

Utilização de Myql e Mongo com Docker

Instalar o Docker Desktop

<https://www.docker.com/products/docker-desktop/>

Descarregar ficheiro zipado e descomprimi-lo debaixo da pasta c:\docker

**Criar a seguinte estrutura de pastas debaixo da pasta que armazena o docker
(c:\docker\dbdata no exemplo)**

Name	Date modified	Type	Size
mongo_files	5/29/2025 10:39 AM	File folder	
mysql_files	5/29/2025 10:32 AM	File folder	
src	5/26/2025 6:29 PM	File folder	
docker-compose.yml	5/29/2025 8:43 AM	Yaml Source File	2 KB
Dockerfile	5/28/2025 6:04 PM	File	1 KB

As pastas mongo_files e mysql_files são opcionais e armazenam ficheiros no disco local que, assim que são ali colocados, são automaticamente copiados (sincronizados) para as pastas internas do Docker (para os containers do mongo e mysql respetivamente). A pasta scr é onde podemos colocar ficheiros php que são lidos pelo apache (equivale a pasta htdocs do XAMPP). Os ficheiros Dockerfile e docker-compose.yml contém a informação para criar os containers. Depois do Docker estar criado vão aparecer mais duas pastas: Mysql_data e mongo_data onde são armazenadas as bases de dados Mysql e mongod.

Breve explicação do ficheiro docker-compose.yml (fragmento do conteúdo na imagem)

```

mysql:
  image: mysql:latest
  restart: no
  environment:
    MYSQL_ROOT_PASSWORD: 'root'
    MYSQL_DEFAULT_AUTHENTICATION_PLUGIN: n
  volumes:
    - ./mysql_data:/var/lib/mysql
    - ./mysql_files:/var/lib/mysql-files/
  ports:
    - "3306:3306"

] phpmyadmin:
  image: phpmyadmin:latest
  restart: always
  ports:
    - 9001:80
  environment:
    PMA_ARBITRARY: 1
  depends_on:
    - mysql

mongodb:
  image: mongo:latest
  restart: always
  ports:
    - "27017:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: root
  volumes:
    - ./mongo_data:/data/db
    - ./mongo_files:/tmp/import

```

Cada Serviço é um container, consideram-se no exemplo 4 containers: php, Mysql, Mongo e PhpMyadmin. DependOn indica que o serviço apenas pode “arrancar” depois de os outros arrancarem. Ou seja, por exemplo, o phpmyadmin só pode arrancar depois do Mysql.

Os volumes são importantes pois fazem o mapeamento entre o disco local e a estrutura de ficheiros interna. Por exemplo, a pasta Mysql_data local (debaixo de dbdata) sincroniza com a pasta /var/lib/mysql que está dentro do container. Ou seja, sempre que informação é escrita em /var/lib/Mysql (pasta onde seriam guardadas as bases de dados) ela é movida para a pasta Mysql_data do disco local. Não teria de ser assim (aliás esta solução não é recomendada em ambiente de produção), poderia ter sido (em vez de - ./Mysql_data: /var/lib/mysql

```

  - Mysql_data: /var/lib/mysql
volumes:
Mysql_data:

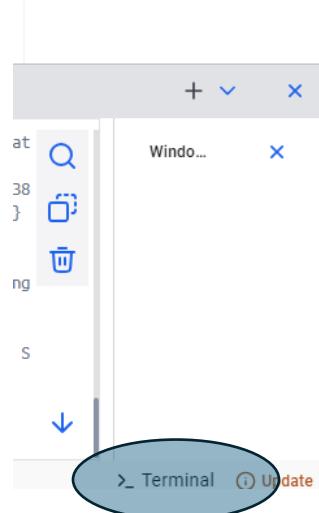
```

Neste cenário os dados são guardados internamente na pasta /var/lib/Mysql dentro do container, o que é mais seguro.

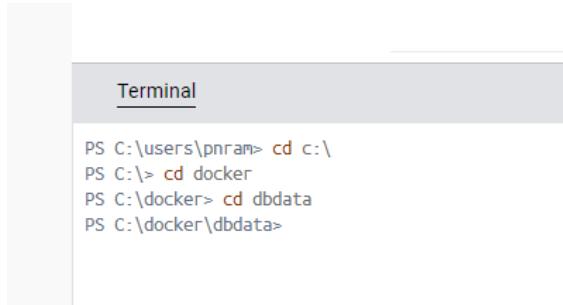
A pasta Mysql_files local (debaixo de dbdata) sincroniza com a pasta /var/lib/Mysql-files, que é a pasta onde por omissão o container vai procurar scripts de Mysql.

O resto do ficheiro é autoexplicativo (senhas, portos, nome de pastas, indicação para reinício automático em caso de falha, etc.).

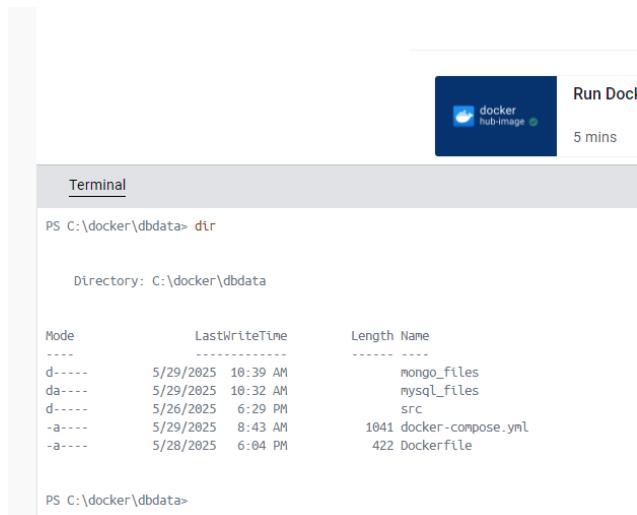
Para abrir o terminal e executar comandos no Docker:



Navegar até à pasta onde está a configuração do Docker:



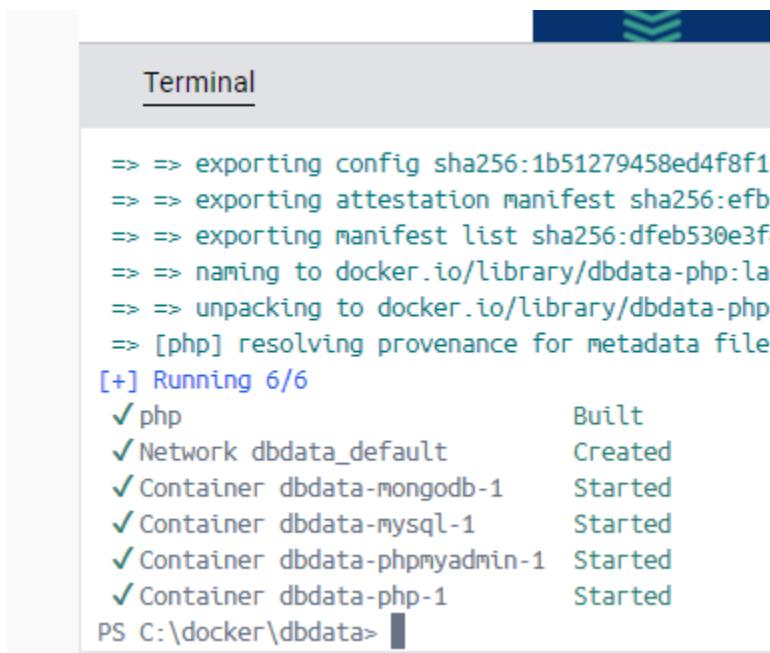
Confirmar que a pasta está completa:



Instalar o Docker (docker-compose up -d)



```
PS C:\docker\dbdata> docker-compose up -d
time="2025-05-29T12:14:11+01:00" level=warning msg="C:\\\\docker\\\\dbdata\\\\docker-
[+] Running 0/3
- mysql Pulling
- mongodb Pulling
- phpmyadmin Pulling
```

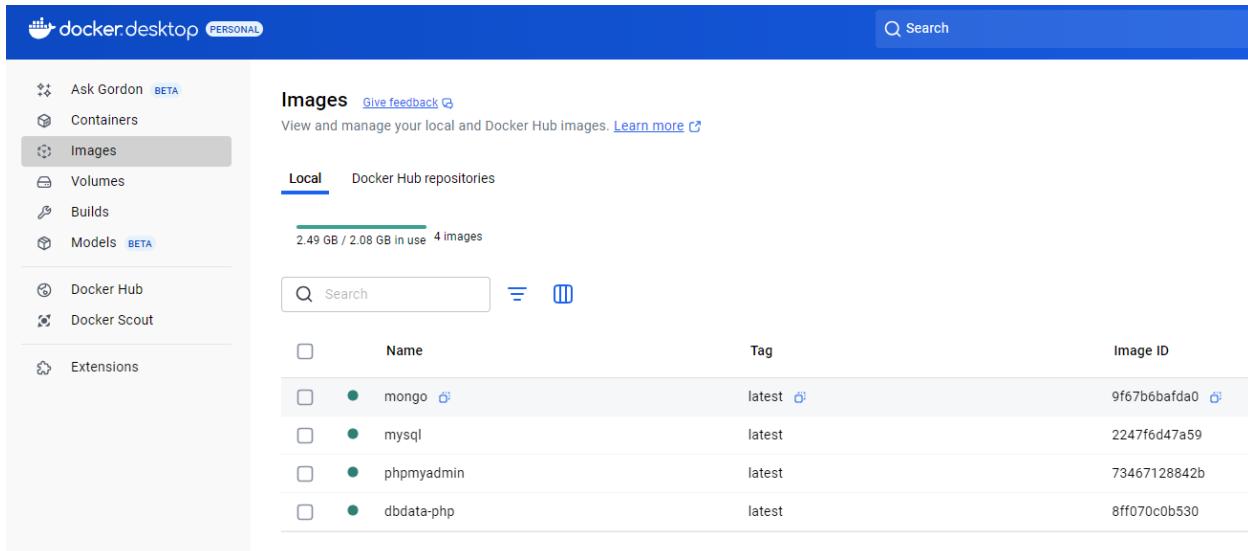


```
=> => exporting config sha256:1b51279458ed4f8f16
=> => exporting attestation manifest sha256:efb5
=> => exporting manifest list sha256:dfeb530e3f4
=> => naming to docker.io/library/dbdata-php:lat
=> => unpacking to docker.io/library/dbdata-php:
=> [php] resolving provenance for metadata file
[+] Running 6/6
✓ php                                Built
✓ Network dbdata_default              Created
✓ Container dbdata-mongodb-1          Started
✓ Container dbdata-mysql-1            Started
✓ Container dbdata-phpmyadmin-1       Started
✓ Container dbdata-php-1              Started
PS C:\docker\dbdata>
```

O terminal pode ser o normal do Windows, não tem de ser aberto dentro do Docker. É o mesmo terminal.

Depois de “instalada” a pasta dbdata (docker-compose up)

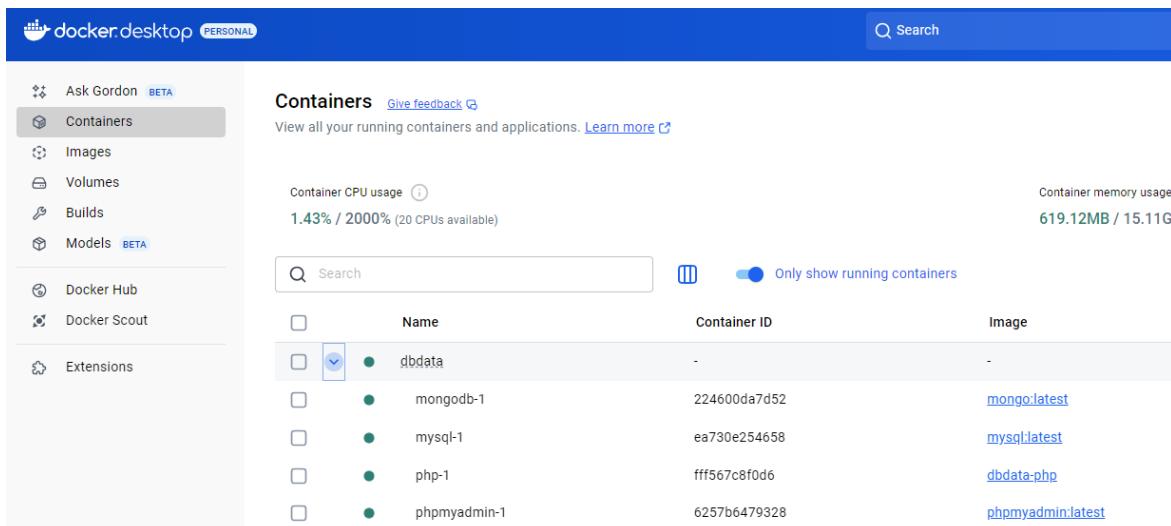
Foram Criadas 4 imagens, equivale aos 4 serviços. A imagem é uma espécie de modelo com informação para executar uma aplicação criada a partir de um dockerfile



The screenshot shows the Docker Desktop interface with the "Images" tab selected. The left sidebar includes options like Ask Gordon, Containers, Images (which is selected), Volumes, Builds, Models, Docker Hub, Docker Scout, and Extensions. The main area shows a "Local" repository with 4 images. A search bar and filter icons are at the top of the list. The table below lists the images:

	Name	Tag	Image ID
<input type="checkbox"/>	mongo	latest	9f67b6bafda0
<input type="checkbox"/>	mysql	latest	2247f6d47a59
<input type="checkbox"/>	phpmyadmin	latest	73467128842b
<input type="checkbox"/>	dbdata-php	latest	8ff070c0b530

E 4 containers, que são as instâncias executáveis das imagens. Enquanto as imagens são apenas de leitura, num container são efetuadas operações de escrita (ficam lá armazenados os dados e ficheiros. Eles são criados, iniciados, parados, reiniciados e destruídos. Se um container estiver com problemas ele pode ser criado a partir da mesma imagem, mas os dados são perdidos (daí ser fundamental as pastas Mysql_data e mongo_data no computador local).



The screenshot shows the Docker Desktop interface with the "Containers" tab selected. The left sidebar includes options like Ask Gordon, Containers (which is selected), Images, Volumes, Builds, Models, Docker Hub, Docker Scout, and Extensions. The main area shows a list of running containers. A search bar and a filter icon are at the top. The table below lists the containers:

	Name	Container ID	Image
<input checked="" type="checkbox"/>	dbdata	-	-
<input type="checkbox"/>	mongodb-1	224600da7d52	mongo:latest
<input type="checkbox"/>	mysql-1	ea730e254658	mysql:latest
<input type="checkbox"/>	php-1	fff567c8f0d6	dbdata-php
<input type="checkbox"/>	phpmyadmin-1	6257b6479328	phpmyadmin:latest

As designações, nomeadamente a componente “-1” são adicionadas automaticamente quando eles são criados pelo comando docker-compose up ou docker run <nome_da_imagem> (por exemplo, docker run Mysql). Para uma mesma imagem podem ser criados vários containers (dai o “-1”). Atenção que o nome do container contém o nome da pasta: dbdata.Mysql-1, por exemplo.

O comando docker-compose up lê o ficheiro docker-compose.yml e para cada serviço cria o container se ele não existir ainda (se já existir volta a criar caso tenha mudado alguma coisa na configuração). Com opção build (docker-compose up –build) ele força a reconstrução das imagens. Com opção d (docker-compose up –d) ele corre em background e liberta o terminal.

A combinação docker-compose up -d –build, mais habitual para iniciar um ambiente de desenvolvimento depois de fazer alterações no Dockerfile.

Para “deitar abaixo” os serviços: docker-compose down

```
PS C:\docker\dbdata> docker-compose down
time="2025-05-29T11:10:16+01:00" level=warning msg
[+] Running 5/5
✓ Container dbdata-phpmyadmin-1 Removed
✓ Container dbdata-php-1 Removed
✓ Container dbdata-mysql-1 Removed
✓ Container dbdata-mongodb-1 Removed
✓ Network dbdata_default Removed
PS C:\docker\dbdata>
```

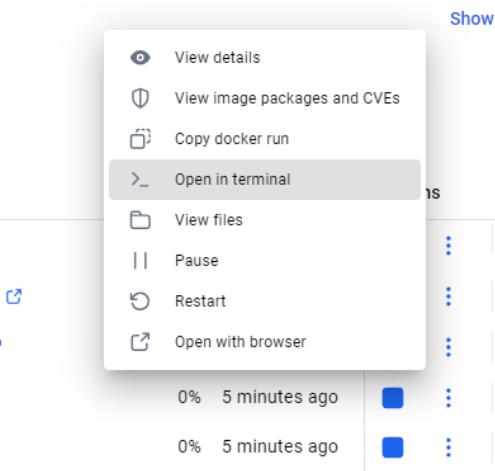
Para remover todos os containers e imagens (para, por exemplo, voltar a criar de novo), depois de assegurar que os containers estão parados:

docker system prune -a --volumes

```
PS C:\docker\dbdata> docker system prune -a --volumes
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all anonymous volumes not used by at least one container
- all images without at least one container associated to them
- all build cache

Are you sure you want to continue? [y/N]
```

Para abrir o terminal Mysql selecionar na linha do container Mysql-1 os “três pontos” e selecionar Open in terminal.



The screenshot shows the Docker interface with a context menu open over a MySQL container named "dbdata-mysql-1". The menu options include: View details, View image packages and CVEs, Copy docker run, Open in terminal (which is highlighted), View files, Pause, Restart, and Open with browser. Below the menu, the container's status is shown as 0% 5 minutes ago. The main view shows the MySQL prompt with the following output:

```
sh-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.3.0 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> Select Version();
+-----+
| Version() |
+-----+
| 9.3.0      |
+-----+
1 row in set (0.000 sec)
```

Para abrir o terminal MongoDB selecionar na linha do container MongoDB-1 os “três pontos” e selecionar Open in terminal. Ou, alternativa, no terminal: docker-compose exec mongodb mongosh --host 127.0.0.1:27017 (ou docker-compose exec mongodb mongosh)

[Containers](#) / dbdata-mongodb-1

dbdata-mongodb-1

<  224600da7d52 ⚡ mongo:latest
27017:27017 ↗

Logs Inspect Bind mounts **Exec** Files Stats

```
Enter password: ****
Current Mongosh Log ID: 683836a4a003997634d861df
Connecting to: mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=5000
Using MongoDB: 8.0.9
Using Mongosh: 2.5.0

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

-----
The server generated these startup warnings when booting
2025-05-29T10:14:12.205+00:00: For customers running the current memory allocator, we suggest changing
2025-05-29T10:14:12.205+00:00: We suggest setting the contents of sysfsFile to 0.
2025-05-29T10:14:12.205+00:00: vm.max_map_count is too low
2025-05-29T10:14:12.205+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance issues.

-----
test> db.version()
8.0.9
test>
```

Caso o docker fique corrupto e tenha de ser recriado e queremos recuperar as bases de dados

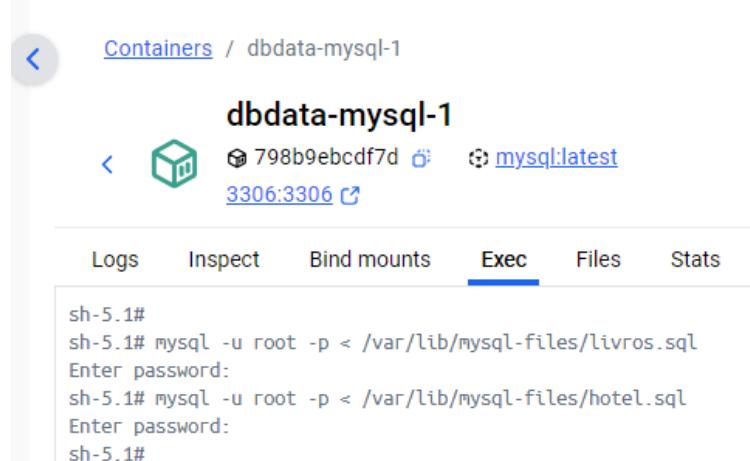
Assumindo que existe cópia das pastas locais mongo_data e mysql_data (c:\docker\copia)

1. docker-compose down -v --rmi all
2. docker system prune -a --volumes -f # -f
(cuidado, o -v apaga os volumes, inclusive os utilizadores criados)
3. apagar conteúdo de pasta c:\docker\dbdata\mysql_data
4. copiar conteúdo de pasta c:\docker\copia\ dbdata\mysql_data para pasta c:\docker\dbdata\mysql_data
5. a mesma coisa para mongo (3 e 4)
6. docker-compose up -d --build

Mysql

Importar Base de Dados Livros e Base de Dados Hotel (SCRIPT)

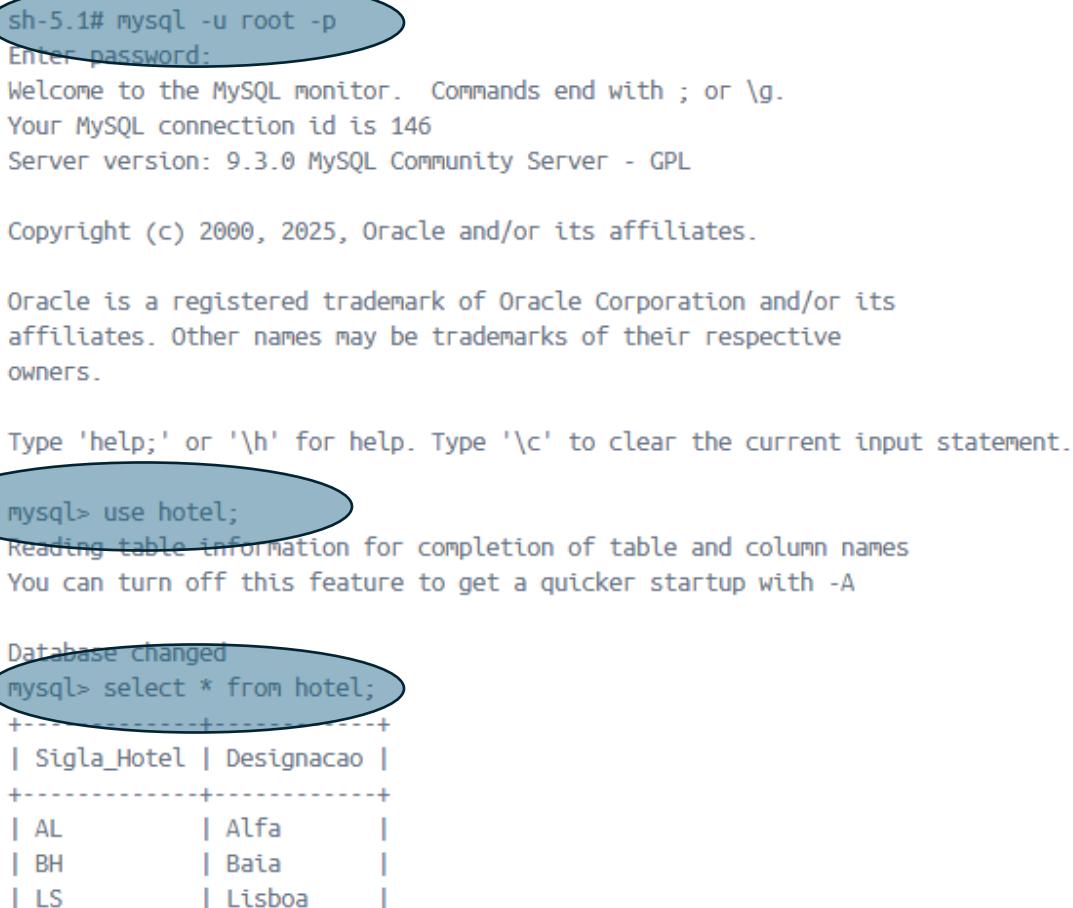
```
mysql -u root -p < /var/lib/mysql-files/livros.sql  
mysql -u root -p < /var/lib/mysql-files/hotel.sql
```



The screenshot shows a Docker container named "dbdata-mysql-1" with the ID "798b9ebcdf7d". It is running the "mysql:latest" image on port 3306. The "Exec" tab is selected in the interface. The terminal output shows the following commands being run:

```
sh-5.1# mysql -u root -p < /var/lib/mysql-files/livros.sql  
Enter password:  
sh-5.1# mysql -u root -p < /var/lib/mysql-files/hotel.sql  
Enter password:  
sh-5.1#
```

Executar Comandos SQL



The screenshot shows the MySQL monitor with the following session:

```
sh-5.1# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 146  
Server version: 9.3.0 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> use hotel;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> select * from hotel;  
+-----+-----+  
| Sigla_Hotel | Designacao |  
+-----+-----+  
| AL          | Alfa       |  
| BH          | Baia       |  
| LS          | Lisboa     |
```

Importar Base de Dados ATP (CSV)

- 1) Importar script que cria a base de dados e a tabela

```
mysql -u root -p < /var/lib/mysql-files/createatp.sql
```

- 2) Importar o ficheiro com registos csv para a tabela

```
mysql -u root -p -D atm -e 'LOAD DATA INFILE "/var/lib/mysql-files/atpplayers.csv" INTO TABLE atpplayers FIELDS TERMINATED BY "," ENCLOSED BY "\"';'
```

[Containers](#) / dbdata-mysql-1

dbdata-mysql-1
798b9ebcdfd 3306:3306 ↗ mysql:latest

Logs Inspect Bind mounts **Exec** Files Stats

Docker Debug brings the tools you need to debug your container with one click.
Requires a paid Docker subscription. [Learn more.](#)

```
sh-5.1# mysql -u root -p < /var/lib/mysql-files/createatp.sql
Enter password:
sh-5.1# mysql -u root -p -D atm -e 'LOAD DATA INFILE "/var/lib/mysql-files/atpplayers.csv" INTO TABLE atpplayers FIELDS TERMINATED BY "," ENCLOSED BY "\"';'
Enter password:
sh-5.1#
```

PhpMyAdmin

The screenshot shows the PhpMyAdmin interface for a MySQL server. The top navigation bar includes links for Bookmarks, Informação geral, Colibri V3, Login Fenix+, moodle24, BOOX Drop, and Imported. The main menu on the left lists databases: New, atm, hotel, information_schema, livro, mysql, performance_schema, and sys. The current view is the 'General settings' section, which contains fields for 'Change password', 'Server connection collation' set to utf8mb4_unicode_ci, and a 'More settings' link. Below this is the 'Appearance settings' section, which includes a 'Language' dropdown set to English and a 'Theme' dropdown set to pmahomme, with a 'View all' button. The title bar indicates the connection is to localhost:9001.

Ligaçāo de Python a Mysql

```
import pymysql
from mysql.connector import Error
import mysql.connector as mariadb

usermysql="root"
passmysql="root"
hostmysql="localhost"
database="hotel"

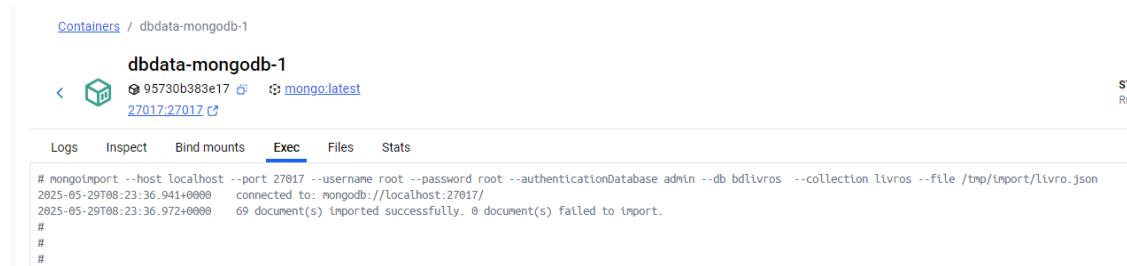
try:
    connection = mariadb.connect(host=hostmysql, user=usermysql,
passwd=passmysql, db=database, connect_timeout=1000, autocommit=True)
    print("Connected to MySQL DOCKET ATP mp")
except Error as e:
    print("Error while connecting to MySQL Docker DBLivros", e)

sql = "SELECT count(*) as total from hotel;"
try:
    cursor = connection.cursor()
    cursor.execute(sql)
    records = cursor.fetchall()
    for row in records:
        print(row[0])
except:
    print ("Error: unable to SELECT count(*) as total from hotel")
```

Mongo DB

Importar Base de Dados Livros (JSON)

```
mongoimport --host localhost --port 27017 --username root --password root --authenticationDatabase admin --db bdlivros --collection livros --file /tmp/import/livro.json
```



The screenshot shows the Docker Container Overview page. A container named 'dbdata-mongodb-1' is selected. It has an IP address of 95730b383e17, is running the mongo:latest image, and is mapped to port 27017. The 'Exec' tab is active, showing the command being run:

```
# mongoimport --host localhost --port 27017 --username root --password root --authenticationDatabase admin --db bdlivros --collection livros --file /tmp/import/livro.json
2025-05-29T08:23:36.941+0000  connected to: mongodb://localhost:27017/
2025-05-29T08:23:36.972+0000  69 document(s) imported successfully. 0 document(s) failed to import.
#
#
#
```

Executar Comandos



The screenshot shows the MongoDB shell (mongosh) running in a terminal. It starts by connecting to the database:

```
# mongosh -u root -p
Enter password: ****
Current Mongosh Log ID: 68381a7729d43755e2d861df
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB:    8.0.9
Using Mongosh:   2.5.0
```

It then lists the available databases:

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

The server generated these startup warnings when booting

```
2025-05-28T20:58:11.729+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-05-28T20:58:11.729+00:00: We suggest setting the contents of sysfsFile to 0.
2025-05-28T20:58:11.729+00:00: vm.max_map_count is too low
2025-05-28T20:58:11.729+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
```

The user then switches to the 'bdlivros' database:

```
test> use bdlivros
switched to db bdlivros
bdlivros> db.livros.find()
[
  {
    _id: ObjectId('68381988ade90a671628b3d7'),
    IDLIVRO: 1,
```

O servidor mongod é iniciado automaticamente quando o container é *iniciado*.

Compass

O Compass é executado no Windows

The screenshot shows the MongoDB Compass application interface. At the top, there is a "New Connection" dialog box with the following details:

- URI:** `mongodb://root:*****@localhost:27017/` (highlighted with a blue oval)
- Name:** (empty input field)
- Color:** `No Color` (dropdown menu)
- Favorite this connection:** (checkbox)
- Advanced Connection Options:** (button) - Submenu tabs: General, **Authentication**, TLS/SSL, Proxy/SSH, In-Use Encryption, Advanced.
- Authentication Method:** `Username/Password` (highlighted with a green border), `OIDC`, `X.509`, `Kerberos`, `LDAP`, `AWS IAM`.
- Username:** `root` (highlighted with a blue oval)
- Password:** `....` (highlighted with a blue oval)
- Authentication Database:** (input field)
- Buttons:** `Cancel`, `Save`, `Close`

Below the dialog, the main Compass interface is visible:

- Toolbar:** MongoDB Compass, Connections, Edit, View, Help
- Header:** Compass, **My Queries**
- Connections:** **CONNECTIONS (1)** (highlighted with a blue oval)
 - Search connections:** (input field)
 - Operations:** `x`, `+`, `...`
 - List:** `localhost:27017` (selected, expanded)
 - `admin`
 - `bdlivros` (expanded)
 - `livros`
 - `config`
 - `local`

Importar Base de Dados Atp (CSV)

```
mongoimport --host localhost --port 27017 --username root --
password root --authenticationDatabase admin --db atp --collection
atpplayers --type csv --headerline --file
/tmp/import/atpplayers.csv

# mongoimport --host localhost --port 27017 --username root --password root --authenticationDatabase admin
2025-05-29T08:54:36.505+0000 connected to: mongodb://localhost:27017/
2025-05-29T08:54:39.504+0000 [###.....] atp.atpplayers 30.9MB/219MB (14.1%)
2025-05-29T08:54:42.504+0000 [#####.....] atp.atpplayers 53.4MB/219MB (24.4%)
2025-05-29T08:54:45.504+0000 [#####....] atp.atpplayers 73.5MB/219MB (33.6%)
2025-05-29T08:54:48.504+0000 [#####....] atp.atpplayers 95.7MB/219MB (43.7%)
2025-05-29T08:54:51.504+0000 [#####....] atp.atpplayers 121MB/219MB (55.1%)
2025-05-29T08:54:54.504+0000 [#####....] atp.atpplayers 148MB/219MB (67.4%)
2025-05-29T08:54:57.504+0000 [#####....] atp.atpplayers 174MB/219MB (79.4%)
2025-05-29T08:55:00.504+0000 [#####....] atp.atpplayers 202MB/219MB (92.2%)
2025-05-29T08:55:02.458+0000 [#####....] atp.atpplayers 219MB/219MB (100.0%)
2025-05-29T08:55:02.460+0000 1308835 document(s) imported successfully. 0 document(s) failed to import.
#
```

Ligaçāo de Python a Mongo

```
from pymongo import MongoClient

from pymongo.errors import ConnectionFailure, OperationFailure

db_name = 'atp'
collection_name = 'atpplayers'
client = MongoClient('localhost', 27017, username='root',
password='root', authSource='admin')
#client =
MongoClient('localhost:27017,localhost:27018,localhost:27019',
replicaSet='ReplicaLivros', username='root', password='root',
authSource='admin')
try:
    db = client[db_name]
    collection = db[collection_name]
    num_documentos = collection.count_documents({})
    print(f"Número de documentos: {num_documentos}")

except Exception as e:
    print(f"Ocorreu um erro inesperado: {e}")
finally:
    client.close()
    print("Conexão ao MongoDB fechada.")
```

Rélicas

Para ter vários servidores tenho de ter docker-compose.yml para o réplica set (ver exemplo na pasta)- Depois de construir os containers, abro um terminal numa réplica mongo e

```
use admin;
var config = {
  _id : "Replica",
  members : [
    {_id : 0, host : "mongodb1:27017"},
    {_id : 1, host : "mongodb2:27017"},
    {_id : 2, host : "mongodb3:27017"} ] };
rs.initiate(config);
```

Uma configuração pode ter um mongo standalone e rélicas, tudo no mesmo ficheiro.

```
version: '3.9'
services:
  ...
  mongodb_standalone:
    image: mongo:latest
    container_name: mongodb-standalone
    restart: always
    ports:
      - "27017:27017"
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: root
    volumes:
      - ./mongo-data-standalone:/data/db
      - ./mongostandalone_files:/tmp/import
    networks:
      - app_network

  mongodb1:
    image: mongo:latest
    container_name: mongodb1
    ports:
      - "27018:27017"
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: root
    volumes:
      - mongo1_data:/data/db
      - ./mongoreplica_files:/tmp/import
      - ./mongo-keyfile:/data/db/mongo-keyfile
    command: mongod --replSet Replica --bind_ip_all --keyFile
    /data/db/mongo-keyfile
    networks:
```

```

- app_network

mongodb2:
  image: mongo:latest
  container_name: mongodb2
  ports:
    - "27019:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: root
  volumes:
    - mongo2_data:/data/db
    - ./mongoreplica_files:/tmp/import
    - ./mongo-keyfile:/data/db/mongo-keyfile
  command: mongod --replSet Replica --bind_ip_all --keyFile
  /data/db/mongo-keyfile
  networks:
    - app_network

mongodb3:
  image: mongo:latest
  container_name: mongodb3
  ports:
    - "27020:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: root
  volumes:
    - mongo3_data:/data/db
    - ./mongoreplica_files:/tmp/import
    - ./mongo-keyfile:/data/db/mongo-keyfile
  command: mongod --replSet Replica --bind_ip_all --keyFile
  /data/db/mongo-keyfile
  networks:
    - app_network

volumes:
  mongo1_data:
  mongo2_data:
  mongo3_data:

networks:
  app_network:
    driver: bridge

```