

Seminario de Actualización Dev Ops

Práctica Formativa Obligatoria N° 2

Comisión: D

Grupo: 15

Integrantes:

Arias Diego
Godoy Sergio
Ferrari Agustín

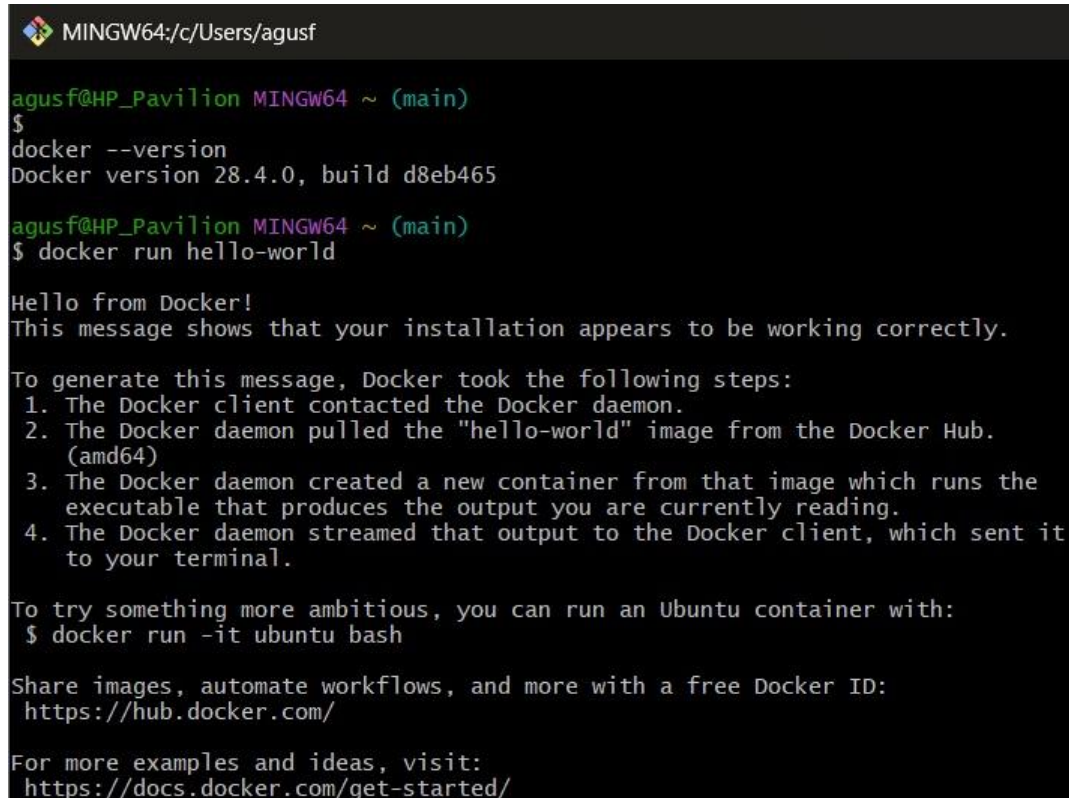
Parte I

1. Una vez instalado Docker, se creó una cuenta en Docker Hub y se instaló Docker Desktop.

Se ejecutaron los siguientes comandos para verificar la instalación de Docker:

docker --version

docker run hello-world



```
MINGW64:/c/Users/agusf

agusf@HP_Pavilion MINGW64 ~ (main)
$
docker --version
Docker version 28.4.0, build d8eb465

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

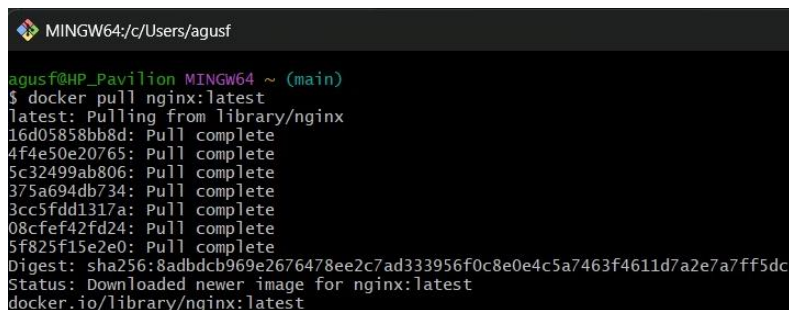
Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

2. Se buscó una imagen Nginx en Docker Hub y se descargó usando los siguientes comandos:

docker search nginx

docker pull nginx:latest



```
MINGW64:/c/Users/agusf

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker pull nginx:latest
latest: Pulling from library/nginx
16d05858bb8d: Pull complete
4f4e50e20765: Pull complete
5c32499ab806: Pull complete
375a694db734: Pull complete
3cc5fdd1317a: Pull complete
08cfef42fd24: Pull complete
5f825f15e2e0: Pull complete
Digest: sha256:8adbdc969e2676478ee2c7ad333956f0c8e0e4c5a7463f4611d7a2e7a7ff5dc
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

3. Luego se crea un contenedor con esa imagen Nginx y se comprueba con los siguientes comandos:

```
docker run -d --name webserver -p 8080:80 nginx
```

```
docker ps
```

```
agusf@HP_Pavilion MINGW64 ~ (main)
$ docker run -d --name webserver -p 8080:80 nginx
3d5b8c4488768b4a0f53ca50a779488b36a293cd103bd124dcc640bed549393

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
3d5b8c448876      nginx              "/docker-entrypoint..." 53 seconds ago      Up 52 seconds      0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
agusf@HP_Pavilion MINGW64 ~ (main)
```

El contenedor se va a comunicar a través del puerto 8080 con la computadora, lo que se verifica con el navegador en la dirección: <http://localhost:8080>



4. Se crea un contenedor MySQL con una imagen descargada de Docker Hub, utilizando los siguientes comandos:

```
docker pull mysql:8
```

```
docker run -d --name dbserver -e MYSQL_ROOT_PASSWORD=admin -p 3306:3306
```

```
mysql:8
```

```
MINGW64:/c/Users/agusf

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker pull mysql:8
8: Pulling from library/mysql
970f697f30e8: Pull complete
69718387e824: Pull complete
7a99a8dca35c: Pull complete
35a745903ff9: Pull complete
d3dc946e9b73: Pull complete
f56ec30544fc: Pull complete
8389b884d3d6: Pull complete
46d5b31c795a: Pull complete
a49b4bec6f69: Pull complete
806f49275cbf: Pull complete
Digest: sha256:5367102acfefeeaa47eb0eb57c8d4f8b96c8c14004859131eac9bbfaa62f81e34
Status: Downloaded newer image for mysql:8
docker.io/library/mysql:8

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker run -d --name dbserver -e MYSQL_ROOT_PASSWORD=admin -p 3306:3306 mysql:8
d64907269868be4d477e358369a0e4c26a7d9494390f254527066cf28873b636

agusf@HP_Pavilion MINGW64 ~ (main)
$
```

5. Linkeamos los contenedores, usando una red brige, con los siguientes comando:

```
docker network create mynet
```

```
docker network connect mynet webserver
```

```
docker network connect mynet dbserver
```

y comprobamos con:

```
docker network inspect mynet
```

```
MINGW64:/c/Users/agusf

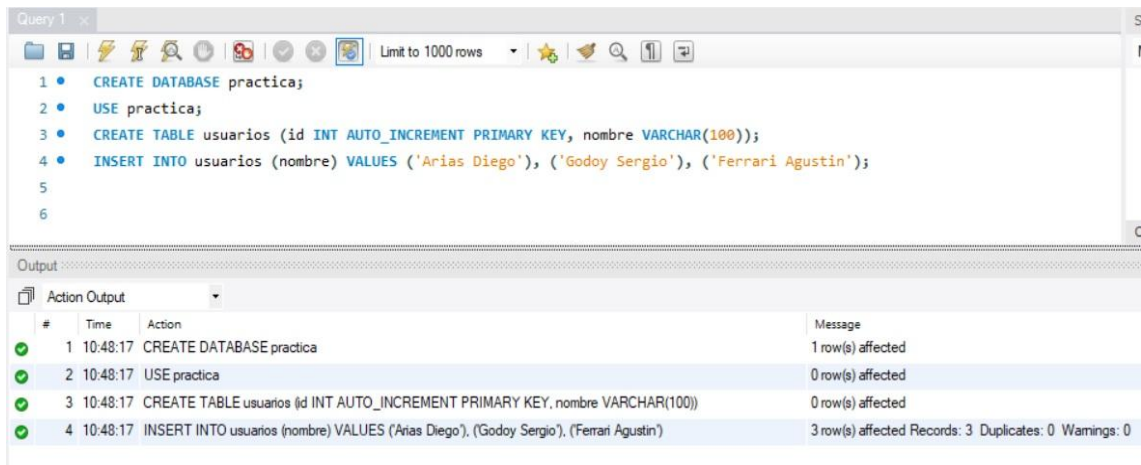
agusf@HP_Pavilion MINGW64 ~ (main)
$ docker network create mynet
ad9a666898a43924ea3ae766a60e71e9f17fd57b405fcb7444af43f0c9efdec4

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker network connect mynet webserver

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker network connect mynet dbserver

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker network inspect mynet
[
  {
    "Name": "mynet",
    "Id": "ad9a666898a43924ea3ae766a60e71e9f17fd57b405fcb7444af43f0c9efdec4",
    "Created": "2025-10-01T01:18:27.606151329Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "3d5b8c4488768b4a0f53ca50a779488b36a293cd103bd124dcc640bed549393": {
        "Name": "webserver",
        "EndpointID": "409071c73a2ea0ef827332b4739cd21cde72f2a20a90c7ae0f8e8d2602b1f720",
        "MacAddress": "a6:69:3c:dd:5a:2d",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      },
      "d64907269868be4d477e358369a0e4c26a7d9494390f254527066cf28873b636": {
        "Name": "dbserver",
        "EndpointID": "360ca4e20aedce47daefc195c7ffd0b7fc2237dd8fbff763a25d71852b3e04c",
        "MacAddress": "f6:83:ce:e1:0a:19",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      }
    },
    "Options": {
      "com.docker.network.enable_ipv4": "true",
      "com.docker.network.enable_ipv6": "false"
    },
    "Labels": {}
  }
]
```

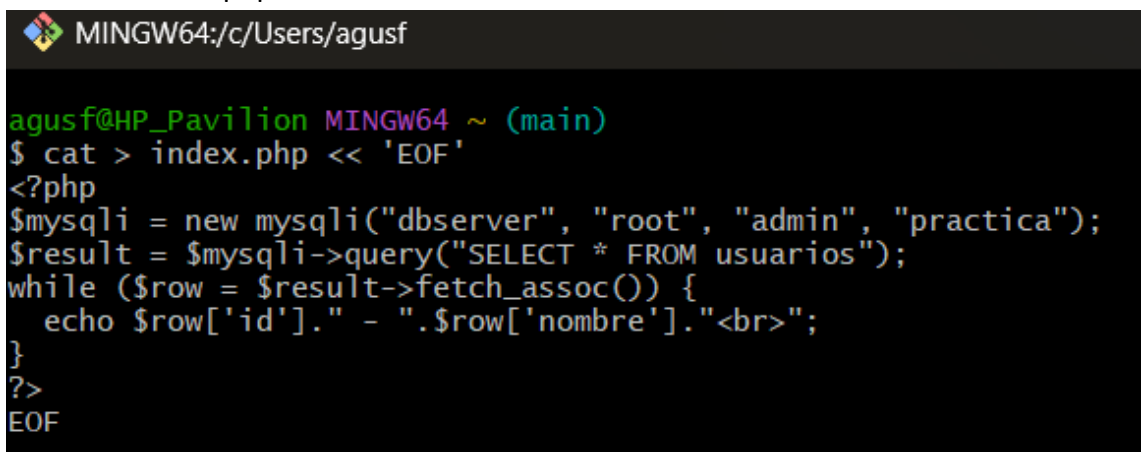
6. Luego utilizando MySql Workbench nos conectamos a dbserver y creamos una base de datos.



```
Query 1
1 • CREATE DATABASE practica;
2 • USE practica;
3 • CREATE TABLE usuarios (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100));
4 • INSERT INTO usuarios (nombre) VALUES ('Arias Diego'), ('Godoy Sergio'), ('Ferrari Agustin');
5
6
```

#	Time	Action	Message
✓ 1	10:48:17	CREATE DATABASE practica	1 row(s) affected
✓ 2	10:48:17	USE practica	0 row(s) affected
✓ 3	10:48:17	CREATE TABLE usuarios (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100))	0 row(s) affected
✓ 4	10:48:17	INSERT INTO usuarios (nombre) VALUES ('Arias Diego'), ('Godoy Sergio'), ('Ferrari Agustin')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0

7. Creamos localmente la aplicación de consulta de registros de la base de datos con el nombre index.php



```
MINGW64:/c/Users/agusf

agusf@HP_Pavilion MINGW64 ~ (main)
$ cat > index.php << 'EOF'
<?php
$mysqli = new mysqli("dbserver", "root", "admin", "practica");
$result = $mysqli->query("SELECT * FROM usuarios");
while ($row = $result->fetch_assoc()) {
    echo $row['id']." - ".$row['nombre']."<br>";
}
?>
EOF
```

8. Utilizamos un dockerfile, con el siguiente contenido para construir la imagen con la aplicación incorporada:

FROM php:8.2-apache

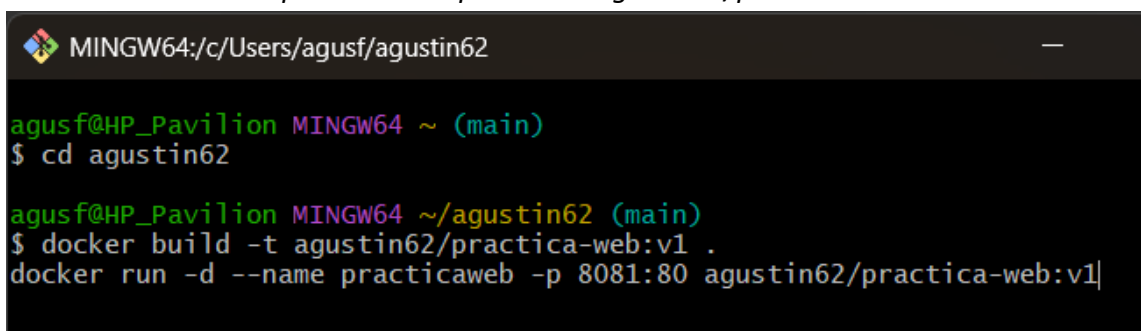
COPY index.php /var/www/html/

RUN docker-php-ext-install mysqli

Luego construimos la imagen con los siguientes comandos

docker build -t agustin62/practica-web:v1 .

docker run -d --name practicaweb -p 8081:80 agustin62/practica-web:v1



```
MINGW64:/c/Users/agusf/agustin62

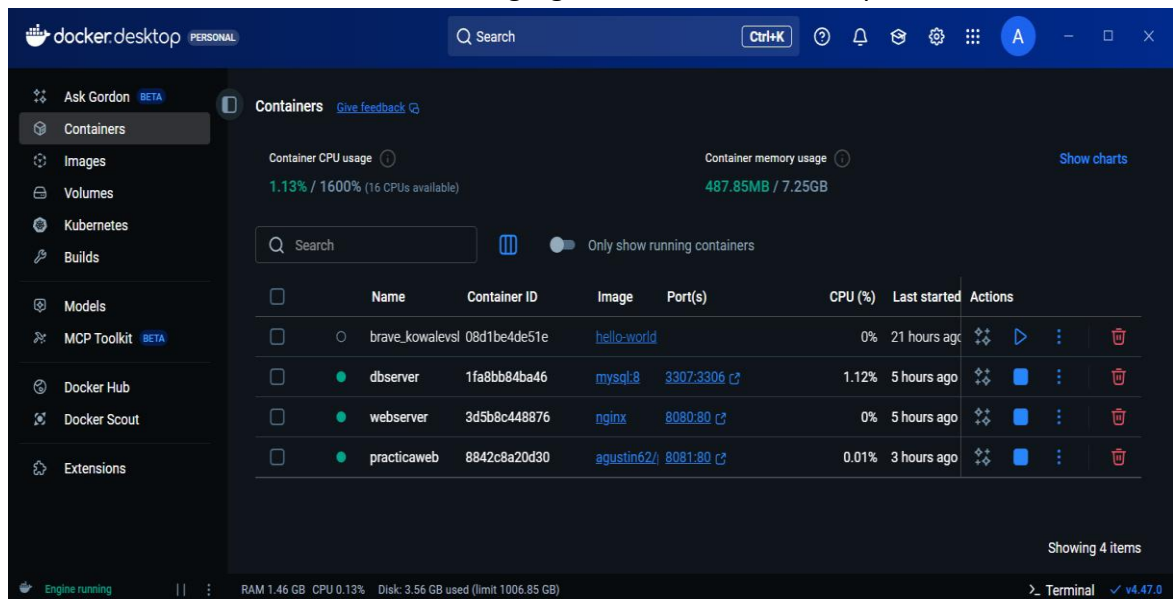
agusf@HP_Pavilion MINGW64 ~ (main)
$ cd agustin62

agusf@HP_Pavilion MINGW64 ~/agustin62 (main)
$ docker build -t agustin62/practica-web:v1 .
$ docker run -d --name practicaweb -p 8081:80 agustin62/practica-web:v1
```



```
MINGW64:/c/Users/agusf/agustin62
[+] Building 41.4s (9/9) FINISHED docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 124B 0.0s
=> [internal] load metadata for docker.io/library/php:8.2-apache 3.0s
=> [auth] library/php:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 251B 0.1s
=> [1/3] FROM docker.io/library/php:8.2-apache@sha256:b3876890595b471c1ee 22.0s
=> => resolve docker.io/library/php:8.2-apache@sha256:b3876890595b471c1eeb 0.0s
=> => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e 32B / 32B 0.4s
=> => sha256:50e2b6face7299e43069931af7f0db8c72d1edb20763ffbc2 891B / 891B 0.7s
=> => sha256:cdb5bbef0e17cc369fe94c20c6641182328878775ab6b87579 247B / 247B 0.9s
=> => sha256:b29d636b5d63db81da11d665e918098a1474c86258807e167d 252B / 252B 0.9s
=> => sha256:4ad1ce053292f2375b8f6a0517dcb2ebcb95355c64319b 2.46kB / 2.46kB 0.5s
=> => sha256:ec1283305d46a106c5964550157c480c0ea7f4d2e726 11.46MB / 11.46MB 2.1s
=> => sha256:2e39efed0f048e4687c422b747beb5b09d50fd4b1910bafa3a 487B / 487B 0.3s
=> => sha256:634ab520a54aba8c5a6d62d3ddbe679ad8ece38b85bc 12.33MB / 12.33MB 4.3s
=> => sha256:08a4a8d8574bdab5eba763e3bf4a4db8fd3cf95640dcf336dd 485B / 485B 0.8s
=> => sha256:4a03afdd881641c99da1c6e6b7ebd9f598387b9a52df2c8525 427B / 427B 0.3s
=> => sha256:042f7bbd46e8cf11ba540d911d86b5906080f716737ab7 4.22MB / 4.22MB 1.6s
=> => sha256:e9d7b3818d3e96587ed711eca7cf7b0c26da753cb25862d3aa 225B / 225B 0.5s
=> => sha256:349592d2c6d1cb2f129cf094d77965d5666bb0637 117.84MB / 117.84MB 13.8s
=> => sha256:3f814cc06e5ab25037f69a4038c0dceae6a11dfe9f8327ad4 225B / 225B 0.5s
=> => sha256:8c7716127147648c1751940b9709b6325f2256290d32 29.78MB / 29.78MB 5.7s
=> => extracting sha256:8c7716127147648c1751940b9709b6325f2256290d3201662ec 1.7s
=> => extracting sha256:3f814cc06e5ab25037f69a4038c0dceae6a11dfe9f8327ad41 0.0s
=> => extracting sha256:349592d2c6d1cb2f129cf094d77965d5666bb06373dd45d9dcd 4.3s
=> => extracting sha256:e9d7b3818d3e96587ed711eca7cf7b0c26da753cb25862d3aa0 0.0s
=> => extracting sha256:042f7bbd46e8cf11ba540d911d86b5906080f716737ab7a94b0 0.3s
=> => extracting sha256:4a03afdd881641c99da1c6e6b7ebd9f598387b9a52df2c85250 0.0s
=> => extracting sha256:08a4a8d8574bdab5eba763e3bf4a4db8fd3cf95640dcf336ddc 0.0s
=> => extracting sha256:634ab520a54aba8c5a6d62d3ddbe679ad8ece38b85bcb152f82 0.1s
=> => extracting sha256:2e39efed0f048e4687c422b747beb5b09d50fd4b1910bafa3a9 0.0s
=> => extracting sha256:ec1283305d46a106c5964550157c480c0ea7f4d2e7263cb746f 0.5s
=> => extracting sha256:4ad1ce053292f2375b8f6a0517dcb2ebcb95355c64319ba2e9a 0.0s
=> => extracting sha256:b29d636b5d63db81da11d665e918098a1474c86258807e167d2 0.0s
=> => extracting sha256:cdb5bbef0e17cc369fe94c20c6641182328878775ab6b875799 0.0s
=> => extracting sha256:50e2b6face7299e43069931af7f0db8c72d1edb20763ffbc21 0.0s
=> => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75 0.0s
=> [2/3] COPY index.php /var/www/html/ 0.8s
=> [3/3] RUN docker-php-ext-install mysqli 15.0s
=> exporting to image 0.4s
=> => exporting layers 0.2s
=> => exporting manifest sha256:85613a93efa8b7da74c3a1e7d31bc0d2788d411d9b1 0.0s
=> => exporting config sha256:ed6659213f21ce1c819becf62dbc0a355b89d5550ab69 0.0s
=> => exporting attestation manifest sha256:44266722531ca9fdccbc5e51a9b92ae 0.0s
=> => exporting manifest list sha256:402806ebebe63496d2ab4f2e5ab5f0e75e97df 0.0s
=> => naming to docker.io/agustin62/practica-web:v1 0.0s
=> => unpacking to docker.io/agustin62/practica-web:v1 0.1s
8842c8a20d30e2620c89a0d8bb391992a48b18e21dcd902ca084fff5dc39fc1f
agusf@HP_Pavilion MINGW64 ~/agustin62 (main)
```

Podemos visualizar los contenedores agregados en Docker Desktop



Luego hacemos el push a Docker Hub, utilizando los siguientes comandos:

docker login

docker push agustin62/practica-web:v1

```
MINGW64:/c/Users/agusf/agustin62
agusf@HP_Pavilion MINGW64 ~/agustin62 (main)
$ docker login
docker push agustin62/practica-web:v1
Authenticating with existing credentials... [Username: agustin62]

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

Login Succeeded
The push refers to repository [docker.io/agustin62/practica-web]
9356cf298740: Pushed
634ab520a54a: Pushed
2e39efed0f04: Pushed
4ad1ce053292: Pushed
cdb5bbef0e17: Pushed
b29d636b5d63: Pushed
3f814cc06e5a: Pushed
50e2b6face72: Pushed
df3c6b4556c5: Pushed
08a4a8d8574b: Pushed
e9d7b3818d3e: Pushed
4f4fb700ef54: Pushed
20cec82b8d54: Pushed
4a03afdd8816: Pushed
8c7716127147: Pushed
042f7bbd46e8: Pushed
349592d2c6d1: Pushed
ec1283305d46: Pushed
v1: digest: sha256:402806ebebe63496d2ab4f2e5ab5f0e75e97df4170ce92d75e5461f0ccb97dab size:
856
agusf@HP_Pavilion MINGW64 ~/agustin62 (main)
$ |
```

Conectamos el contenedor practicaweb a la red mynet, donde están conectados dbserver y webserver, mediante el comando:

docker network connect mynet practicaweb

Y luego inspeccionamos la red para confirmar la incorporación del contenedor:

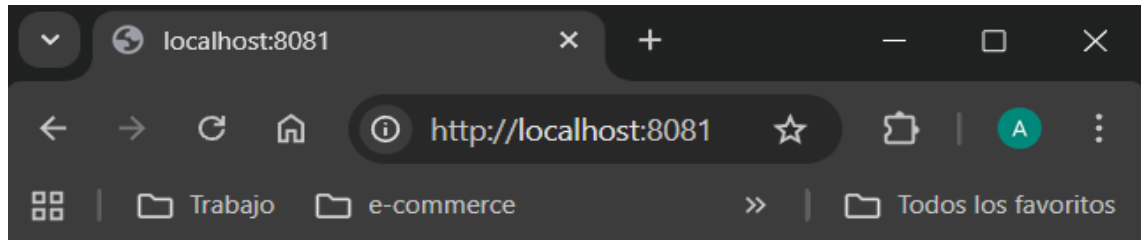
docker network inspect mynet

```
"Containers": {
  "1fa8bb84ba46430bd3cae9409af8d5efea86c186e6ce754ec2e1cc8ca31b7bdd": {
    "Name": "dbserver",
    "EndpointID": "1d883cb4f5699719d6b0621273ca7c1384cd8325b4fef3f2568e2d67ec7c65",
    "MacAddress": "ca:c4:0c:e9:07:00",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  },
  "3d5b8c4488768b4a0f53ca50a779488b36a293cd103bd124ccb640bed549393": {
    "Name": "webserver",
    "EndpointID": "ad989b7729acea2c748e8a5296ea596efc565d832db4ea6bbaaf52ac119b12",
    "MacAddress": "66:8b:de:57:29:e4",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  },
  "8842c8a20d30e2620c89a0d8bb391992a48b18e21dcd902ca084fff5dc39fc1f": {
    "Name": "practicaweb",
    "EndpointID": "b53e0b42d401868118be7b63d2d6a816f073534a1547e6c64af08e4d967161",
    "MacAddress": "e2:25:21:d9:cc:7c",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  }
}
```

Ahora, utilizando el navegador podemos ver en la dirección:

<http://localhost:8081>

Se comprueba la ejecución de la aplicación web index.php leyendo los datos de la tabla usuarios.



1 - Arias Diego
2 - Godoy Sergio
3 - Ferrari Agustin

Parte II

Se creó un repositorio en Github, y se clono posteriormente en la PC, utilizando los siguientes comandos:

git init

git remote add origin <https://github.com/AgustinCFerrari/practica-docker.git>

```
MINGW64:/c/Users/agusf/onedrive/github/practica-docker

agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ git init
Initialized empty Git repository in C:/Users/agusf/OneDrive/GitHub/practica-docker/.git/

agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ git remote add origin https://github.com/AgustinCFerrari/practica-docker.git

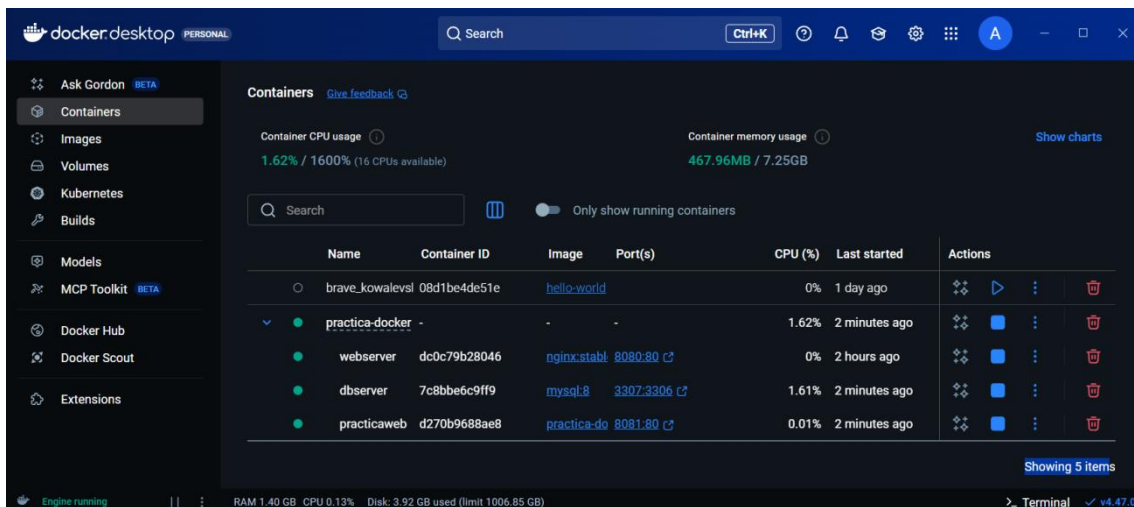
agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ |
```

Una vez hechos el Dockerfile y el docker-compose.yml , se levantan los contenedores con la siguiente instrucción:

Docker compose up -d --build


```
MINGW64/c/Users/agusf/onedrive/github/practica-docker
agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ docker compose up -d --build
[+] Building 2.7s (12/12) FINISHED
=> [internal] load local bake definitions 0.0s
=> => reading from stdin 5928 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 365B 0.0s
=> [internal] load metadata for docker.io/library/php:8.2-apache 1.6s
=> [auth] library/php:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/php:8.2-apache@sha256:b3876890595b471c1eebe0b073a81070f18100045c92761cb9 0.0s
=> => resolve docker.io/library/php:8.2-apache@sha256:b3876890595b471c1eebe0b073a81070f18100045c92761cb9 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 58B 0.0s
=> CACHED [2/4] RUN docker-php-ext-install mysqli 0.0s
=> CACHED [3/4] WORKDIR /var/www/html 0.0s
=> CACHED [4/4] COPY src/ ./ 0.0s
=> exporting to image 0.2s
=> => exporting layers 0.0s
=> => exporting manifest sha256:ce401e04c6d6e5e7a2a18e7a6560468c88a1299f3cfe7dcf6dfaa7ca4f94049 0.0s
=> => exporting config sha256:d9014e3854af7569ef1d9d4e3f69abb1db3759e5726d0b992a479f992a1d7e46 0.0s
=> => exporting attestation manifest sha256:a4b7a0402733873ff9d2ca8a108dd56d3a8590ac2b0d13f44c022823b8864 0.0s
=> => exporting manifest list sha256:d15a8782ee82737b57fbcdd87d01c35d25286483eaa603d71ddc61d1737c466f 0.0s
=> => naming to docker.io/library/practica-docker-practicaweb:latest 0.0s
=> => unpacking to docker.io/library/practica-docker-practicaweb:latest 0.0s
=> resolving provenance for metadata file 0.0s
[+] Running 4/4
✔ practica-docker-practicaweb Built 0.0s
✔ Container dbserver Started 4.1s
✔ Container practicaweb Started 2.7s
✔ Container webserver Running 0.0s
agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$
```

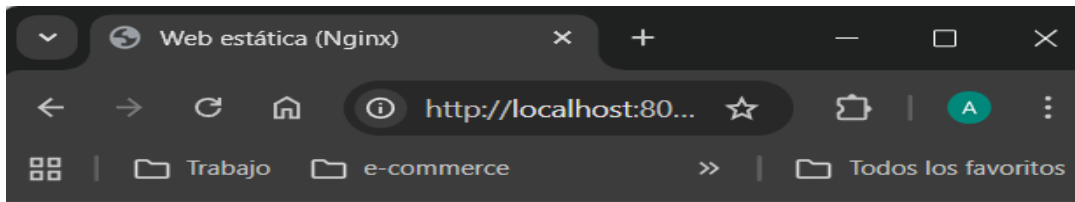
Se verifica que los contenedores están corriendo en el Docker desktop



También se verifica el acceso a los servicios a través del navegador:

<http://localhost:8080> ==> Nginx (contenedor webserver)

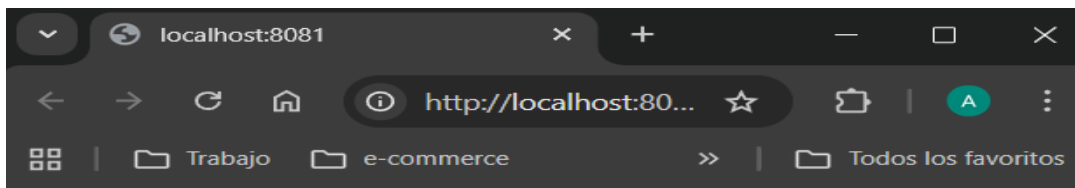
<http://localhost:8081> ==> Apache+PHP (contenedor practicaweb), la app PHP consulta la tabla Usuarios de la BD en el contenedor dbserver.



Hola desde Nginx (webserver)

Este sitio sirve archivos estáticos.

La app PHP está en <http://localhost:8081>



Usuarios

- 1 - Arias Diego
- 2 - Godoy Sergio
- 3 - Ferrari Agustin

Ahora subimos al repositorio de Github los archivos de la práctica formativa, mediante los siguientes comandos:

git add .

git commit -m "Parte II práctica formativa - dockerizado"

git push origin main

```
MINGW64:/c:/Users/agusf/onedrive/github/practica-docker

agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ git add .

agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ git commit -m "Parte II práctica formativa - dockerizado"
[main (root-commit) d2c64c9] Parte II práctica formativa - dockerizado
4 files changed, 87 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 docker-compose.yml
 create mode 100644 src/index.php
 create mode 100644 web/index.html

agusf@HP_Pavilion MINGW64 ~/onedrive/github/practica-docker (main)
$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 1.65 KiB | 338.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AgustinCFerrari/practica-docker.git
 * [new branch]      main -> main
```

Por último subimos la imagen donde se encuentra la aplicación PHP, al repositorio Docker Hub, con los siguientes comandos

Etiqueta imagen

`docker tag practicaweb:latest agustin62/practica-docker-practicaweb`

#Sube al repositorio

`docker push agustin62/practica-docker-practicaweb`

```
MINGW64:/c/Users/agusf

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
practica-docker-practicaweb  latest     d15a8782ee82  19 hours ago  707MB
practicaweb          latest     58b2cec36d1d  19 hours ago  707MB
agustin62/practica-web  v1         402806ebeb66  26 hours ago  707MB
mysql                8          5367102acfef  10 days ago   1.07GB
nginx                latest     8adbdc969e22  7 weeks ago   279MB
nginx                stable     3b4019335070  7 weeks ago   279MB
hello-world          latest     54e66cc1dd1f  8 weeks ago   20.3kB

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker tag practica-docker-practicaweb:latest agustin62/practica-docker-practicaweb

agusf@HP_Pavilion MINGW64 ~ (main)
$ docker push agustin62/practica-docker-practicaweb
Using default tag: latest
The push refers to repository [docker.io/agustin62/practica-docker-practicaweb]
3b7a47b7497e: Pushed
c5b5bbef0e17: Mounted from agustin62/practica-web
4ad1ce053292: Mounted from agustin62/practica-web
349592d2c6d1: Mounted from agustin62/practica-web
634ab520a54a: Mounted from agustin62/practica-web
b29d636b5d63: Mounted from agustin62/practica-web
4a03afdd8816: Mounted from agustin62/practica-web
08a4a8d8574b: Mounted from agustin62/practica-web
ec1283305d46: Mounted from agustin62/practica-web
042f7bbd46e8: Mounted from agustin62/practica-web
c914d24536a5: Pushed
e9d7b3818d3e: Mounted from agustin62/practica-web
3f814cc06e5a: Mounted from agustin62/practica-web
4f4fb700ef54: Mounted from agustin62/practica-web
efe5dc41c844: Pushed
2e39efed0f04: Mounted from agustin62/practica-web
8c7716127147: Mounted from agustin62/practica-web
50e2b6face72: Mounted from agustin62/practica-web
latest: digest: sha256:d15a8782ee82737b57fbc87d01c35d25286483eaa6f603d71ddc61d1737c466f size: 856
```

A partir del archivo Dockerfile.web creamos la imagen Nginx donde estará ubicado el archivo index.html, con el comando:

`docker build -t agustin62/practica-docker-webstatic -f Dockerfile.web .`

Luego subimos el contenedor al repositorio de Docker Hub, con la instrucción:

`docker push agustin62/practica-docker-webstatic`

```
PS C:\Users\agusf\Downloads\DevOps\practica-docker> docker push agustin62/practica-docker-webstatic
Using default tag: latest
The push refers to repository [docker.io/agustin62/practica-docker-webstatic]
ae7b49ada9e3: Mounted from library/nginx
8e5924dfa87c: Mounted from library/nginx
1d9a18bc0c05: Mounted from library/nginx
82eb62151b9d: Mounted from library/nginx
5c32499ab806: Mounted from library/nginx
08d4f638eff8: Mounted from library/nginx
be90cf255959: Mounted from library/nginx
08d6db973230: Pushed
375a53b5fa2e: Pushed
latest: digest: sha256:7a6d9e0cec72c3c61e77de6dfaf24ec66f0bc4ad5881ab1ebc807d404924b7ff size: 856
```

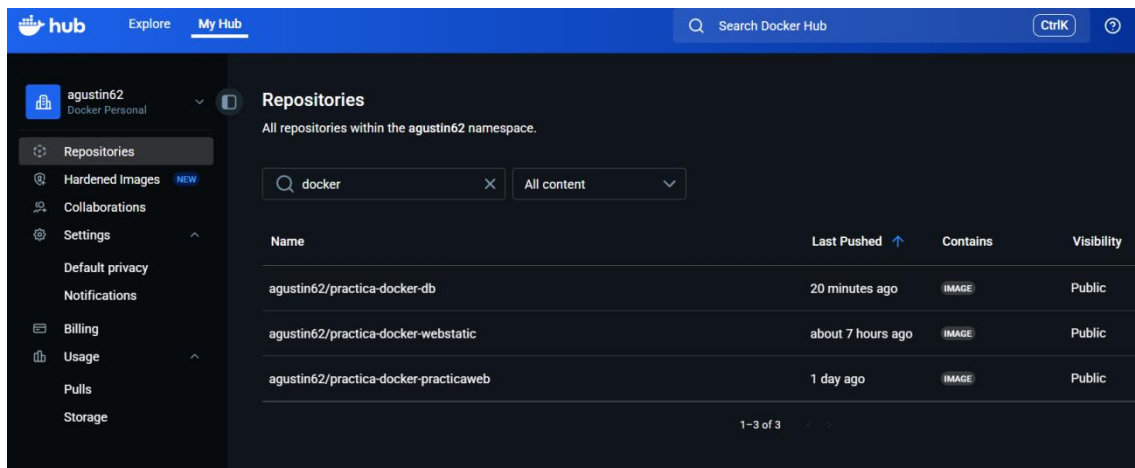
Con el archivo Dockerfile.db creamos la imagen MySQL donde estará ubicado el archivo init.sql que crea la base de datos, mediante el comando:

```
docker build -t agustin62/practica-docker-db -f Dockerfile.db .
```

Luego subimos el contenedor al repositorio de Docker Hub, con la instrucción:

```
docker push agustin62/practica-docker-db
```

Accediendo a Docker Hub por el navegador vemos ambos repositorios subidos



Con Docker Desktop previamente instalado y además con el archivo docker-compose.yml ubicado en la misma carpeta donde se ejecuta desde la consola de cualquier otra máquina:

```
Docker compose down -v # borra contenedores + volumen (datos)
```

```
Docker compose up -d
```

Se levantarán aplicaciones multi-contenedor definidas en el archivo docker-compose.yml en segundo plano.