestra en la imagen.

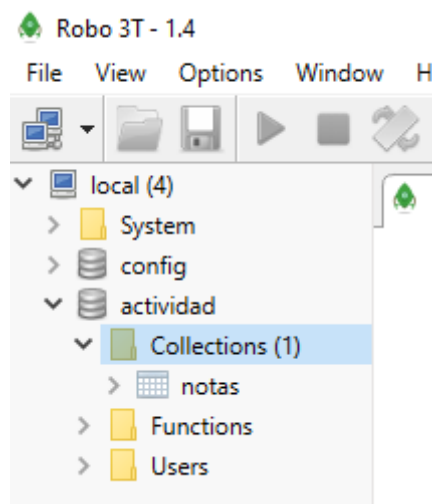


### Crear una colección “notas”

Para crear la colección, hacemos clic en el botón derecho encima de “**Collections**” dentro de la DB “**actividad**” y seleccionamos “**Create Collection...**” como se muestra en la siguiente imagen.



Escribimos en “**Collection Name:**” notas tal y como se nos solicita y pulsamos en el botón “**Create**”



## Script mongodb\_javierbermejo.py

Realizamos un script para el cual crearemos un fichero “.env” con las variables de entorno y con los datos de conexión a la DB.

El script se puede descargar de mi github en:

[https://raw.githubusercontent.com/jbermejog/pyhton-mongodb/master/mongodb\\_javierbermejo/mongodb\\_javierbermejo.py](https://raw.githubusercontent.com/jbermejog/pyhton-mongodb/master/mongodb_javierbermejo/mongodb_javierbermejo.py)

el fichero de ejemplo “.env” se puede descargar en:

[https://raw.githubusercontent.com/jbermejog/pyhton-mongodb/master/mongodb\\_javierbermejo/.env.example](https://raw.githubusercontent.com/jbermejog/pyhton-mongodb/master/mongodb_javierbermejo/.env.example)

Se tendría que descargar y renombrar a “.env” poniendo los datos de acceso a tu servidor MongoDB

Una vez realizado ese proceso de descarga y configuración procedemos a realizar las tareas solicitadas que son las que realmente puntúan:

## 1 - Insertar colección de datos

Cuando se ejecuta la app se obtiene el siguiente resultado:

```
λ py mongodb_javierbermejo.py
Presiona enter para inciar las tareas...|
```

Pulsamos “Enter” y realiza la inserción a la base de datos y se nos muestra lo siguiente:

```
Tarea 1 insertar datos realizada correctamente ....
Pulsa enter para iniciar la segunda Tarea ...|
```

Para verificar que los datos se encuentran en la DB usamos Robo3T.

Robo 3T - 1.4

File View Options Window Help

local (4)

- System
- actividad
  - Collections (1)
    - notas
  - Indexes
  - Functions
  - Users
  - config

db.getCollection('...'

db.getCollection('...'

db.getCollection('...'

db.getCollection('...'

db.getCollection('...'

local localhost:27017 actividad

db.getCollection('notas').find({})

notas 0.005 sec. 0

Key	Value	Type
(1) ObjectId("6190e483ab02b5ce7006db3e")	{ 6 fields }	Object
_id	ObjectId("6190e483ab02b5ce7006db3e")	ObjectId
nombre	Pedro López	String
edad	25	Int32
email	pedro@eip.com	String
nota	5.2	Double
fecha	2021-07-12 12:45:26.000Z	Date
(2) ObjectId("6190e483ab02b5ce7006db3f")	{ 6 fields }	Object
_id	ObjectId("6190e483ab02b5ce7006db3f")	ObjectId
nombre	Julia García	String
edad	22	Int32
email	julia@eip.com	String
nota	7.3	Double
fecha	2021-07-12 12:45:26.000Z	Date
(3) ObjectId("6190e483ab02b5ce7006db40")	{ 6 fields }	Object
_id	ObjectId("6190e483ab02b5ce7006db40")	ObjectId
nombre	Amparo Mayoral	String
edad	28	Int32
email	amparo@eip.com	String
nota	8.4	Double
fecha	2021-07-12 12:45:26.000Z	Date
(4) ObjectId("6190e483ab02b5ce7006db41")	{ 6 fields }	Object
_id	ObjectId("6190e483ab02b5ce7006db41")	ObjectId
nombre	Juan Martínez	String
edad	30	Int32
email	juan@eip.com	String
nota	6.8	Double
fecha	2021-07-12 12:45:26.000Z	Date

Como se puede observar se insertan los datos manteniendo el tipo de dato, para ello se ha creado una función dentro del script que ajusta la fecha para que sea del tipo “Date”.

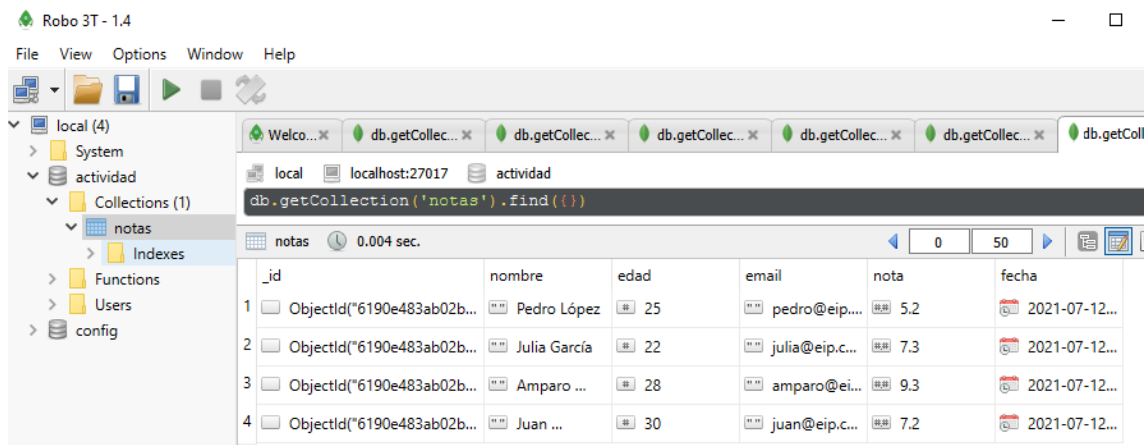
## 2 - Actualización de los datos

Se nos pide la actualización de las notas de dos personas.

Pulsamos “Enter” para que se realice la tarea 2 y se nos muestra lo siguiente:

```
Pulsa enter para iniciar la segunda Tarea ...
Tarea 2 actualizar datos realizada correctamente ....
Pulsa enter para iniciar la tercera Tarea ...|
```

Ahora revisamos que los cambios se han realizado, esta vez cambiamos el modo de la vista a tabla y adjuntamos pantallazo como se nos pide.



Podemos observar los cambios a 9.3 y 7.2 solicitados

## 3 - Lectura de los datos

Continuamos con la ejecución del script pulsando “Enter” y se nos muestran los datos:

```
Pulsa enter para iniciar la tercera Tarea ...
ID          NOMBRE      EDAD  EMAIL          NOTA  FECHA
6190e483ab02b5ce7006db3e  Pedro López    25    pedro@eip.com  5.2   2021-07-12 12:45:26
6190e483ab02b5ce7006db3f  Julia García   22    julia@eip.com  7.3   2021-07-12 12:45:26
6190e483ab02b5ce7006db40  Amparo Mayoral 28    amparo@eip.com 9.3   2021-07-12 12:45:26
6190e483ab02b5ce7006db41  Juan Martínez  30    juan@eip.com   7.2   2021-07-12 12:45:26
Tarea 3 listado de datos realizada correctamente ....
Pulsa enter para iniciar la cuarta Tarea ...
```

## 4 - Buscar las notas entre 7 y 7.5

Continuamos con la ejecución del script pulsando “Enter” y se nos muestran los datos:

```
ID          NOMBRE      EDAD  EMAIL          NOTA  FECHA
6190e483ab02b5ce7006db3f  Julia García   22    julia@eip.com  7.3   2021-07-12 12:45:26
6190e483ab02b5ce7006db41  Juan Martínez  30    juan@eip.com   7.2   2021-07-12 12:45:26
Tarea 4 filtrar notas de 7 a 7.5 realizada correctamente ....
Pulsa enter para iniciar la quinta Tarea ...
```

Mostramos captura de pantalla en Robo3T tal y como se nos pide aplicando el filtro:

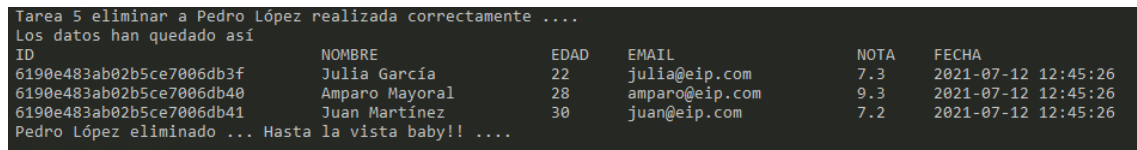
## Actividad lección 5 CRUD en Pymongo y visualizando con Robo3T

Javier Bermejo



### 5 - Eliminar los datos

Continuamos el script pulsando “Enter” y se nos muestra



Revisamos el Robo3T

