Lab-02

Ejecutar un contenedor que corre **MongoDB** y con el cual nos conectaremos por medio de **Python**

 # docker run -d -p 27017:27017 --name m1 mongo (Iniciar contenedor corriendo MongoDB)

```
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
1bc677758ad7: Pull complete
7eb83bb7be98: Pull complete
e95121721c4c: Pull complete
799041b403ca: Pull complete
1828e70ef29a: Pull complete
8e3781beae9e: Pull complete
5d5753162333: Pull complete
44dd404b40f4: Pull complete
44599c9d5d1b: Pull complete
Digest: sha256:928347070dc089a596f869a22a4204c0feace3eb03470a6a2de6814f11fb7309
Status: Downloaded newer image for mongo:latest
3c89c1b6364ecf2789bdeb3ad66e47b2d3f8a7768b65a8e30064372f8da8952b
```

 # docker exec -it m1 /bin/bash (Conectarse al contenedor, ejecutando bash)

 # mongosh (Conectarse a MongoDB)

```
Current Mongosh Log ID: 64625494f54a3b6d12837aaf
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectio
nTimeoutMS=2000&appName=mongosh+1.8.2
Using MongoDB:          6.0.5
Using Mongosh:          1.8.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB per
iodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2023-05-15T15:48:46.193+00:00: Using the XFS filesystem is strongly recommended with
 the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2023-05-15T15:48:46.762+00:00: Access control is not enabled for the database. Read
and write access to data and configuration is unrestricted
   2023-05-15T15:48:46.762+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'alway
s'. We suggest setting it to 'never'
   2023-05-15T15:48:46.762+00:00: vm.max_map_count is too low
------

test> exit
root@3c89c1b6364e:/#

root@3c89c1b6364e:/# exit
exit
```

 # apt install python (Instalar Python)

```
root@10:/home/usuario/DevOps/docker/Lab-02# apt install python
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'python-is-python2' instead of 'python'
The following additional packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python2
  python2-minimal python2.7 python2.7-minimal
Suggested packages:
  python2-doc python-tk python2.7-doc binfmt-support
The following NEW packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-is-python2
  python2 python2-minimal python2.7 python2.7-minimal
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 3,977 kB of archives.
After this operation, 16.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

 # apt-get update **(**Obtiene información sobre las versiones más recientes de paquetes y sus dependencias**)**

 # apt-get install python3-pymongo (Instalar librería de python para mongoDB)

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-bson python3-bson-ext python3-gridfs python3-pymongo-ext
Suggested packages:
  python-pymongo-doc
The following NEW packages will be installed:
  python3-bson python3-bson-ext python3-gridfs python3-pymongo
  python3-pymongo-ext
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 432 kB of archives.
After this operation, 1,702 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

 # python3 populate.py (Ejecuta script de python que crea colección “pet” y agrega registros)

```
Nombre de la DB:  mi-db
<pymongo.results.InsertManyResult object at 0x7f79f1592600>
```

 # python3 find.py (Ejecuta script de python que recorre los registros de la colección “pet” y los imprime)

```
Imprime un registro
 {'_id': ObjectId('6462851244336180c284f38c'), 'name': 'firulais', 'owner': 'jahir', 'specie': 'perro'}

Imprime todos los registros
{'_id': ObjectId('6462851244336180c284f38c'), 'name': 'firulais', 'owner': 'jahir', 'specie': 'perro'}
{'_id': ObjectId('6462851244336180c284f38d'), 'name': 'taco', 'owner': 'jonathan', 'specie': 'perro'}
{'_id': ObjectId('6462851244336180c284f38e'), 'name': 'garfield', 'owner': 'erick', 'specie': 'gato'}
{'_id': ObjectId('6462851244336180c284f38f'), 'name': 'charlotte', 'owner': 'juan daniel', 'specie': 'araña'}
{'_id': ObjectId('6462851244336180c284f390'), 'name': 'solovino', 'owner': 'jorge', 'specie': 'cuyo'}
root@10:/home/usuario/DevOps/docker/Lab-02#
```

populate.py

```python
import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["mi-db"]

# Name
print("Nombre de la DB: ", db.name)

# Crea colección e inserta un registro
print(db.pet.insert_many([
    {
        "name": "firulais",
        "owner": "jahir",
        "specie": "perro"
    },
    {
        "name": "taco",
        "owner": "jonathan",
        "specie": "perro"
    },
    {
        "name": "garfield",
        "owner": "erick",
        "specie": "gato"
    },
    {
        "name": "charlotte",
        "owner": "juan daniel",
        "specie": "araña"
    },
    {
        "name": "solovino",
        "owner": "jorge",
        "specie": "cuyo"
    },
]))
```

find.py

```python
import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["mi-db"]

# Obtén un registro
print("Imprime un registro\n", db.pet.find_one(), "\n")

# Obtén todos los registros
print("Imprime todos los registros")
for pet in db.pet.find():
    print(pet)
```

 # docker exec -it m1 /bin/bash (Conectarse al contenedor, ejecutando bash)

 # mongosh (Conectarse a MongoDB)

test> show dbs (Mostrar bases de datos)

```
admin    40.00 KiB
config  108.00 KiB
local    40.00 KiB
mi-db    40.00 KiB
```

Test> use mi-db (Seleccionar base de datos “mi-db”)

```
test> use mi-db
switched to db mi-db
```

mi-db> show collections (Mostrar colecciones de la base de datos actual)

```
mi-db> show collections
pet
```

mi-db> db.pet.find() (Muestra todos los registros de la colección “pet”)

```
mi-db> db.pet.find()
[
  {
    _id: ObjectId("6462851244336180c284f38c"),
    name: 'firulais',
    owner: 'jahir',
    specie: 'perro'
  },
  {
    _id: ObjectId("6462851244336180c284f38d"),
    name: 'taco',
    owner: 'jonathan',
    specie: 'perro'
  },
  {
    _id: ObjectId("6462851244336180c284f38e"),
    name: 'garfield',
    owner: 'erick',
    specie: 'gato'
  },
  {
    _id: ObjectId("6462851244336180c284f38f"),
    name: 'charlotte',
    owner: 'juan daniel',
    specie: 'araña'
  },
  {
    _id: ObjectId("6462851244336180c284f390"),
    name: 'solovino',
    owner: 'jorge',
    specie: 'cuyo'
  }
]
```

```
mi-db> exit
root@3c89c1b6364e:/# exit
exit
root@10:/home/usuario/DevOps/docker/Lab-02#
```

 # docker stop m1 (Detener el contenedor m1)

 # docker rm m1 (Eliminar contenedor m1)

 # docker ps –a (Listar contenedores en ejecución y detenidos actuales)

```
root@10:/home/usuario/DevOps/docker/Lab-02# docker stop m1
m1
root@10:/home/usuario/DevOps/docker/Lab-02# docker rm m1
m1
root@10:/home/usuario/DevOps/docker/Lab-02# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@10:/home/usuario/DevOps/docker/Lab-02#
```