Trabajo Práctico N°2

Docker

Ejercicio 3

```
Microsoft Windows [Versión 10.0.19045.3324]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
3f4d90098f5b: Pull complete
Digest: sha256:3fbc632167424a6d997e74f52b878d7cc478225cffac6bc977eedfe51c7f4e79
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

What's Next?
  View summary of image vulnerabilities and recommendations □  docker scout quickview busybox

C:\Users\Usuario>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
busybox      latest    a416a98b71e2   3 weeks ago   4.26MB

C:\Users\Usuario>docker run busybox
```

Ejercicio 4

Cuando se ejecuta el comando `docker run busybox`, se intenta crear y ejecutar un contenedor utilizando la imagen de BusyBox. Sin embargo, en este caso, no se obtiene ningún resultado visible porque el contenedor se inicia, realiza una tarea muy breve y luego se detiene automáticamente. Esto sucede debido a la naturaleza efímera de los contenedores.

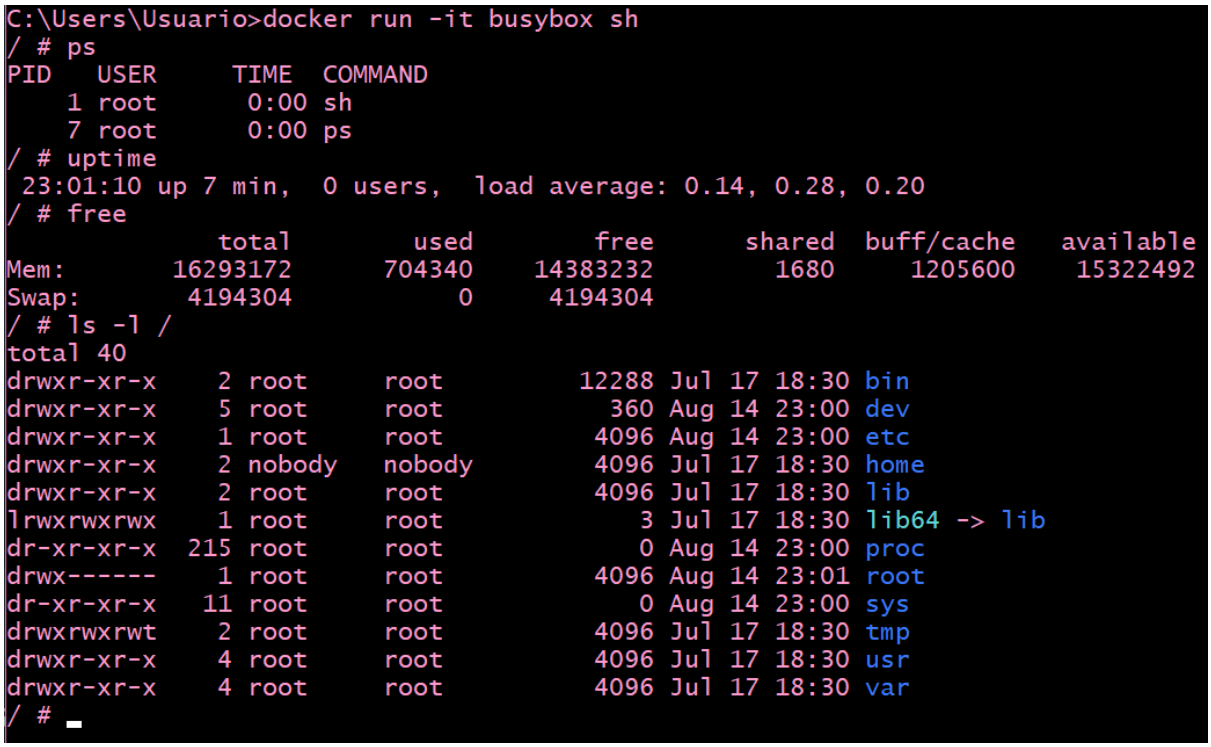
Cuando se ejecuta `docker run busybox`, el contenedor se inicia y ejecuta el programa predeterminado de BusyBox, que en este caso es simplemente un shell interactivo. Sin embargo, como no se está interactuando directamente con el contenedor a través de una terminal, el shell se inicia y se detiene inmediatamente después, lo que hace que parezca que no se obtuvo ningún resultado.

```
C:\Users\Usuario>docker run busybox echo "Hola Mundo"
Hola Mundo

C:\Users\Usuario>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Users\Usuario>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS     NAMES
206b4ec1d611   busybox   "echo 'Hola Mundo'"      15 seconds ago   Exited (0) 15 seconds ago             sharp_sanderson
8daa8aec99dd   busybox   "sh"                     2 minutes ago    Exited (0) 2 minutes ago              nifty_grothendieck
```

En esta salida:

- `CONTAINER ID` es el ID único del contenedor.
- `IMAGE` es la imagen utilizada para crear el contenedor.
- `COMMAND` es el comando que se ejecutó dentro del contenedor.
- `CREATED` indica cuándo se creó el contenedor.
- `STATUS` muestra el estado actual del contenedor. En este caso, "Exited (0)" significa que el contenedor finalizó su ejecución con un código de salida 0, lo que indica que se detuvo exitosamente.
- `NAMES` es el nombre asignado automáticamente al contenedor.

Ejercicio 5



El modo interactivo (`-it`) en Docker permite interactuar directamente con el contenedor a través de una terminal. Al usar esta combinación de opciones junto con el comando `docker run`, se está indicando que se desea acceder al interior del contenedor para ejecutar comandos y ver sus resultados en tiempo real.

El comando `sh` es el intérprete de comandos de shell en sistemas Unix y Unix-like.

Ejercicio 6

```
C:\Users\Usuario>docker ps -a
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS                      PORTS     NAMES
c88dad3212a3   busybox   "sh"                 5 minutes ago    Exited (0) 38 seconds ago             youthful_edison
206b4ec1d611   busybox   "echo 'Hola Mundo'"  8 minutes ago    Exited (0) 4 minutes ago              sharp_sanderson
8daa8aec99dd   busybox   "sh"                 10 minutes ago   Exited (0) 6 minutes ago              nifty_grothendieck

C:\Users\Usuario>docker rm 206b4ec1d611
206b4ec1d611

C:\Users\Usuario>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
c88dad3212a3b2f4413a6b39e610ee69539e088769556c261ff02f1fd83e12ef
8daa8aec99dde9a763636f454879dc8f5749f44cde7146add53276fa75b34684

Total reclaimed space: 28B

C:\Users\Usuario>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Ejercicio 7

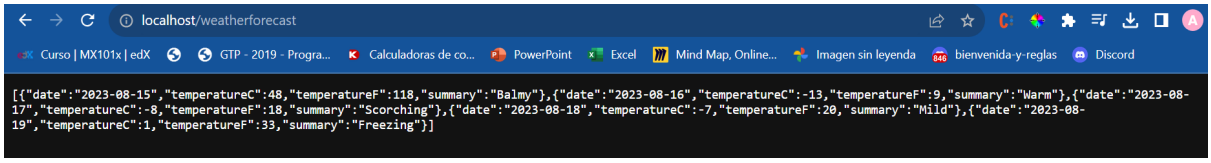
```
$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
mywebapi     latest    7a5ac4e1719f   6 days ago   216MB
```

```
$ docker run -p 80:80 mywebapi
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
```

```
Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/acc
ess-tokens/

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker tag mywebapi aaraya0/mywebapi

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker push aaraya0/mywebapi
Using default tag: latest
The push refers to repository [docker.io/aaraya0/mywebapi]
6757d714f40c: Pushed
5f70bf18a086: Pushed
8f1d52db5f25: Pushed
34801eb4d606: Pushed
726c780cb4c1: Pushed
37d2dea456a9: Pushed
f6c4eaf2bdb1: Pushed
8ce178ff9f34: Pushed
latest: digest: sha256:07dda60d2dfca0746c93cd7d0ebf97a687a3c1f51ec1a0db2d07c34c3b638b39 size: 1996
```

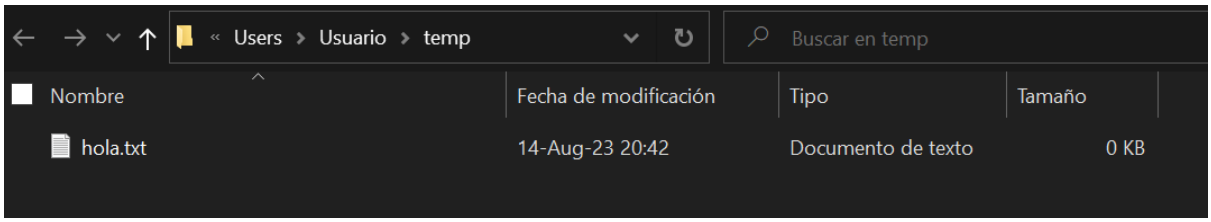
Ejercicio 8

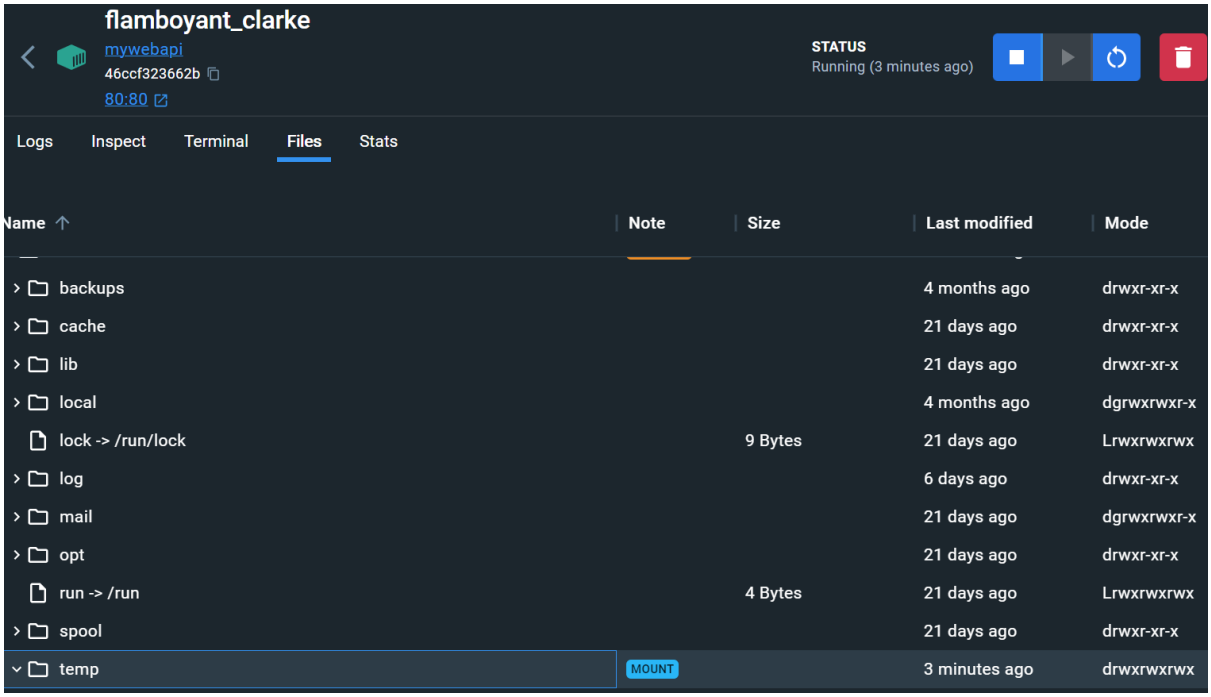
```
Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker run --name myapi -d mywebapi
a2c32b63d906a99cf2c88e37c6d234e10d969652efbb6e6d297c79fe2a87b9ed

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS
a2c32b63d906   mywebapi   "dotnet SimpleWebAPI…"   9 seconds ago   Up 8 seconds   80/tcp, 443/

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker rm myapi
Error response from daemon: You cannot remove a running container a2c32b63d906a99cf2c88e37c6d2
fe2a87b9ed. Stop the container before attempting removal or force remove

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker stop myapi
myapi

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker rm myapi
myapi

Usuario@DESKTOP-8G1HL0R MINGW64 ~/OneDrive/Escritorio/2023/Ing SW 3/SimpleWebAPI (main)
$ docker run --name myapi -d -p 80:80 -p 5254:5254 mywebapi
105721ca26636d625be6ed27dc9b4370fb16ed1fd5387d6dc6021c59f2d6b0b8
```



Ejercicio 10

```
C:\Users\Usuario>docker run -it --rm -p 80:80 -v C:\Users\Usuario\temp:/var/temp mywebapi
root@46ccf323662b:/app# ls -l /var/temp
total 0
root@46ccf323662b:/app# touch /var/temp/hola.txt
root@46ccf323662b:/app#
```

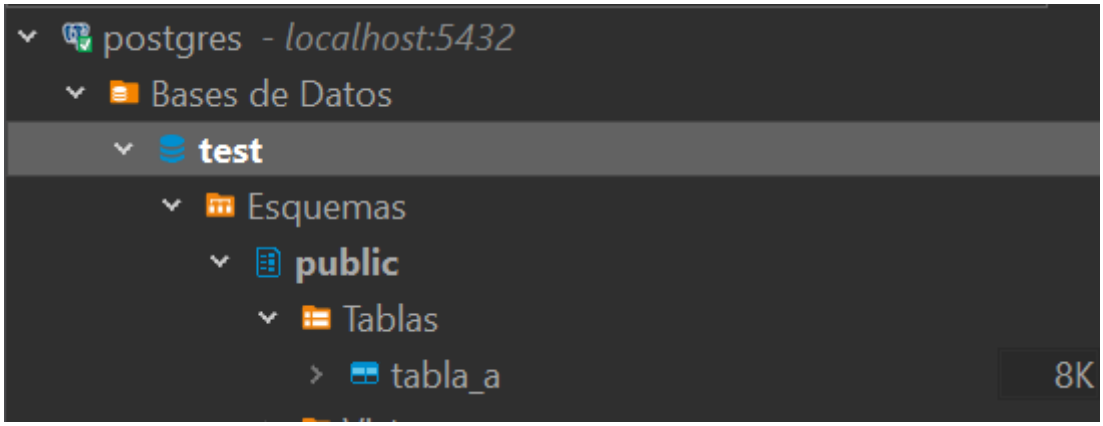


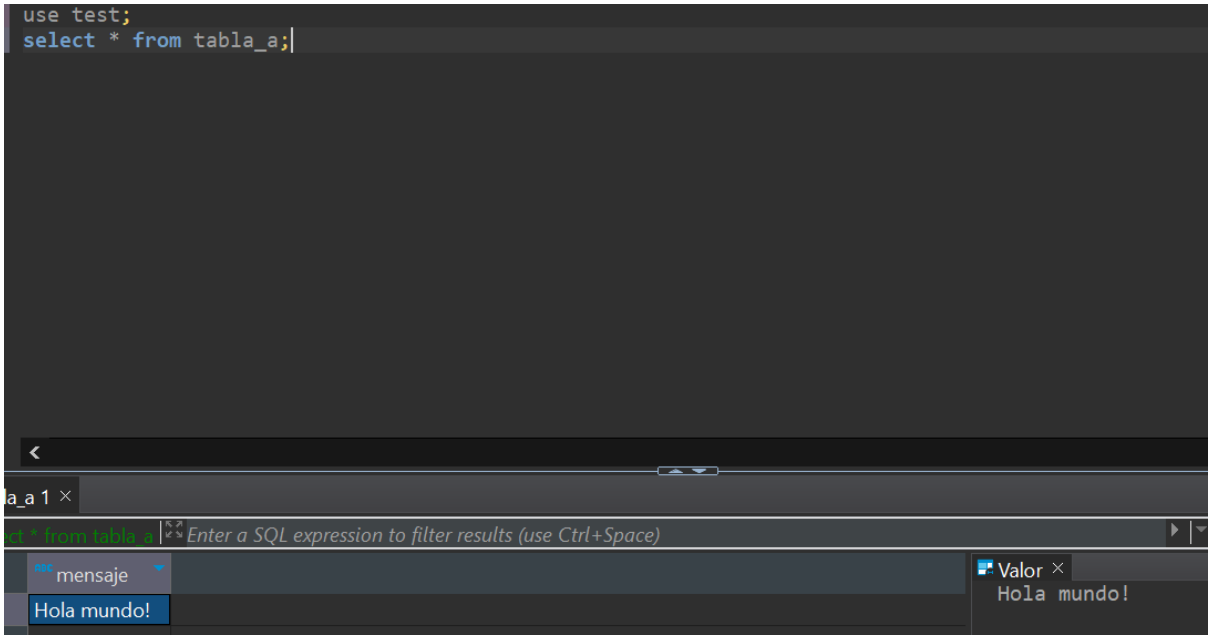


Ejercicio 11

```
C:\Users\Usuario>docker run --name my-postgres -e POSTGRES_PASSWORD=mysecretpassword -v C:\Users\Usuario\.postgres:/var/lib/postgresql/data -p 54
32:5432 -d postgres:9.4
Unable to find image 'postgres:9.4' locally
9.4: Pulling from library/postgres
619014d83c02: Pull complete
7ec0fe6664f6: Pull complete
9ca7ba8f7764: Pull complete
9e1155d037e2: Pull complete
febcfb7f8870: Pull complete
8c78c79412b5: Pull complete
5a35744405c5: Pull complete
27717922e067: Pull complete
36f0c5255550: Pull complete
dbf0a396f422: Pull complete
ec4c06ea33e5: Pull complete
e8dd33eba6d1: Pull complete
51c81b3b2c20: Pull complete
2a03dd76f5d7: Pull complete
Digest: sha256:42a7a6a647a602efa9592edd1f56359800d079b93fa52c5d92244c58ac4a2ab9
Status: Downloaded newer image for postgres:9.4
2e8c10ff239c281e66a0b826c7f1fe2be29ae9f45031983bc988f666e566c5b8
```

```
C:\Users\Usuario>docker exec -it my-postgres /bin/bash
root@2e8c10ff239c:/# psql -h localhost -U postgres
psql (9.4.26)
Type "help" for help.

postgres=#
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
(3 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \connect test
You are now connected to database "test" as user "postgres".
test=# create table tabla_a (mensaje varchar(50));
CREATE TABLE
test=# insert into tabla_a (mensaje) values('Hola mundo!');
INSERT 0 1
test=# select * from tabla_a;
   mensaje
-------------
 Hola mundo!
(1 row)

test=# \q
root@2e8c10ff239c:/# exit
exit
```





El comando `docker exec` se utiliza para ejecutar comandos dentro de un contenedor en ejecución. En este caso, se está utilizando `docker exec` para conectarse al contenedor de PostgreSQL y luego ejecutar el cliente `psql` para interactuar con la base de datos PostgreSQL directamente desde la línea de comandos. Estos comandos permiten crear y gestionar una instancia de PostgreSQL en un contenedor Docker, y brindan la capacidad de interactuar con la base de datos tanto desde la línea de comandos como desde una herramienta de IDE.

Ejercicio 12

```
C:\Users\Usuario>docker run -e MYSQL_ROOT_PASSWORD=aaraya0 -p 3307:3306 --name mysql-server -d mysql:latest
96156e684459baf3aa88872e0dddd1d084d04d3bd387a4c5599432a86154a26f

C:\Users\Usuario>docker exec -it mysql-server mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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.
```

```
mysql> create database docker_test;
Query OK, 1 row affected (0.03 sec)

mysql> use docker_test;
Database changed
mysql> create table tabla_a (mensaje varchar(50));
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> INSERT INTO tabla_a (mensaje) VALUES ('HOLA MUNDOOOO');
Query OK, 1 row affected (0.09 sec)

mysql> table tabla_a
    -> ;
+---------------+
| mensaje       |
+---------------+
| HOLA MUNDOOOO |
+---------------+
1 row in set (0.00 sec)
```