

Mi vida en un contenedor (Retos Docker)

Reto 1:

Actualizar el sistema

```
ub@ub:~$ sudo apt update && sudo apt upgrade
```

Extraiga y ejecute el contenedor Ubuntu

- docker pull ubuntu

```
ub@ub:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
20043066d3d5: Pull complete
Digest: sha256:c35e29c9450151419d9448b0fd75374fec4fff364a27f176fb458d472dfc9e54
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

- docker run -it --name reto-ubuntu ubuntu

```
ub@ub:~$ docker run -it --name reto-ubuntu ubuntu
```

```
root@26d9767c43a1:/#
```

Instalar dependencias dentro del contenedor

- apt update
- apt install python3 python3-pip -y

```
root@26d9767c43a1:/# apt install python3 python3-pip -y
```

- apt install python3-requests -y

```
root@26d9767c43a1:/# apt install python3-requests -y
```

- apt install mysql-client python3-mysql.connector python3-mysqldb -y

```
root@26d9767c43a1:/# apt install mysql-client python3-mysql.connector python3-mysqldb -y
```

Enviar como imagen personalizada

- docker commit reto-ubuntu img-reto-ubuntu

```
ub@ub:~$ docker commit reto-ubuntu img-reto-ubuntu
sha256:df7c1e8c4e8fe37446d9bb977ff61b0d423e4d983f39a12e84870a174b641f20
```

Reto 2:

Cree un directorio de montaje y ejecute un contenedor con un montaje Bind

```
ub@ub:~/reto-ubuntu-python$ tree
```

```
.
├── bing_volume
├── dockerfiles
│   ├── Dockerfile
│   └── requirements.txt
```

```
docker run -d --name ubuntu_app \
-v ~/reto-ubuntu-python/bing_volume:/data \
myubuntu-python:1.0 tail -f /dev/null
```

```
ub@ub:~/reto-ubuntu-python$ docker run -d --name ubuntu_app \
-v ~/reto-ubuntu-python/bing_volume:/data \
myubuntu-python:1.0 tail -f /dev/null
537a4e02a68528ded70d2d1044781650a29da6420305c10efedd0b6a01289d1d
```

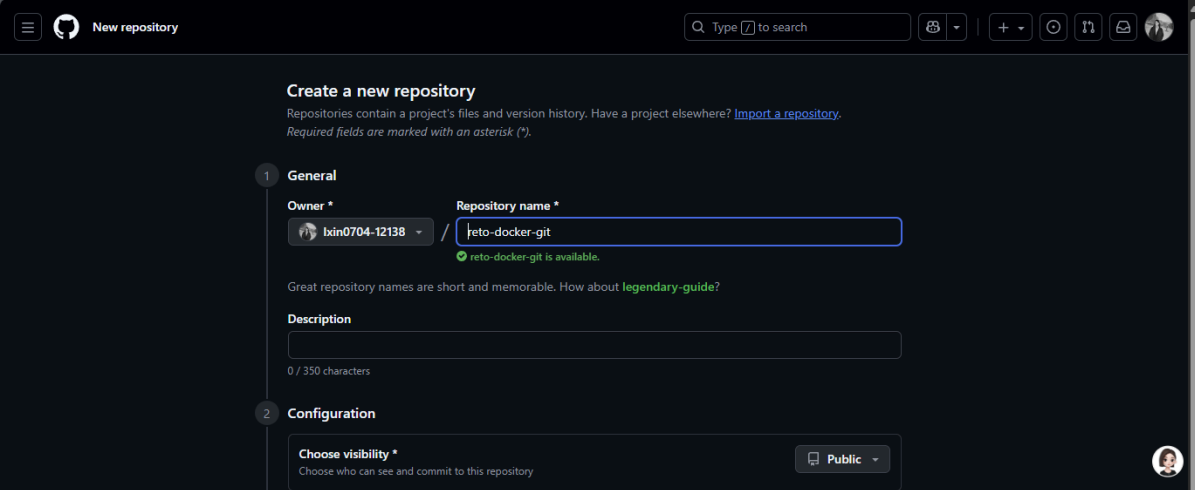
Introducir prueba de contenedor

- docker exec -it ubuntu_app bash
- ls -la /data
- echo "Hola" > /data/test.txt
- exit
- cat ~/reto-ubuntu-python/bing_volume/test.txt

```
ub@ub:~/reto-ubuntu-python$ docker exec -it ubuntu_app bash
root@537a4e02a685:/app# ls -la /data
total 8
drwxrwxr-x 2 1000 1000 4096 Nov 16 22:29 .
drwxr-xr-x 1 root root 4096 Nov 16 23:00 ..
root@537a4e02a685:/app# echo "Hola" > /data/test.txt
root@537a4e02a685:/app# exit
exit
ub@ub:~/reto-ubuntu-python$ cat ~/reto-ubuntu-python/bing_volume/test.txt
Hola
```

Reto 3:

Crear un repositorio



New repository

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

1 General

Owner *

Repository name *

Great repository names are short and memorable. How about [legendary-guide](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

Configurar la información global del usuario

```
ub@ub:~/reto-ubuntu-python$ git config --global user.name "lxin0704-12138"
ub@ub:~/reto-ubuntu-python$ git config --global user.email "lxin0704@gmail.com"
```

Conectarse a GitHub

```

ub@ub:~/reto-ubuntu-python$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ub/reto-ubuntu-python/.git/

```

```

ub@ub:~/reto-ubuntu-python$ git branch -M main

```

```

ub@ub:~/reto-ubuntu-python$ git remote add origin https://github.com/lxin0704-12138/reto-docker-git.git
ub@ub:~/reto-ubuntu-python$ git push -u origin main
Username for 'https://github.com': lxin0704-12138
Password for 'https://lxin0704-12138@github.com':
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 6 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (16/16), 1.52 KiB | 1.52 MiB/s, done.
Total 16 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/lxin0704-12138/reto-docker-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

The screenshot shows the GitHub interface for a repository named 'reto-docker-git' owned by 'lxin0704-12138'. The repository is public and has 1 branch (main) and 0 tags. The 'main' branch has 5 commits. The repository contains three files: 'bing_volume' (last commit 35 minutes ago), 'dockerfiles' (last commit 1 hour ago), and 'README.md' (last commit 18 minutes ago). The README file is open, showing the title 'reto-docker' and the repository name 'reto-docker-git'. On the right side, there are sections for 'About' (no description), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages'.

Reto 4:

Extraiga la imagen MySQL

```

ub@ub:~/reto-ubuntu-python$ docker pull mysql:8.0
8.0: Pulling from library/mysql
Digest: sha256:f37951fc3753a6a22d6c7bf6978c5e5fefcf6f31814d98c582524f98eae52b21
Status: Image is up to date for mysql:8.0
docker.io/library/mysql:8.0

```

Crear un contenedor MySQL

```
docker run -d \
  --name mysql_coches \
  -e MYSQL_ROOT_PASSWORD=Lxy20050704. \
  -e MYSQL_DATABASE=coches_db \
  -p 3306:3306 \
  mysql:8.0
```

```
ub@ub:~/reto-ubuntu-python$ docker run -d \
  --name mysql_coches \
  -e MYSQL_ROOT_PASSWORD=Lxy20050704. \
  -e MYSQL_DATABASE=coches_db \
  -p 3306:3306 \
  mysql:8.0
9eab38365185c6c126472dad4b9fe25a8550338c2e0572955db5be7522478230
```

Acceder a MySQL

```
- docker exec -it mysql_coches mysql -u root -p
```

```
ub@ub:~/reto-ubuntu-python$ docker exec -it mysql_coches mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Crear tabla

```
USE coches_db;

CREATE TABLE coches (
  id INT AUTO_INCREMENT PRIMARY KEY,
  marca VARCHAR(50),
  modelo VARCHAR(50),
  color VARCHAR(30),
  km INT,
  precio INT
);
```

```
mysql> CREATE TABLE coches (
-> id INT AUTO_INCREMENT PRIMARY KEY,
-> marca VARCHAR(50),
-> modelo VARCHAR(50),
-> color VARCHAR(30),
-> km INT,
-> precio DECIMAL(10,2)
-> );
Query OK, 0 rows affected (0.10 sec)
```

```
INSERT INTO coches (marca, modelo, color, km, precio) VALUES
('Toyota','Corolla','Blanco',20000,15000),
('Honda','Civic','Rojo',30000,17000),
('Ford','Focus','Azul',25000,16000),
('Nissan','Sentra','Gris',35000,15500),
('Volkswagen','Jetta','Blanco',40000,16500),
('Hyundai','Elantra','Negro',45000,14500),
('Kia','Rio','Azul',30000,13500),
('Mazda','Mazda3','Rojo',48000,15800),
('BMW','Serie 3','Negro',70000,28000),
('Audi','A4','Azul',69000,29000);
```

```
mysql> INSERT INTO coches (marca, modelo, color, km, precio) VALUES
-> ('Toyota','Corolla','Blanco',20000,15000),
-> ('Honda','Civic','Rojo',30000,17000),
-> ('Ford','Focus','Azul',25000,16000),
-> ('Nissan','Sentra','Gris',35000,15500),
-> ('Volkswagen','Jetta','Blanco',40000,16500),
-> ('Hyundai','Elantra','Negro',45000,14500),
-> ('Kia','Rio','Azul',30000,13500),
-> ('Mazda','Mazda3','Rojo',48000,15800),
-> ('BMW','Serie 3','Negro',70000,28000),
-> ('Audi','A4','Azul',69000,29000);
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Reto 5:

Crear un script en Python

```
ub@ub:~/reto-ubuntu-python$ cd bing_volume
ub@ub:~/reto-ubuntu-python/bing_volume$ sudo nano read_mysql.py|
```

```
GNU nano 7.2 read_mysql.py *
import mysql.connector

conexion = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Lxy20050704.",
    database="coches_db"
)

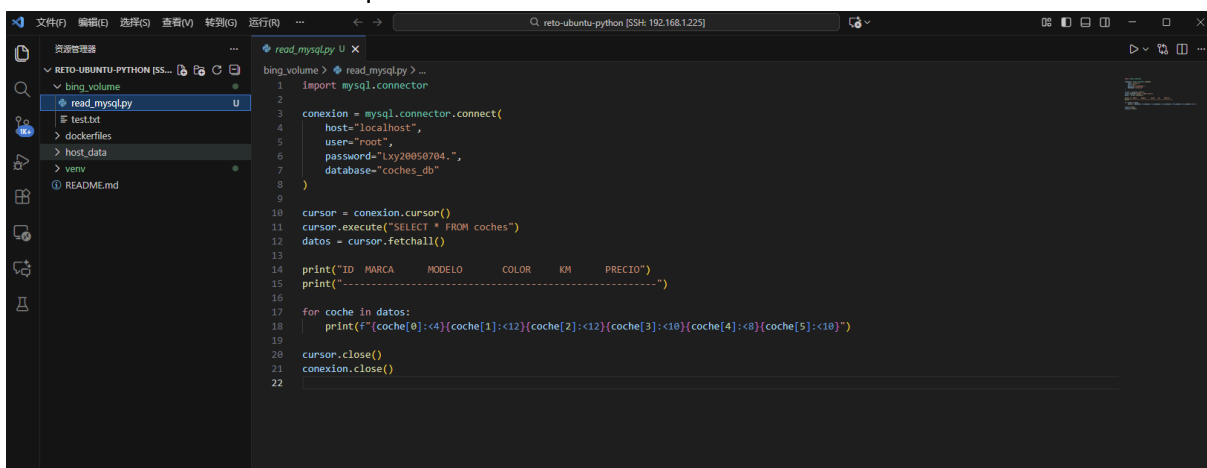
cursor = conexion.cursor()
cursor.execute("SELECT * FROM coches")
datos = cursor.fetchall()

print("ID  MARCA      MODELO      COLOR      KM      PRECIO")
print("-----")

for coche in datos:
    print(f"{coche[0]:<4}{coche[1]:<12}{coche[2]:<12}{coche[3]:<10}{coche[4]:<8}{coche[5]:<10}")

cursor.close()
conexion.close()
```

Conexión remota a una máquina virtual en vs code



Instalar pip3

- sudo apt install python3-pip -y

```
ub@ub:~/reto-ubuntu-python$ sudo apt update && sudo apt install python3-pip -y
```

Instala la herramienta de entorno virtual venv

- sudo apt install python3-venv -y

```
ub@ub:~/reto-ubuntu-python$ sudo apt install python3-venv -y
```

Crear un directorio

- python3 -m venv venv

```
ub@ub:~/reto-ubuntu-python$ python3 -m venv venv
ub@ub:~/reto-ubuntu-python$ ls
bing_volume  dockerfiles  host_data  README.md  venv
```

Activar el entorno virtual

- source venv/bin/activate

```
ub@ub:~/reto-ubuntu-python$ source venv/bin/activate
```

Instalar MySQL Connector (dentro del entorno virtual)

- pip install mysql-connector-python

```
(venv) ub@ub:~/reto-ubuntu-python$ pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.5.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (7.5 kB)
  Downloading mysql_connector_python-9.5.0-cp312-cp312-manylinux_2_28_x86_64.whl (34.1 MB)
    34.1/34.1 MB 15.6 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.5.0
```

- pip install tabulate

```
(venv) ub@ub:~/reto-ubuntu-python$ pip install tabulate
Collecting tabulate
  Downloading tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)
  Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Installing collected packages: tabulate
Successfully installed tabulate-0.9.0
```

Ejecuta el script

- python3 bing_volume/read_mysql.py

```
ub@ub:~/reto-ubuntu-python$ python bing_volume/read_mysql.py
```

ID	MARCA	MODELO	COLOR	KM	PRECIO
1	Toyota	Corolla	Blanco	20000	15000.00
2	Honda	Civic	Rojo	30000	17000.00
3	Ford	Focus	Azul	25000	16000.00
4	Nissan	Sentra	Gris	35000	15500.00
5	Volkswagen	Jetta	Blanco	40000	16500.00
6	Hyundai	Elantra	Negro	45000	14500.00
7	Kia	Rio	Azul	30000	13500.00
8	Mazda	Mazda3	Rojo	48000	15800.00
9	BMW	Serie 3	Negro	70000	28000.00
10	Audi	A4	Azul	69000	29000.00

Confirmar en Git

- git add .
- git commit -m "..."
- git push

```
ub@ub:~/reto-ubuntu-python$ git add .
ub@ub:~/reto-ubuntu-python$ git commit -m "Reto5 python script"
```

```
ub@ub:~/reto-ubuntu-python$ git push
```

```

Username for 'https://github.com': lxin0704-12138
Password for 'https://lxin0704-12138@github.com':
Enumerating objects: 1382, done.
Counting objects: 100% (1382/1382), done.
Delta compression using up to 6 threads
Compressing objects: 100% (1367/1367), done.
Writing objects: 100% (1380/1380), 26.23 MiB | 3.80 MiB/s, done.
Total 1380 (delta 118), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (118/118), done.
To https://github.com/lxin0704-12138/reto-docker-git.git
657c321..be84295  main -> main

```

Reto 6:

Crear un archivo JSON

```

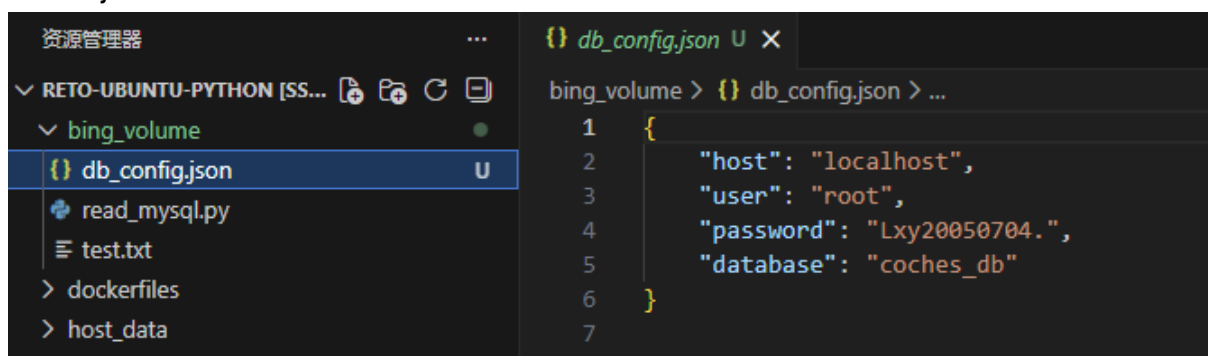
ub@ub:~/reto-ubuntu-python$ cd bing_volume
ub@ub:~/reto-ubuntu-python/bing_volume$ sudo nano db_config.json

```

```

{
    "host": "localhost",
    "user": "root",
    "password": "Lxy20050704.",
    "database": "coches_db"
}

```



Crear un nuevo script Python para leer JSON

```

ub@ub:~/reto-ubuntu-python/bing_volume$ sudo nano mysql_json.py

```

```

import json
import mysql.connector

with open("config.json") as f:
    config = json.load(f)

conexion = mysql.connector.connect(**config)

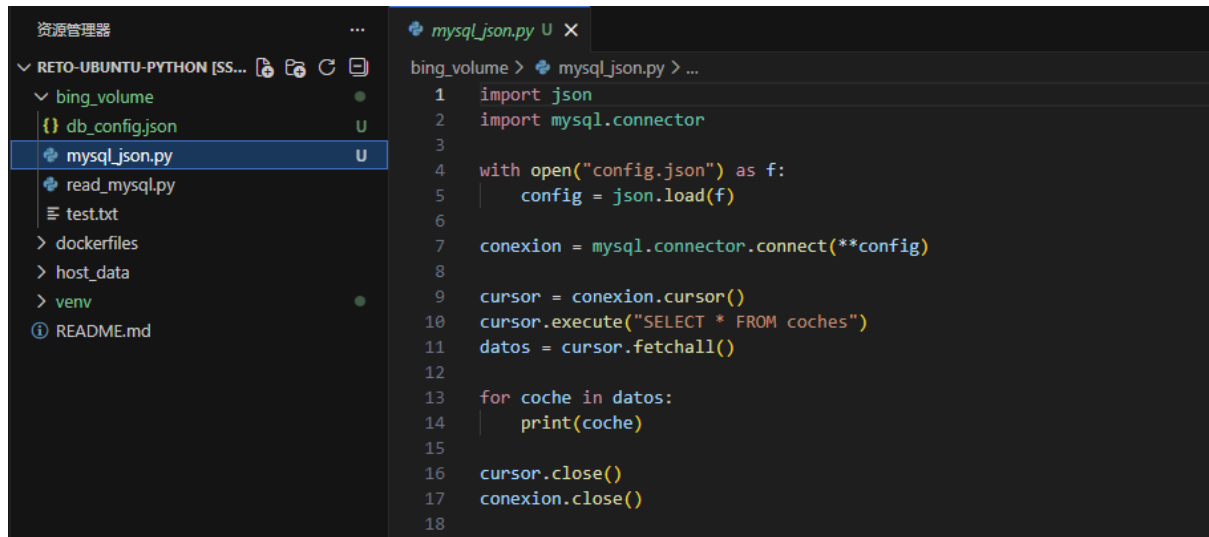
cursor = conexion.cursor()
cursor.execute("SELECT * FROM coches")

```

```
datos = cursor.fetchall()
```

```
for coche in datos:  
    print(coche)
```

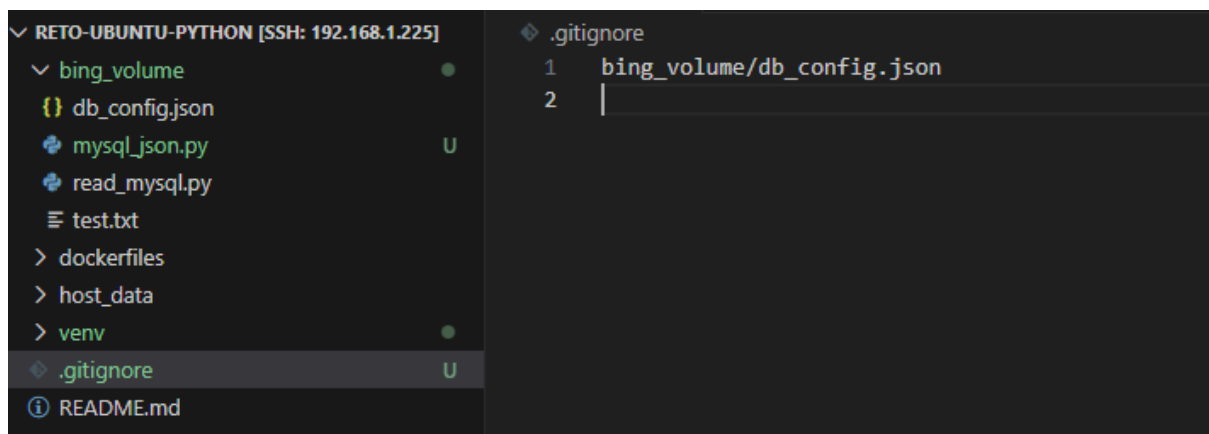
```
cursor.close()  
conexion.close()
```



```
mysql_json.py U X  
bing_volume > mysql_json.py > ...  
1 import json  
2 import mysql.connector  
3  
4 with open("config.json") as f:  
5     config = json.load(f)  
6  
7 conexion = mysql.connector.connect(**config)  
8  
9 cursor = conexion.cursor()  
10 cursor.execute("SELECT * FROM coches")  
11 datos = cursor.fetchall()  
12  
13 for coche in datos:  
14     print(coche)  
15  
16 cursor.close()  
17 conexion.close()  
18
```

Crea un archivo .gitignore

```
ub@ub:~/reto-ubuntu-python$ sudo nano .gitignore
```



```
RETO-UBUNTU-PYTHON [SSH: 192.168.1.225]  
bing_volume  
db_config.json  
mysql_json.py U  
read_mysql.py  
test.txt  
dockerfiles  
host_data  
venv  
.gitignore U  
README.md  
.gitignore  
1 bing_volume/db_config.json  
2
```

Confirmar en Git

```

(venv) ub@ub:~/reto-ubuntu-python$ git add .
(venv) ub@ub:~/reto-ubuntu-python$ git commit -m "Reto6 config de json y ignore"
[main 93fd89d] Reto6 config de json y ignore
 2 files changed, 18 insertions(+)
   create mode 100644 .gitignore
   create mode 100644 bing_volume/mysql_json.py
(venv) ub@ub:~/reto-ubuntu-python$ git push
Username for 'https://github.com': lxin0704-12138
Password for 'https://lxin0704-12138@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 728 bytes | 728.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lxin0704-12138/reto-docker-git
 be84295..93fd89d  main -> main

```

Reto 7:

Embelllecimiento del formato

```

bing_volume > read_mysql.py > ...
1  import mysql.connector
2
3  conexion = mysql.connector.connect(
4      host="localhost",
5      user="root",
6      password="Lxy20050704.",
7      database="coches_db"
8  )
9
10 cursor = conexion.cursor()
11 cursor.execute("SELECT * FROM coches")
12 datos = cursor.fetchall()
13
14 print("ID  MARCA  MODELO  COLOR  KM  PRECIO")
15 print("-----")
16
17 for coche in datos:
18     print(f"{coche[0]:<4}{coche[1]:<12}{coche[2]:<12}{coche[3]:<10}{coche[4]:<8}{coche[5]:<10}")
19
20 cursor.close()
21 conexion.close()
22

```

Reto 8:

Crear ruta

```

ub@ub:~/reto-ubuntu-python$ mkdir -p mongo_db

```

Crear un contenedor MongoDB

```

docker run -d \
  --name mongo_coches \
  -p 27017:27017 \
  -v ~/reto-ubuntu-python/mongo_db:/data/db \
  -e MONGO_INITDB_ROOT_USERNAME=root \
  -e MONGO_INITDB_ROOT_PASSWORD=Lxy20050704. \
  mongo:4.4

```

```
(venv) ub@ub:~/reto-ubuntu-python$ docker run -d \
  --name mongo_coches \
  -p 27017:27017 \
  -v ~/reto-ubuntu-python/mongo_db:/data/db \
  -e MONGO_INITDB_ROOT_USERNAME=root \
  -e MONGO_INITDB_ROOT_PASSWORD=Lxy20050704. \
  mongo:4.4
Unable to find image 'mongo:4.4' locally
4.4: Pulling from library/mongo
d4c3c94e5e10: Pull complete
bca5893fe8bd: Pull complete
35ec036951f8: Pull complete
ddb77a597b02: Pull complete
7ab9eb5a4d9d: Pull complete
a6c1ba219414: Pull complete
83b651df5384: Pull complete
e233f2d1b360: Pull complete
Digest: sha256:52c42cbab240b3c5b1748582cc13ef46d521ddacae002bbbda645cebed270ec0
Status: Downloaded newer image for mongo:4.4
aba9ca933d2f52978b788a2021240c57c90a29f26a853f4a9f74bce8617a1ea2
```

Acceder a MongoDB

- `docker exec -it mongo_coches mongo -u root -p Lxy20050704.`

```
(venv) ub@ub:~/reto-ubuntu-python$ docker exec -it mongo_coches mongo -u root -p Lxy20050704.
MongoDB shell version v4.4.29
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("159bdd25-8b98-4e66-b7cd-fcb9474f5239") }
MongoDB server version: 4.4.29
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
---
The server generated these startup warnings when booting:
  2025-11-18T00:26:39.767+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://
dochub.mongodb.org/core/prodnotes-filesystem
> |
```

Crear la base de datos y verificar

- `use coches_mongo`
`db.cars.insertMany([`
`{ marca: "Toyota", modelo: "Corolla", color: "Rojo", km: 25000, precio: 15000 },`
`{ marca: "Honda", modelo: "Civic", color: "Azul", km: 30000, precio: 18000 },`
`{ marca: "Ford", modelo: "Focus", color: "Blanco", km: 40000, precio: 17000 }`
`])`

```
> use coches_mongo
switched to db coches_mongo
> db.coches.insertMany([
... { marca: "Toyota", modelo: "Corolla", color: "Rojo", km: 25000, precio: 15000 },
... { marca: "Honda", modelo: "Civic", color: "Azul", km: 30000, precio: 18000 },
... { marca: "Ford", modelo: "Focus", color: "Blanco", km: 40000, precio: 17000 }
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("691bbf7d907faba9bf37cbab"),
    ObjectId("691bbf7d907faba9bf37cbac"),
    ObjectId("691bbf7d907faba9bf37cbad")
  ]
}
```

- db.coches.find().pretty()

```
> db.coches.find().pretty()
{
  "_id" : ObjectId("691bbf7d907faba9bf37cbab"),
  "marca" : "Toyota",
  "modelo" : "Corolla",
  "color" : "Rojo",
  "km" : 25000,
  "precio" : 15000
}
{
  "_id" : ObjectId("691bbf7d907faba9bf37cbac"),
  "marca" : "Honda",
  "modelo" : "Civic",
  "color" : "Azul",
  "km" : 30000,
  "precio" : 18000
}
{
  "_id" : ObjectId("691bbf7d907faba9bf37cbad"),
  "marca" : "Ford",
  "modelo" : "Focus",
  "color" : "Blanco",
  "km" : 40000,
  "precio" : 17000
}
```

Salir

```
> exit
bye
```

Crear un script para leer desde MongoDB

```
ub@ub:~/reto-ubuntu-python/bing_volume$ sudo nano r_mongo_db.py
```

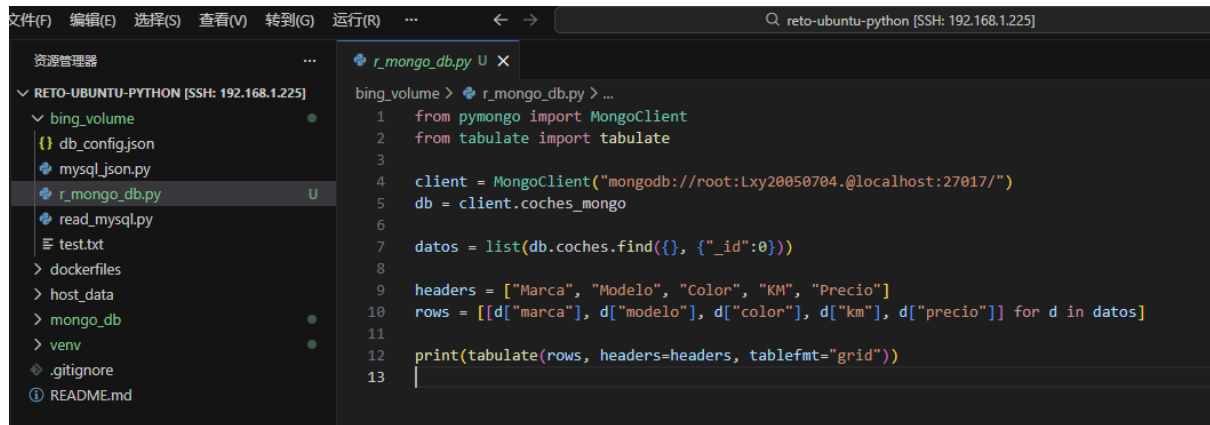
```
from pymongo import MongoClient
from tabulate import tabulate

client = MongoClient("mongodb://root:Lxy20050704.@localhost:27017/")
db = client.coches_mongo

datos = list(db.coches.find({}, {"_id":0}))

headers = ["Marca", "Modelo", "Color", "KM", "Precio"]
rows = [[d["marca"], d["modelo"], d["color"], d["km"], d["precio"]] for d in datos]

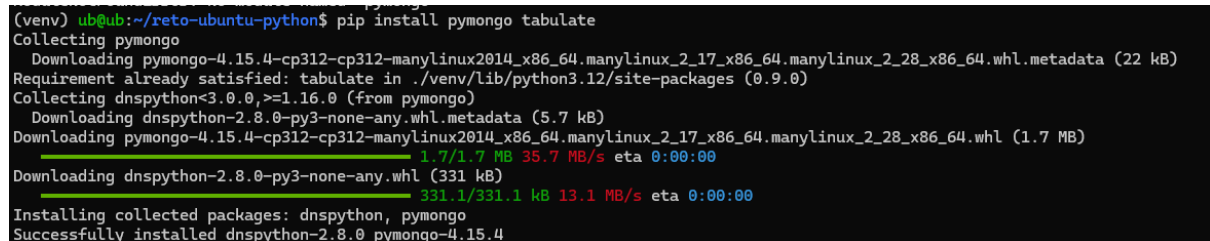
print(tabulate(rows, headers=headers, tablefmt="grid"))
```



```
bing_volume > r_mongo_db.py > ...
1 from pymongo import MongoClient
2 from tabulate import tabulate
3
4 client = MongoClient("mongodb://root:Lxy20050704.@localhost:27017/")
5 db = client.coches_mongo
6
7 datos = list(db.coches.find({}, {"_id":0}))
8
9 headers = ["Marca", "Modelo", "Color", "KM", "Precio"]
10 rows = [[d["marca"], d["modelo"], d["color"], d["km"], d["precio"]] for d in datos]
11
12 print(tabulate(rows, headers=headers, tablefmt="grid"))
13
```

Instalar pymongo y tabulate

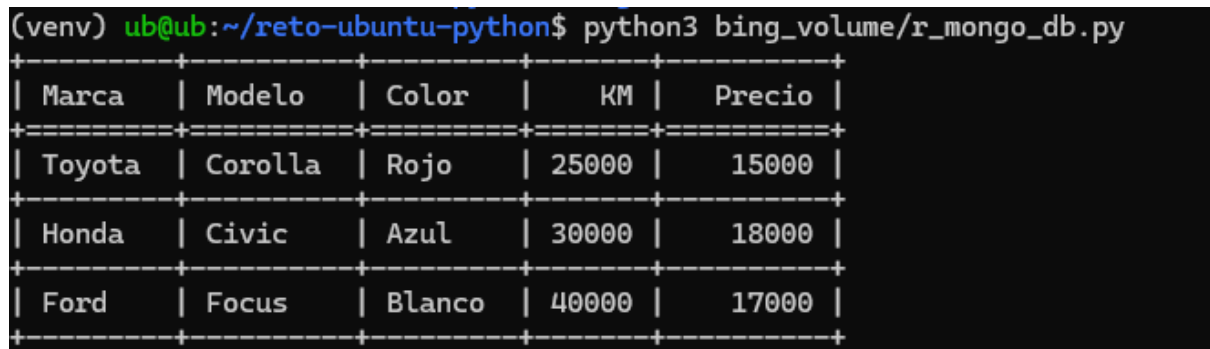
- pip install pymongo tabulate



```
(venv) ub@ub:~/reto-ubuntu-python$ pip install pymongo tabulate
Collecting pymongo
  Downloading pymongo-4.15.4-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (22 kB)
Requirement already satisfied: tabulate in ./venv/lib/python3.12/site-packages (0.9.0)
Collecting dnspython<3.0.0, >=1.16.0 (from pymongo)
  Downloading dnspython-2.8.0-py3-none-any.whl.metadata (5.7 kB)
  Downloading pymongo-4.15.4-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (1.7 MB)
    1.7/1.7 MB 35.7 MB/s eta 0:00:00
  Downloading dnspython-2.8.0-py3-none-any.whl (331 kB)
    331.1/331.1 kB 13.1 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.8.0 pymongo-4.15.4
```

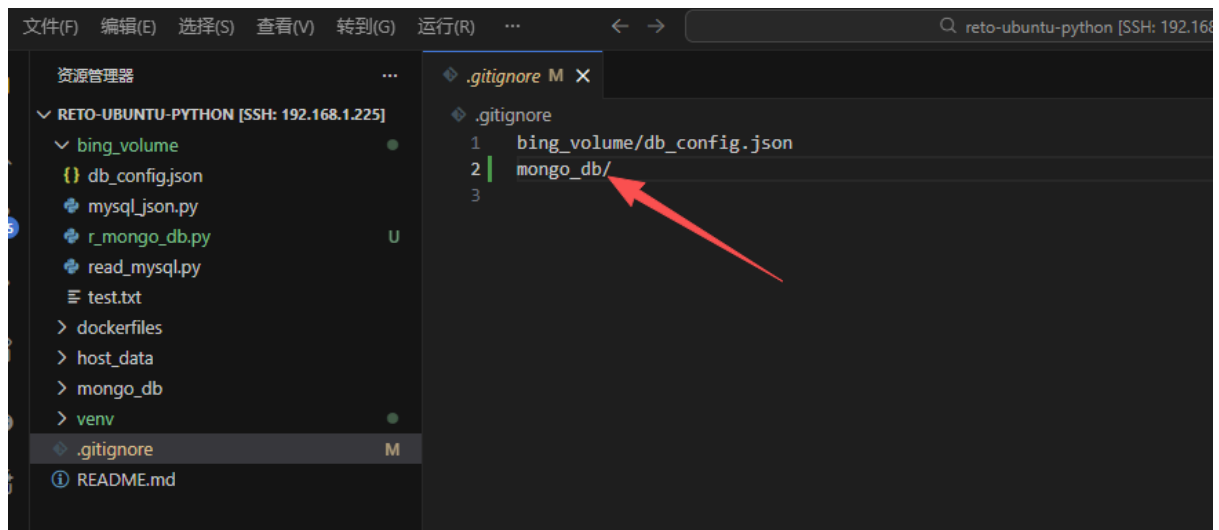
Ejecutar

- python3 bing_volume/r_mongo_db.py



```
(venv) ub@ub:~/reto-ubuntu-python$ python3 bing_volume/r_mongo_db.py
+-----+-----+-----+-----+-----+
| Marca | Modelo | Color | KM | Precio |
+=====+=====+=====+=====+=====+
| Toyota | Corolla | Rojo | 25000 | 15000 |
+-----+-----+-----+-----+-----+
| Honda | Civic | Azul | 30000 | 18000 |
+-----+-----+-----+-----+-----+
| Ford | Focus | Blanco | 40000 | 17000 |
+-----+-----+-----+-----+-----+
```

Añade la ruta `mongo_db/` al archivo .gitignore.



Confirmar en Git

```
(venv) ub@ub:~/reto-ubuntu-python$ git add .
(venv) ub@ub:~/reto-ubuntu-python$ git commit -m "Reto8 mongo_db"
```

```
(venv) ub@ub:~/reto-ubuntu-python$ git push
Username for 'https://github.com': lxin0704-12138
Password for 'https://lxin0704-12138@github.com':
Enumerating objects: 655, done.
Counting objects: 100% (655/655), done.
Delta compression using up to 6 threads
Compressing objects: 100% (641/641), done.
Writing objects: 100% (648/648), 2.68 MiB | 2.20 MiB/s, done.
Total 648 (delta 170), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (170/170), completed with 1 local object.
To https://github.com/lxin0704-12138/reto-docker-git.git
93fd89d..8cdd7ee main -> main
```

Reto 9:

Inicie sesión de Docker Hub

- docker login

```
(venv) ub@ub:~/reto-ubuntu-python$ docker login
Authenticating with existing credentials... [Username: lxin0704]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
```

Etiqueta la imagen

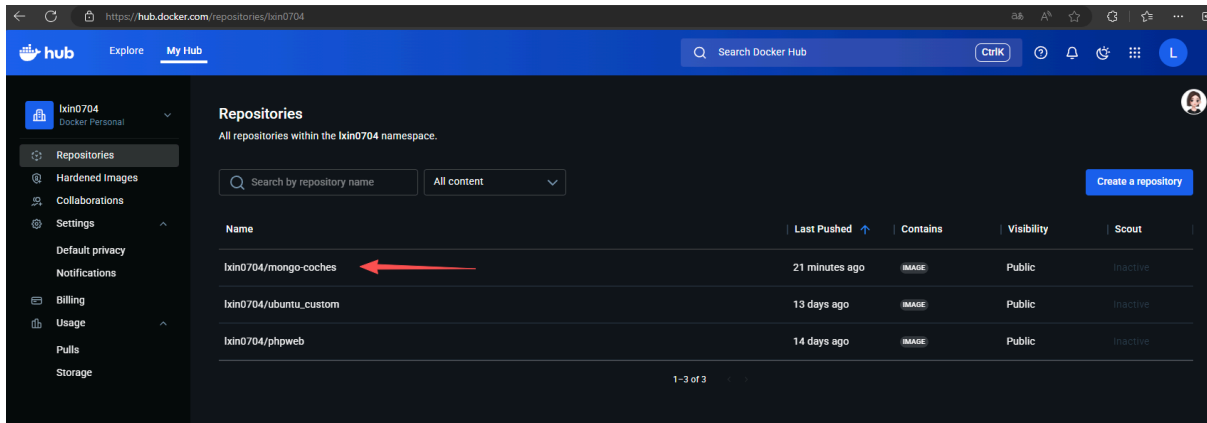
- docker tag mongo:latest lxin0704/mongo-coches:1.0

```
(venv) ub@ub:~/reto-ubuntu-python$ docker tag mongo:latest lxin0704/mongo-coches:1.0
```

Enviar la imagen a Docker Hub

- docker push lxin0704/mongo-coches:1.0

```
(venv) ub@ub:~/reto-ubuntu-python$ docker push lxin0704/mongo-coches:1.0
The push refers to repository [docker.io/lxin0704/mongo-coches]
04d815887f6a: Pushed
21b6d0e8837f: Pushed
1de80e04bc48: Pushed
346b4c6035a4: Pushed
1375c2d7e085: Pushed
afa3ab502591: Pushed
51385384509d: Pushed
e8bce0aabd68: Pushed
1.0: digest: sha256:7e049eb67f1f7ad81e3fc2d5de5e4dbf1ba8b496323385067c3f12dc1f0495fc size: 1993
```



Arrastrar y ejecutar en otra máquina virtual

- docker pull lxin0704/mongo-coches:1.0

```
ub@ub:~$ sudo docker pull lxin0704/mongo-coches:1.0
1.0: Pulling from lxin0704/mongo-coches
46a33c43d8ef: Pull complete
d97335354a5b: Pull complete
a801c4a80002: Pull complete
02de03a7213b: Pull complete
eef62aa7c052: Pull complete
9d906841bd13: Pull complete
a0ca46313493: Pull complete
fdc0018776b0: Pull complete
Digest: sha256:7e049eb67f1f7ad81e3fc2d5de5e4dbf1ba8b496323385067c3f12dc1f0495fc
Status: Downloaded newer image for lxin0704/mongo-coches:1.0
docker.io/lxin0704/mongo-coches:1.0
```

- docker run -d --name mongo_parrot -p 27017:27017 lxin0704/mongo-coches:1.0

```
ub@ub:~$ sudo docker run -d --name mongo_parrot -p 27017:27017 lxin0704/mongo-coches:1.0
7df29042678a1333c8371b9b6a19fe863d61a5d791101b88430c4d9adc522089
```

- docker ps

```
ub@ub:~$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
7df29042678a   lxin0704/mongo-coches:1.0          "docker-entrypoint.s..." 35 seconds ago Up 35 seconds 0.0.0:27017->27017/tcp, [::]:27017->27017/tcp
mongo_parrot
```