

Ejercicio 3 - imagen con Dockerfile - Aplicación web

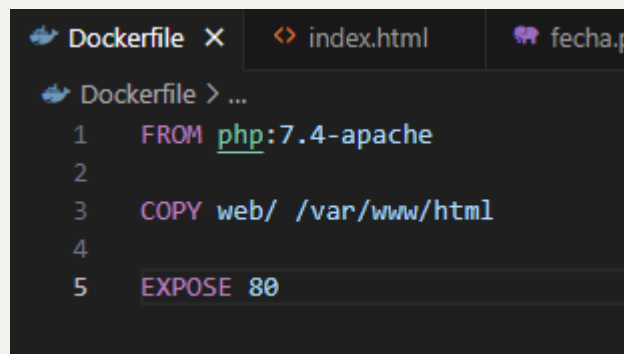
Usa un contenedor que ejecute una instancia de la imagen
Coloca en el directorio raíz del servicio web (/var/www/html) un "sitio web" donde figure tu nombre el sitio deberá tener al menos un
Coloca en ese mismo directorio raíz el siguiente script php , llámalo fecha.php
Ver la salida del script fecha.php y de la página index.html en el navegador
Una vez creada la imagen, súbela a tu cuenta de Docker Hub
Borra la imagen de tu Docker local
Ejecuta un contenedor usando esa imagen
creación de un nuevo contenedor con esa imagen y su ejecución. Cambia el puerto del contenedor, por ejemplo, - p 1234:80

Ejercicio 3 - imagen con Dockerfile - Aplicación web

Usa un contenedor que ejecute una instancia de la imagen

Coloca en el directorio raíz del servicio web (/var/www/html) un "sitio web" donde figure tu nombre el sitio deberá tener al menos un

archivo index.html sencillo y un archivo .css



```
Dockerfile X index.html fecha.p
Dockerfile > ...
1 FROM php:7.4-apache
2
3 COPY web/ /var/www/html
4
5 EXPOSE 80
```

##

```
Dockerfile  index.html X fecha.php # style.css
web > <> index.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  Haga clic para agregar un punto de interrupción
5  <title>MI sitio web</title>
6  <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  <h1>Bienvenido al sitio de Jesús Alberto López Vázquez</h1>
10 <p>Haz clic para ver la <a href="fecha.php">fecha y hora actual</a>.</p>
11 </body>
12 </html>
```

```
# style.css > h1
body {
  font-family: Arial, sans-serif;
  background-color: #f3f3f3;
  color: #333;
  text-align: center;
  margin-top: 50px;
}

h1 {
  color: #2a75bb;
}
```

Coloca en ese mismo directorio raíz el siguiente script php , llámalo fecha.php

```
fecha.php
<?php
setlocale(LC_TIME, "es_ES.UTF-8");
$mes_actual = strftime("%B");
$fecha_actual = date("d/m/Y");
$hora_actual = date("H:i:s");
echo "<h1>Información</h1>";
echo "<p>Hoy es $fecha_actual</p>";
echo "<p>El mes es: <strong>$mes_actual</strong></p>";
echo "<p>Hora: $hora_actual</p>";
```

Ver la salida del script fecha.php y de la página index.html en el navegador

Construimos la imagen con el comando

```
docker build -t ejercicio3
```

Ejecutamos el contenedor y aquí es donde voy a exponer el puerto 8000 como se indica en la tarea

```
docker run -d --name ejercicio3 -p 8000:80 ejercicio3
```

```

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/SegundoParcial/
Tarea3-JesusAlbertoLopezVazquez/ejercicio3 (ejercicio3)
$ docker build -t ejercicio3 .
[+] Building 28.6s (8/8) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 96B                                0.0s
=> [internal] load metadata for docker.io/library/php:7.4-apache  2.3s
=> [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                                  0.0s
=> => transferring context: 957B                                   0.0s
=> [1/2] FROM docker.io/library/php:7.4-apache@sha256:c9d7e608f7383267 25.4s
=> => resolve docker.io/library/php:7.4-apache@sha256:c9d7e608f73832673 0.0s
=> => sha256:05e465aaa99a358add4acecdade8f39843089069f31fea 892B / 892B 0.2s
=> => sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f766 246B / 246B 0.3s
=> => sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c466 2.46kB / 2.46kB 0.4s
=> => sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e1 10.20MB / 10.20MB 2.5s
=> => sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171 491B / 491B 0.3s
=> => sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c2 10.76MB / 10.76MB 8.9s
=> => sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a2 514B / 514B 0.2s
=> => sha256:9b233e420ac7bbca645bb82c213029762acf1742400c07 475B / 475B 0.2s
=> => sha256:25f85b498fd5bfc6cce951513219fe480850dab 19.25MB / 19.25MB 10.3s
=> => sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea421 270B / 270B 0.3s
=> => sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade3 91.63MB / 91.63MB 20.9s
=> => sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0 226B / 226B 0.5s
=> => sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a 31.41MB / 31.41MB 12.0s
=> => extracting sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73deeaab 1.3s
=> => extracting sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e 0.0s
=> => extracting sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376adf916 2.3s
=> => extracting sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea4217 0.0s
=> => extracting sha256:25f85b498fd5bfc6cce951513219fe480850daba71e6e99 0.4s
=> => extracting sha256:9b233e420ac7bbca645bb82c213029762acf1742400c076 0.0s
=> => extracting sha256:fe42347c4ecfc90333acd9cad13912387eea39d13827a25 0.0s
=> => extracting sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829 0.1s
=> => extracting sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171c 0.0s
=> => extracting sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fb 0.4s
=> => extracting sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade9 0.0s
=> => extracting sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f7662 0.0s
=> => extracting sha256:05e465aaa99a358add4acecdade8f39843089069f31fea0 0.0s
=> [2/2] COPY web/ /var/www/html                                  0.5s
=> exporting to image                                              0.2s
=> => exporting layers                                           0.1s
=> => exporting manifest sha256:456d961ff6aea26ffe54a6287b5cd7f64909542 0.0s
=> => exporting config sha256:5686c162eb162de90743b2b8b059261ee72a14cc4 0.0s
=> => exporting attestation manifest sha256:bdea23a6e65bba3b9b754844354 0.0s
=> => exporting manifest list sha256:8a05bef40889a066565e65b8c6a5af4c09 0.0s
=> => naming to docker.io/library/ejercicio3:latest              0.0s
=> => unpacking to docker.io/library/ejercicio3:latest           0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linu
x/zlyef2t7xqx8tdonxh3l2plcn

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/SegundoParcial/
Tarea3-JesusAlbertoLopezVazquez/ejercicio3 (ejercicio3)
$ docker run -d --name ejercicio3 -p 8000:80 ejercicio3
756a870a180c39ee7f22bf3a290df41a1bb7974eca0768a16ca03f0eb7158e44

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/SegundoParcial/
Tarea3-JesusAlbertoLopezVazquez/ejercicio3 (ejercicio3)
$

```

En el navegador ponemos localhost:8000 y comprobamos que funciona correctamente

Bienvenido al sitio de Jesús Alberto López Vázquez

Haz clic para ver la [fecha y hora actual](#).

Información

Hoy es 07/04/2025

El mes es: **April**

Hora: 09:39:43

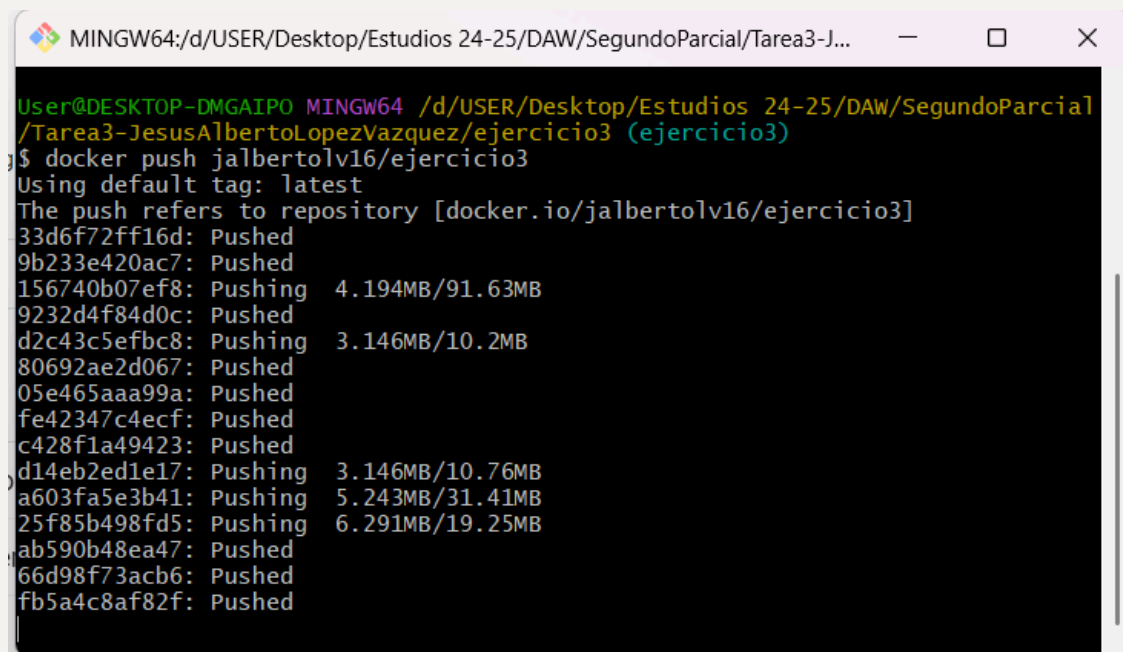
Una vez creada la imagen, súbela a tu cuenta de Docker Hub

Debemos etiquetar la imagen ejercicio3 con mi nombre de usuario de docker hub más el nombre de la imagen

```
docker tag ejercicio3 jalbertolv16/ejercicio3
```

Realizamos el push

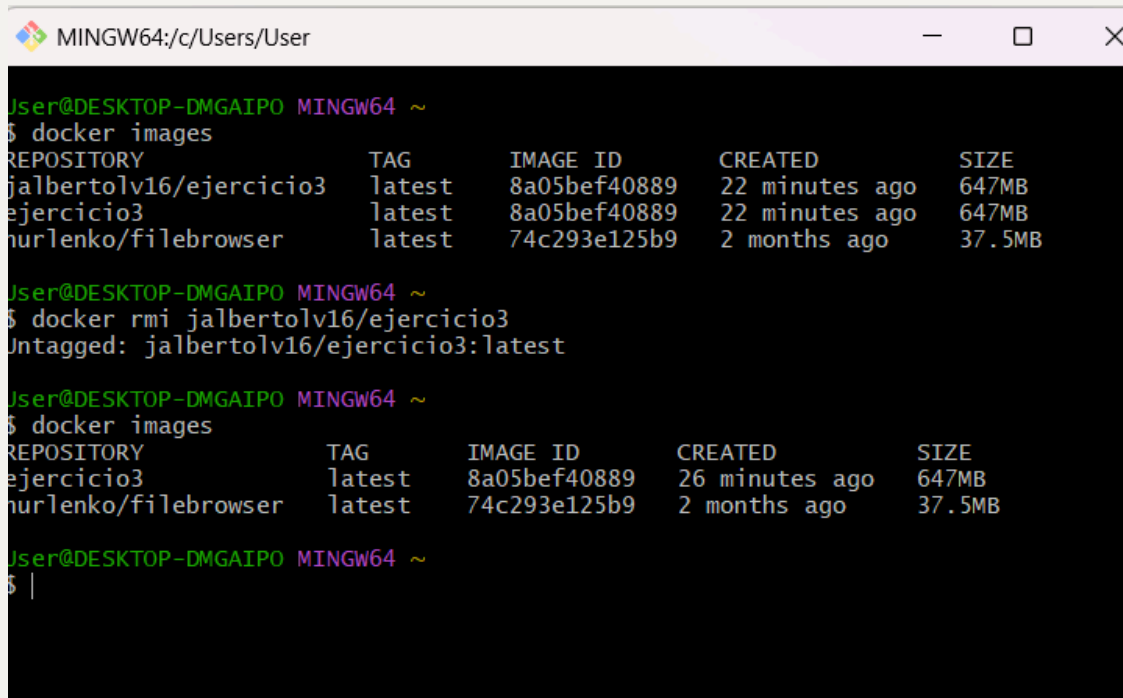
```
docker push jalbertolv16/ejercicio3
```



```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/SegundoParcial/Tarea3-J...
User@DESKTOP-DMGAIPO MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/SegundoParcial
/Tarea3-JesusAlbertoLopezVazquez/ejercicio3 (ejercicio3)
$ docker push jalbertolv16/ejercicio3
Using default tag: latest
The push refers to repository [docker.io/jalbertolv16/ejercicio3]
33d6f72ff16d: Pushed
9b233e420ac7: Pushed
156740b07ef8: Pushing 4.194MB/91.63MB
9232d4f84d0c: Pushed
d2c43c5efbc8: Pushing 3.146MB/10.2MB
80692ae2d067: Pushed
05e465aaa99a: Pushed
fe42347c4ecf: Pushed
c428f1a49423: Pushed
d14eb2ed1e17: Pushing 3.146MB/10.76MB
a603fa5e3b41: Pushing 5.243MB/31.41MB
25f85b498fd5: Pushing 6.291MB/19.25MB
ab590b48ea47: Pushed
66d98f73acb6: Pushed
fb5a4c8af82f: Pushed
```

Borra la imagen de tu Docker local

```
docker rmi jalbertolv16/ejercicio3
```



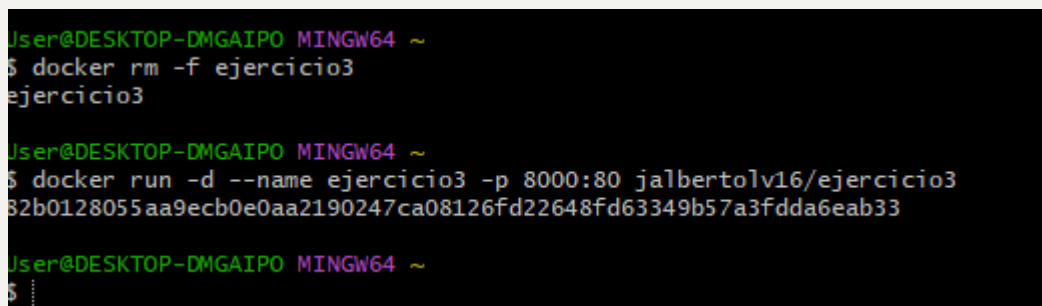
The screenshot shows a Windows terminal window titled 'MINGW64:/c/Users/User'. The user runs the command 'docker images' and receives a table of local images. Then, they run 'docker rmi jalbertolv16/ejercicio3' and see the output 'Untagged: jalbertolv16/ejercicio3:latest'. Finally, they run 'docker images' again to see the updated list of images.

```
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ docker images  
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE  
jalbertolv16/ejercicio3 latest      8a05bef40889  22 minutes ago 647MB  
ejercicio3          latest      8a05bef40889  22 minutes ago 647MB  
hurlenko/filebrowser latest      74c293e125b9  2 months ago  37.5MB  
  
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ docker rmi jalbertolv16/ejercicio3  
Untagged: jalbertolv16/ejercicio3:latest  
  
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ docker images  
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE  
ejercicio3          latest      8a05bef40889  26 minutes ago 647MB  
hurlenko/filebrowser latest      74c293e125b9  2 months ago  37.5MB  
  
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ |
```

Ejecuta un contenedor usando esa imagen

```
docker run -d --name ejercicio3 -p 8000:80 jalbertolv16/ejercicio3
```

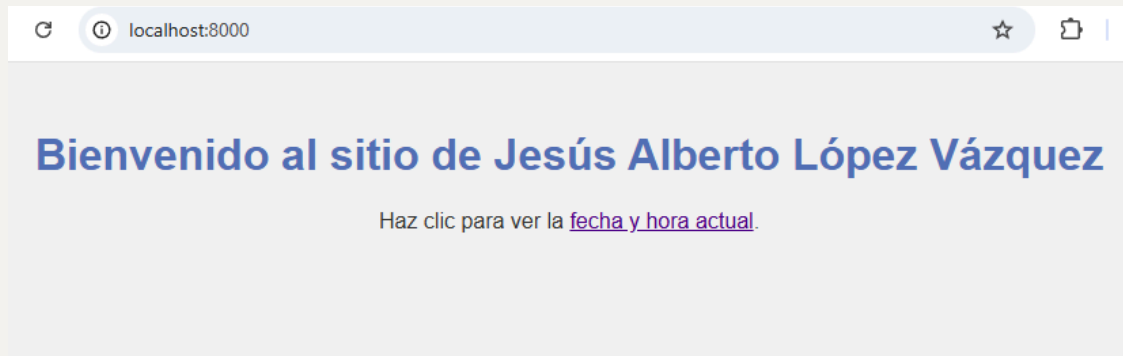
para ello antes he parado el contenedor ejercicio3, lo he borrado también y por último he lanzado el contenedor de nuevo usando esa imagen con ese comando.



The screenshot shows a Windows terminal window titled 'MINGW64:/c/Users/User'. The user runs 'docker rm -f ejercicio3' to remove the container. Then, they run 'docker run -d --name ejercicio3 -p 8000:80 jalbertolv16/ejercicio3' to create and start a new container. The output shows the container ID '82b0128055aa9ecb0e0aa2190247ca08126fd22648fd63349b57a3fdda6eab33'.

```
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ docker rm -f ejercicio3  
ejercicio3  
  
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ docker run -d --name ejercicio3 -p 8000:80 jalbertolv16/ejercicio3  
82b0128055aa9ecb0e0aa2190247ca08126fd22648fd63349b57a3fdda6eab33  
  
User@DESKTOP-DMGAIP0 MINGW64 ~  
$ .....
```

Y comprobamos que efectivamente vuelve a funcionar



Información

Hoy es 07/04/2025

El mes es: April

Hora: 10:15:19

creación de un nuevo contenedor con esa imagen y su ejecución. Cambia el puerto del contenedor, por ejemplo, - p 1234:80

```
docker stop ejercicio3
docker rm ejercicio3
```

Lanzar el contenedor

```
docker run -d --name ejercicio3 -p 1234:80 jalbertolv16/ejercicio3
```

```
User@DESKTOP-DMGAIP0 MINGW64 ~
$ docker stop ejercicio3
ejercicio3

User@DESKTOP-DMGAIP0 MINGW64 ~
$ docker rm ejercicio3
ejercicio3

User@DESKTOP-DMGAIP0 MINGW64 ~
$ docker run -d --name ejercicio3 -p 1234:80 jalbertolv16/ejercicio3
2df9b056d195b2ec2c153315dda48dfef0e5fe9cf837fe6ebd905d6fd4717aa5

User@DESKTOP-DMGAIP0 MINGW64 ~
$ |
```

