

## Práctica 10. Docker

Lee detenidamente cada uno de los puntos antes de realizar las tareas solicitadas.  
Revisa los recursos incluidos.

Al adjuntar ficheros, introducelos en carpetas con el nombre de la agrupación a la que corresponde. Revisa el apartado “A entregar” al final del doc.

### Parte 1: Docker (Dockerfile)

Actividades 1-5 se realizan en conjunto: Crear imagen docker para web y arrancar.

1. Crea un directorio **src/** en tu equipo y dentro coloca un fichero **index.php** de prueba con el siguiente código (y muestra capturas):  
<?php phpinfo(); ?>

```
PS C:\Users\USUARIO\src> mkdir src
```

```
Directorio: C:\Users\USUARIO

Mode                LastWriteTime         Length
----                -
d-----          10/02/2025         15:40
```

```
PS C:\Users\USUARIO> cd src
PS C:\Users\USUARIO\src>
```

```
PS C:\Users\USUARIO> echo "<?php phpinfo();
?>" > src/index.php
```

```
PS C:\Users\USUARIO\src> Get-Content index.p
hp
<?php phpinfo(); ?>
```

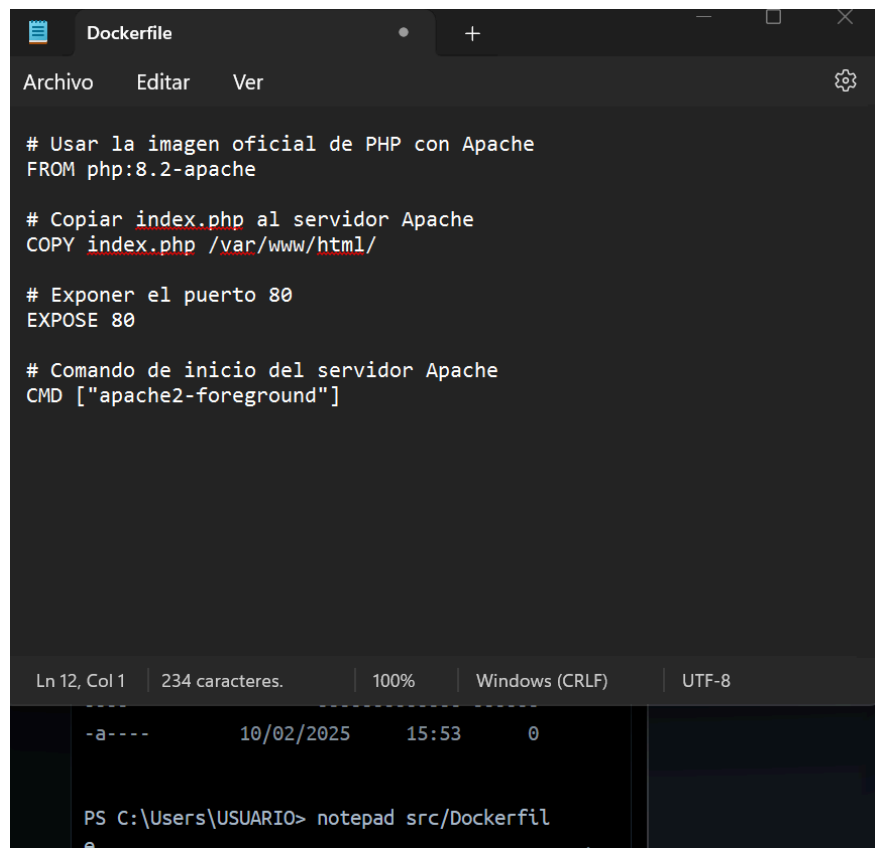
2. Prepara un contenedor con **Apache y PHP** y el programa .php anterior usando **Dockerfile**. Crea el dockerfile en /src (adjunta o copia el contenido de este fichero).

```
PS C:\Users\USUARIO> New-Item src/Dockerfile -ItemType File
```

```
Directorio: C:\Users\USUARIO\src
```

Mode	LastWriteTime	Length
----	-----	-----
-a----	10/02/2025 15:53	0

- Emplea una imagen oficial de PHP con Apache (ej. php:8.2-apache)
- Copia index.php en el contenedor, guarda en /var/www/html
- Expón el puerto 80
- Recuerda finalizar con un comando de inicio



```
Archivo  Editar  Ver  [Settings Icon]

# Usar la imagen oficial de PHP con Apache
FROM php:8.2-apache

# Copiar index.php al servidor Apache
COPY index.php /var/www/html/

# Exponer el puerto 80
EXPOSE 80

# Comando de inicio del servidor Apache
CMD ["apache2-foreground"]

Ln 12, Col 1 | 234 caracteres. | 100% | Windows (CRLF) | UTF-8

-a---- 10/02/2025 15:53 0

PS C:\Users\USUARIO> notepad src/Dockerfile
```

3. Construye la **imagen** de la aplicación (capturas del docker build y su resultado)

```
PS C:\Users\USUARIO> cd src
```

```
PS C:\Users\USUARIO\src> docker build -t m
i-app-web:1.1 .
[+] Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/0) docker:desktop-li[+]
Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/1) docker:desktop-li[+] Buil
ding 0.2s (1/2) docker:desktop-li[+] Building 0.4s (1/2) docker:desktop-li[+] Building
0.5s (1/2) docker:desktop-li[+] Building 0.7s (1/2) docker:desktop-li[+] Building 0.8
s (1/2) docker:desktop-li[+] Building 1.0s (1/2) docker:desktop-li[+] Building 1.1s (1
/2) docker:desktop-li[+] Building 1.3s (1/2) docker:desktop-li[+] Building 1.4s (1/2)
docker:desktop-li[+] Building 1.6s (1/2) docker:desktop-li[+] Building 1.7s (1/2) doc
ker:desktop-li[+] Building 1.9s (1/2) docker:desktop-li[+] Building 2.0s (1/2) docker:
desktop-li[+] Building 2.2s (1/2) docker:desktop-li[+] Building 2.3s (1/2) docker:desk
```

```
(7/8) docker:desktop-li[+] Building 4.0s (7/8) docker:desktop-li[+] Building 4.2s (8/8
) FINISHED docker:desktop-linux[2/2] COPY index.php /va 0.0s
=> [internal] load build definitio 0.1s
=> => transferring dockerfile 282B 0.0s
=> [internal] load metadata for do 3.6s
=> [auth] library/php:pull token f 0.0s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 2B 0.1s
=> [internal] load build context 0.0s
=> => transferring context: 30B 0.0s
=> [1/2] FROM docker.io/library/ph 0.1s
```

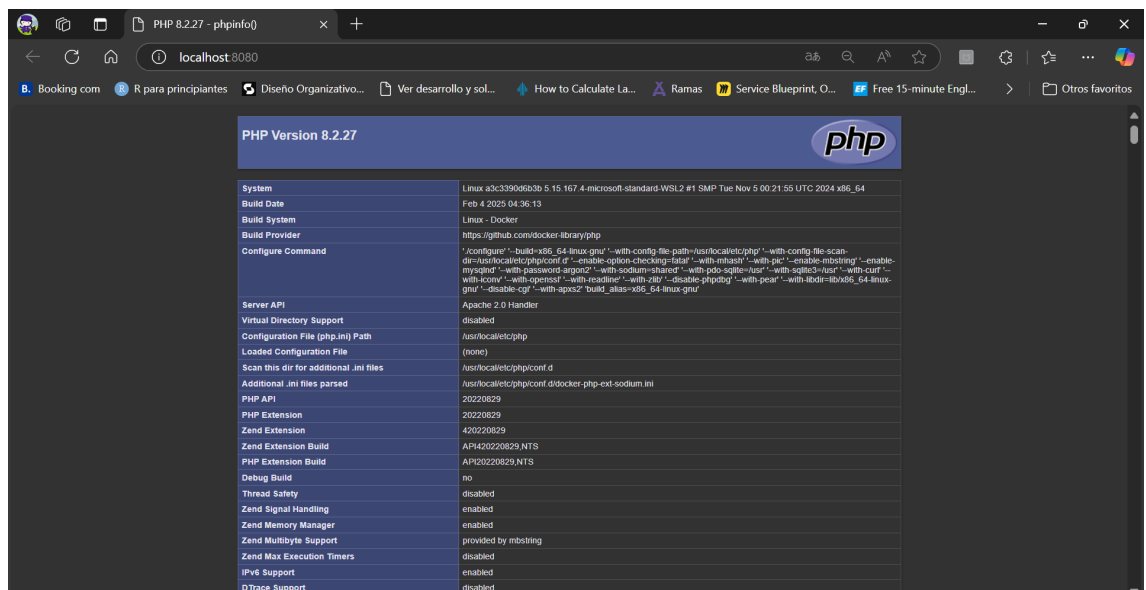
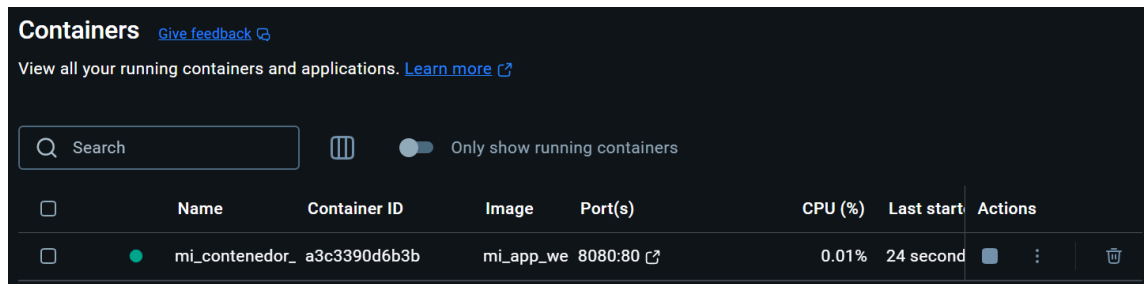
```
=> [1/2] FROM docker.io/library/ph 0.1s
=> => resolve docker.io/library/ph 0.1s
=> CACHED [2/2] COPY index.php /va 0.0s
=> exporting to image 0.2s
=> => exporting layers 0.0s
=> => exporting manifest sha256:c1 0.0s
=> => exporting config sha256:c769 0.0s
=> => exporting attestation manife 0.0s
=> => exporting manifest list sha2 0.0s
=> => naming to docker.io/library/ 0.0s
=> => unpacking to docker.io/libra 0.0s
```

```
PS C:\Users\USUARIO\src> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
src-web	latest	89abf67b797c	4 hours ago	709MB
mi_app_web	1.1	fc46f2305214	30 hours ago	709MB

4. Ejecuta el **contenedor** (captura del run). Establece vínculos de puertos y un nombre específico. Abre <http://localhost:8080> y verifica que se muestra la información de PHP (captura).

```
PS C:\Users\USUARIO\src> docker run -d -p 8080:80 --name mi_contenedor_web mi_app_web:1.
1
a3c3390d6b3b6a81a12a7bca31d87bc90178c43d81df1414d5fcd124ab0de37f
PS C:\Users\USUARIO\src> 
```



5. Vía comandos, muestra los contenedores que tengas en marcha y detén el contenedor creado (capturas de las instrucciones).

```
PS C:\Users\USUARIO\src> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3c3390d6b3b   mi_app_web:1.1 "docker-php-entrypoi..." About a minute ago Up About a minute   0.0.0.0:8080->80/tcp   mi_contenedor_web
```

```
PS C:\Users\USUARIO\src> docker stop mi_contenedor_web
mi_contenedor_web
PS C:\Users\USUARIO\src> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
PS C:\Users\USUARIO\src>
```

## Parte 2: Docker-compose (YAML)

Actividades 6-8 se realizan en conjunto: Añadir MySQL.

6. Ahora vamos a agregar **MySQL**, pero este se ejecutará desde **otro contenedor** diferente al creado previamente. Para que puedan trabajar en conjunto, crea un archivo **docker-compose.yml**. Establece las *variables de entorno* de MySQL para definir una contraseña al usuario root (a tu elección,

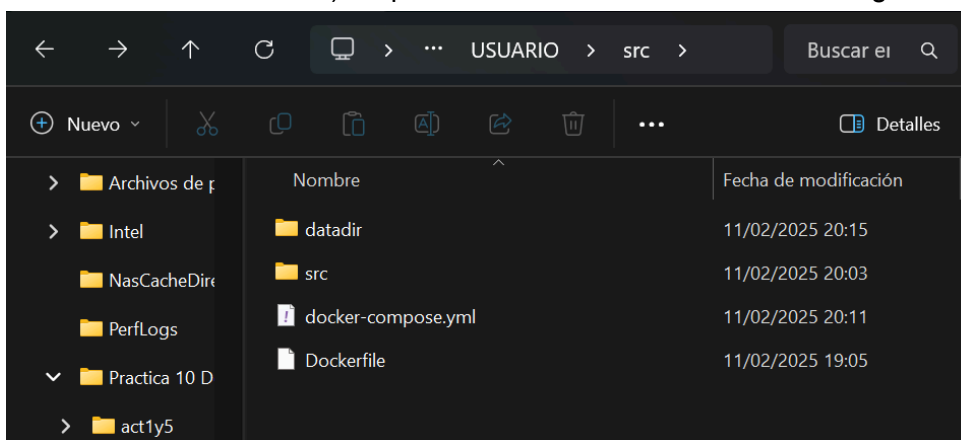
pero toma nota), crear una base de datos con tu nombre y `_db` al final (formato `[nombre_db]`) y un usuario con tu nombre (contraseña también a tu elección).




```

1  services:
2    web:
3      build: .
4      ports:
5        - "8080:80"
6      volumes:
7        - ./src:/var/www/html
8      depends_on:
9        - db
10
11   db:
12     image: mysql:8.0
13     restart: always
14     environment:
15       MYSQL_ROOT_PASSWORD: hidemi1
16       MYSQL_DATABASE: hidemi_db
17       MYSQL_USER: hidemi
18       MYSQL_PASSWORD: hidemi1
19     ports:
20       - "3306:3306"
21     volumes:
22       - /c/Users/USUARIO/src/datadir:/var/lib/mysql
23

```

- Continuando con el punto anterior, asegúrate de definir un **volumen** para poder tener persistencia a nivel local con el contenedor de MySQL. Para esto, dentro de `src/`, en local, crea una **subcarpeta llamada "datadir"**, a vincular con la carpeta correspondiente del contenedor (ver documentación – en recomendaciones). Explica cómo has introducido esta configuración.



Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
src	-	-	-	0.7%	3 minutes	  

Terminal





PS C:\Users\USUARIO\src> docker-compose up --build  
[+] Running 0/1  
- Service web Bui... 0.2s  
[+] Running 0/10s (0/0) docker:desktop-li[+] Building 0.0s (0/0) docker:desktop-li[+] Building 0.0s (0/1) docker:desktop-li[+] Building 0.2s (1/2) docker:desktop-li[+] Building 0.3s (1/2) docker:desktop-li[+] Building 0.5s (1/2) docker:desktop-li[+] Building 0.6s (1/2) docker:desktop-li[+] Building 0.8s (1/2) docker:desktop-li[+] Building 0.9s (1/2) docker:desktop-li[+] Building 1.1s (1/2) docker:desktop-li[+] Building 1.2s (1/2) docker:desktop-li[+] Building 1.4s (1/2) docker:desktop-li[+] Building 1.5s (1/2) docker:desktop-li[+] Building 1.7s (1/2) docker:desktop-li[+] Building 1.8s (1/2) docker:desktop-li[+] Building 2.0s (1/2) docker:desktop-li[+] Building 2.1s (1/2) docker:desktop-li[+] Building 2.3s (1/2) docker:desktop-li[+] Building 2.4s (1/2) docker:desktop-li[+] Building 2.6s (1/2) docker:desktop-li[+] Building 2.7s (1/2) docker:desktop-

Windows PowerShell

```
PS C:\Users\USUARIO\src> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
f3f2807d63cc	src-web	"docker-php-entrypoi..."	4 minutes ago	Up 11 seconds	0.0.0.0:8080->80/tcp
ca502e849b13	mysql:8.0	"docker-entrypoint.s..."	4 minutes ago	Up 11 seconds	0.0.0.0:3306->3306/tcp, 33060/tcp
src-web-1					
src-db-1					

Este equipo > Windows-SSD (C:) > Usuarios > USUARIO > src > datadir >

Ordenar Ver ...

Nombre	Fecha de modificación	Tipo	Tamaño
#innodb_redo	11/02/2025 20:19	Carpeta de archivos	
#innodb_temp	11/02/2025 20:19	Carpeta de archivos	
hidemi_db	11/02/2025 20:15	Carpeta de archivos	
mysql	11/02/2025 20:15	Carpeta de archivos	
performance_schema	11/02/2025 20:15	Carpeta de archivos	
sys	11/02/2025 20:15	Carpeta de archivos	
#ib_16384_0.dblwr	11/02/2025 20:19	Archivo DBLWR	192 KB
#ib_16384_1.dblwr	11/02/2025 20:14	Archivo DBLWR	8.384 KB
auto.cnf	11/02/2025 20:14	Archivo CNF	1 KB
binlog.000001	11/02/2025 20:15	Archivo 000001	2.905 KB
binlog.000002	11/02/2025 20:19	Archivo 000002	1 KB
binlog.000003	11/02/2025 20:19	Archivo 000003	1 KB
binlog.index	11/02/2025 20:19	Archivo INDEX	1 KB
ca.pem	11/02/2025 20:15	Archivo PEM	2 KB
ca-key.pem	11/02/2025 20:15	Archivo PEM	2 KB
client-cert.pem	11/02/2025 20:15	Archivo PEM	2 KB
client-key.pem	11/02/2025 20:15	Archivo PEM	2 KB

**/c/Users/USUARIO/src/datadir:** Especifica el directorio en la máquina local donde quiero que se almacenen los datos. Es para la persistencia, por si alguna vez quiero eliminar y volver a crear el contenedor, no se perderán los datos, ya que están guardados en esta carpeta datadir.

**/var/lib/mysql:** Es el directorio dentro del contenedor donde MySQL guarda sus datos, lo que garantiza que los datos sean persistentes incluso si el contenedor se elimina.

8. Crea un nuevo fichero de PHP llamado **"db\_test.php"** con el código:

```
<?php
$conn = new mysqli("db", "user", "password", "[nombre_db]");
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
echo "Conexión establecida con MySQL!";
?>
```

Cambia la información de "db" (nombre del servicio), "user", "password", "[nombre\_db]" por los datos definidos en los puntos anteriores para establecer la conexión con MySQL.



Edita **dockerfile** y genera nuevamente la **imagen** de tu aplicación, para incluir db\_test.php.

```
Dockerfile
Archivo  Editar  Ver  ⚙️

FROM php:8.2-apache

# Copiar los archivos PHP dentro del contenedor
COPY ./src /var/www/html/

# Dar permisos necesarios a la carpeta web
RUN chown -R www-data:www-data /var/www/html

# Instalar extensiones de PHP necesarias para MySQL
RUN docker-php-ext-install mysqli pdo pdo_mysql

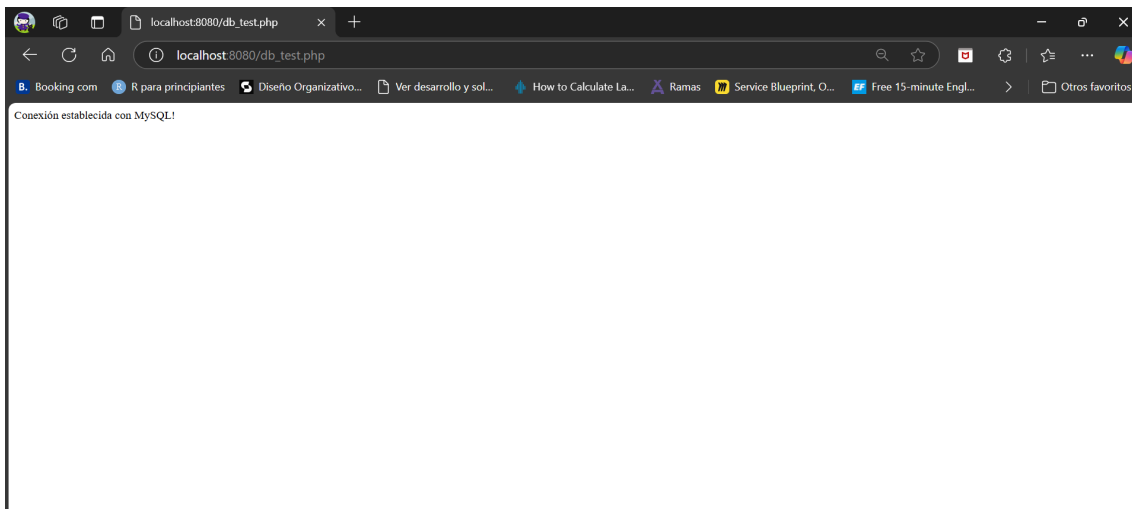
# Exponer el puerto 80 para acceder a la aplicación
EXPOSE 80
```

```
PS C:\Users\USUARIO\src> docker-compose up --build
[+] Running 0/0
 - Service web Build... 0.1s
[+] Running 0/10s (0/0) docker:desktop-linu[+] Running 0/10s (0/0) docker:desktop-linu[+] Running 0/10s (0/0)
[+] Running 0/10s (0/1) docker:desktop-linu[+] Building 0.2s (1/2) docker:desktop-linu[+] Building 0.4s (1/2)
[+] Running 0.4s (1/2) docker:desktop-linu[+] Building 0.5s (1/2) docker:desktop-linu[+] Building 0.7s (1/2)
[+] Running 0.7s (1/2) docker:desktop-linu[+] Building 0.8s (1/2) docker:desktop-linu[+] Building 1.0s (1/2)
[+] Running 1.0s (1/2) docker:desktop-linu[+] Building 1.1s (1/2) docker:desktop-linu[+] Building 1.3s (1/2)
[+] Running 1.3s (1/2) docker:desktop-linu[+] Building 1.4s (1/2) docker:desktop-linu[+] Building 1.6s (1/2)
[+] Running 1.6s (1/2) docker:desktop-linu[+] Building 1.7s (1/2) docker:desktop-linu[+] Building 1.9s (1/2)
[+] Running 1.9s (1/2) docker:desktop-linu[+] Building 2.0s (1/2) docker:desktop-linu[+] Building 2.2s (1/2)
[+] Running 2.2s (1/2) docker:desktop-linu[+] Building 2.3s (1/2) docker:desktop-linu[+] Building 2.5s (1/2)
[+] Running 2.5s (1/2) docker:desktop-linu[+] Building 2.6s (1/2) docker:desktop-linu[+] Building 2.8s (1/2)
[+] Running 2.8s (1/2) docker:desktop-linu[+] Building 2.9s (1/2) docker:desktop-linu[+] Building 3.1s (1/2)
[+] Running 3.1s (1/2) docker:desktop-linu[+] Building 3.2s (1/2) docker:desktop-linu[+] Building 3.3s (1/2)
```

```
PS C:\Users\USUARIO\src> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
eeba087bae35   src-web    "docker-php-entrypoint..." 2 minutes ago Up 1 second   0.0.0.0:8080->80/tcp
src-web-1
ca502e849b13   mysql:8.0 "docker-entrypoint.s..." 19 minutes ago Up 2 seconds  0.0.0.0:3306->3306/tcp, 33060/tcp
src-db-1
```

**Arranca los contenedores** y verifica que [http://localhost:8080/db\\_test.php](http://localhost:8080/db_test.php) funciona (captura del comando de *docker-compose up*, captura de navegador, y captura de *logs del contenedor de MySQL* donde se vea la conexión realizada desde el navegador).





```
PS C:\Users\USUARIO\src> docker logs src-db-1
2025-02-11 19:40:07+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.41-1.el9 started.
2025-02-11 19:40:07+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-02-11 19:40:07+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.41-1.el9 started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
2025-02-11T19:40:08.192803Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and
will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2025-02-11T19:40:08.200200Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.41) starting as proc
ess 1
2025-02-11T19:40:08.205431Z 0 [Warning] [MY-010159] [Server] Setting lower_case_table_names=2 because file sy
stem for /var/lib/mysql/ is case insensitive
2025-02-11T19:40:08.233235Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2025-02-11T19:40:09.446422Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
```

```
2025-02-11T19:40:10.382028Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2025-02-11T19:40:10.382581Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Enc
rypted connections are now supported for this channel.
2025-02-11T19:40:10.403507Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location
'/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2025-02-11T19:40:10.534913Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::
' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2025-02-11T19:40:10.535361Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version:
'8.0.41' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
2025-02-11T19:40:57.513245Z 0 [System] [MY-013172] [Server] Received SHUTDOWN from user <via user signal>. Sh
utting down mysqld (Version: 8.0.41).
2025-02-11T19:40:58.550141Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.4
1) MySQL Community Server - GPL.
```

```
2025-02-11 19:41:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.41-1.el9 started.
2025-02-11 19:41:03+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-02-11 19:41:03+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.41-1.el9 started.
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
2025-02-11T19:41:04.299405Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and
will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2025-02-11T19:41:04.303748Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.41) starting as proc
ess 1
2025-02-11T19:41:04.308855Z 0 [Warning] [MY-010159] [Server] Setting lower_case_table_names=2 because file sy
stem for /var/lib/mysql/ is case insensitive
stem for /var/lib/mysql/ is case insensitive
[Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all O
S users. Consider choosing a different directory.
```

```
2025-02-11T19:41:04.308855Z 0 [Warning] [MY-010159] [Server] Setting lower_case_table_names=2 because file sy
stem for /var/lib/mysql/ is case insensitive
stem for /var/lib/mysql/ is case insensitive
[Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the [Server] Insecure configura
tion for --pid-file: Location '/var/run/mysqld' in th[Server] Insecure configuration for --pid-file: Location
[Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all O
S users. Consider choosing a different directory.
2025-02-11T19:41:06.132597Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::
' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2025-02-11T19:41:06.132723Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version:
'8.0.41' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
```

### Parte 3: Datos persistentes y despliegue de App

Actividades 9-10 se realizan en conjunto: Añadir PHPMyAdmin y desplegar app web.

9. **Detén** los contenedores con *docker-compose down* (muestra captura). Edita *docker-compose.yml* para incluir **phpMyAdmin**. Configura las variables de entorno con la contraseña de root que declaraste anteriormente e indicando el nombre del servicio/contenedor de MySQL (ver documentación). Vincula los puertos 8081:80.

Una vez configurado, haz *docker-compose up* y accede a PHPMyAdmin en <http://localhost:8081> con usuario root y verifica que aparece la base de datos con tu nombre (captura de PHPMyAdmin).

```
PS C:\Users\USUARIO\src> docker-compose down
[+] Running 3/3
✓ Container src-web-1 Removed 0.1s
✓ Container src-db-1 Removed 0.1s
✓ Network src_default Removed 0.2s
```

```
PS C:\Users\USUARIO\src> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED
STATUS        PORTS    NAMES
↓
```

```

🐳 docker-compose.yml X
C: > Users > USUARIO > src > 🐳 docker-compose.yml
1  services:
2    web:
3      build: .
4      ports:
5        - "8080:80"
6      volumes:
7        - ./src:/var/www/html
8      depends_on:
9        - db
10
11   db:
12     image: mysql:8.0
13     restart: always
14     environment:
15       MYSQL_ROOT_PASSWORD: hidemi1
16       MYSQL_DATABASE: hidemi_db
17       MYSQL_USER: hidemi
18       MYSQL_PASSWORD: hidemi1
19     ports:
20       - "3306:3306"
21     volumes:
22       - ./src/datadir:/var/lib/mysql
23
24   phpmyadmin:
25     image: phpmyadmin/phpmyadmin
26     restart: always
27     ports:
28       - "8081:80"
29     environment:
30       PMA_HOST: db
31       MYSQL_ROOT_PASSWORD: hidemi1
32     depends_on:
33       - db
34

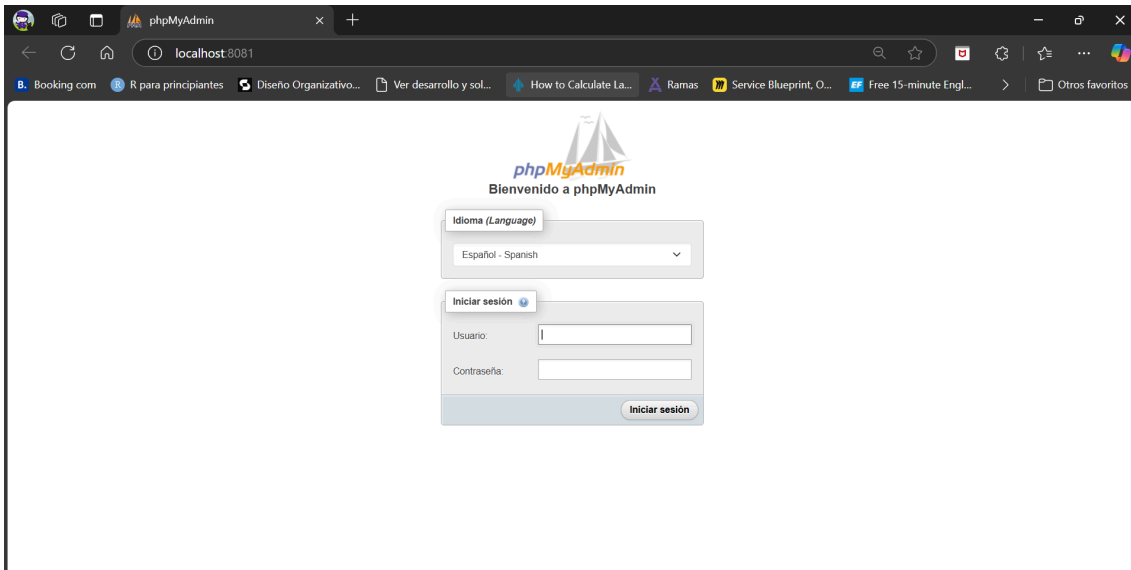
```

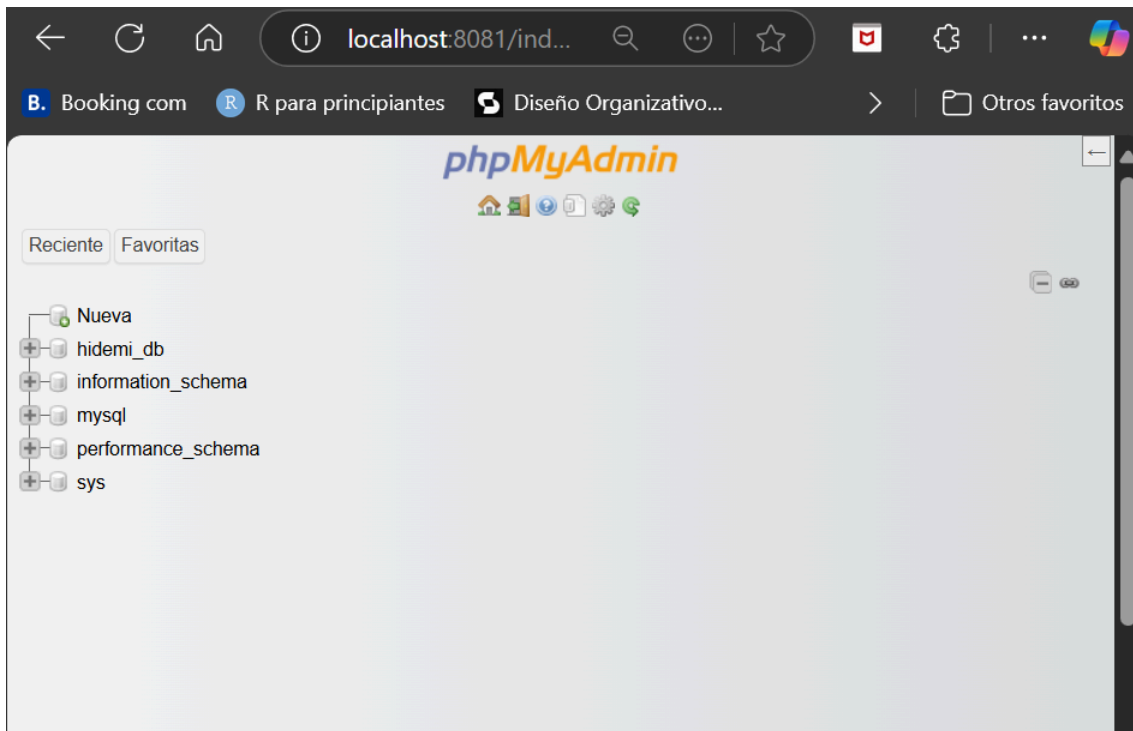
**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/> >	src	-	-	-	0.99%	23 second	<input type="checkbox"/> ⋮ 🗑







10. Detén los contenedores nuevamente. **Descarga el fichero index.php adjunto** al enunciado. Se debe desplegar con docker como una aplicación web:

```
PS C:\Users\USUARIO\src> docker-compose down
[+] Running 0/0
- Container src-phpmyadmin-1 Stopping0.1
[+] Running 0/2
- Container src-phpmyadmin-1 Stopping0.2
[+] Running 0/2
- Container src-phpmyadmin-1 Stopping0.3
[+] Running 0/2
- Container src-phpmyadmin-1 Stopping0.4
[+] Running 0/2
- Container src-phpmyadmin-1 Stopping0.5
[+] Running 0/2
```

- a. Edita el inicio del código PHP según los datos que has definido previamente, concretamente:
- // Configuración de la conexión a la base de datos**
- ```
$host = "localhost:8080"; // A DEFINIR POR EL DESARROLLADOR
$user = "nombre"; // A DEFINIR POR EL DESARROLLADOR
$pass = "password"; // A DEFINIR POR EL DESARROLLADOR
$db = "nombre_db"; // A DEFINIR POR EL DESARROLLADOR
```
- (Adjunta el fichero index.php modificado)

```
index.php X
C: > Users > USUARIO > src > src > index.php
1  <?php
2  /*
3  Requiere crear la tabla en base de datos:
4  CREATE TABLE mensajes (
5      id INT AUTO_INCREMENT PRIMARY KEY,
6      usuario VARCHAR(50) NOT NULL,
7      mensaje TEXT NOT NULL,
8      fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
9  );
10
11  */
12
13  session_start();
14
15  // Configuración de la conexión a la base de datos
16  $host = "db"; // A DEFINIR POR EL DESARROLLADOR
17  $user = "hidemi"; // A DEFINIR POR EL DESARROLLADOR
18  $pass = "hidemi1"; // A DEFINIR POR EL DESARROLLADOR
19  $db = "hidemi_db"; // A DEFINIR POR EL DESARROLLADOR
20
21  // Conexión a MySQL
22  $conn = new mysqli($host, $user, $pass, $db);
23
```

- b. Vuelve a editar el dockerfile para incluir este fichero index.php en la copia (añádalo a la carpeta src) y genera nuevamente la imagen (muestra los cambios y la creación de la imagen).

```
Dockerfile
+
Archivo Editar Ver
FROM php:8.2-apache

# Habilitar el módulo mysqli de PHP
RUN docker-php-ext-install mysqli

# Copiar el archivo index.php al contenedor
COPY ./src/index.php /var/www/html/index.php

# Exponer el puerto 80
EXPOSE 80

# Iniciar Apache
CMD ["apache2-foreground"]
```

```
PS C:\Users\USUARIO\src> docker-compose build
[+] Building 0/0
  - Service web  Bui... 0.1s
[+] Building 0/1s (0/0) docker:desktop-li
[+] Building 0.0s (0/0) docker:desktop-li
[+] Building 0/1s (0/1) docker:desktop-li
[+] Building 0.2s (1/2) docker:desktop-li
[+] Building 0.4s (1/2) docker:desktop-li
[+] Building 0.5s (1/2) docker:desktop-li
[+] Building 0.8s (2/3) docker:desktop-li
linux Building 0.7s (2/3) docker:desktop-li
=> [web internal] load build defin 0.0s
```

```
PS C:\Users\USUARIO\src> docker images
```

| REPOSITORY | TAG    | IMAGE ID     | CREATED       | SIZE  |
|------------|--------|--------------|---------------|-------|
| src-web    | latest | 7cefca72ae23 | 2 minutes ago | 709MB |

- Edita docker-compose.yml si es necesario (indica si has tenido que cambiar algo y muestra los cambios).
- Arranca los contenedores empleando el fichero YAML (captura de la instrucción up y de los contenedores en funcionamiento).

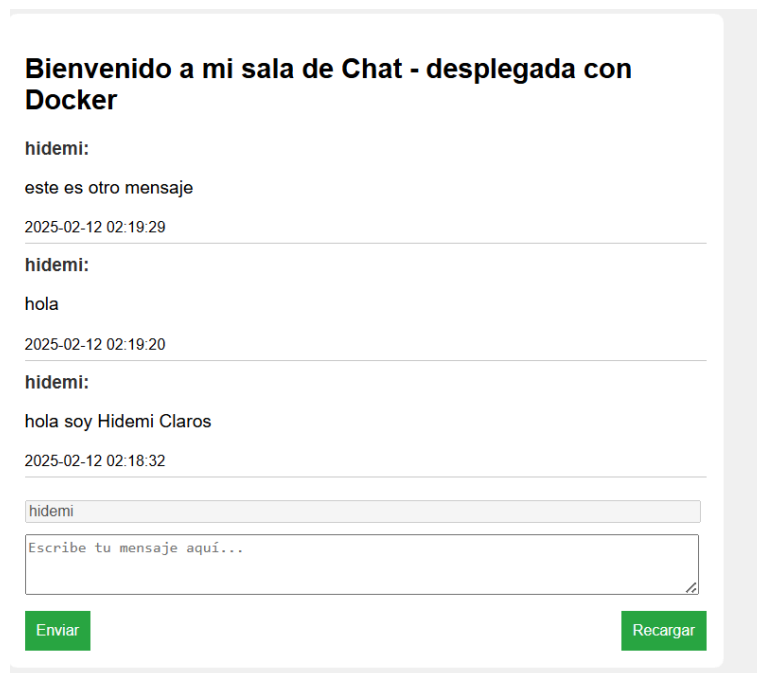
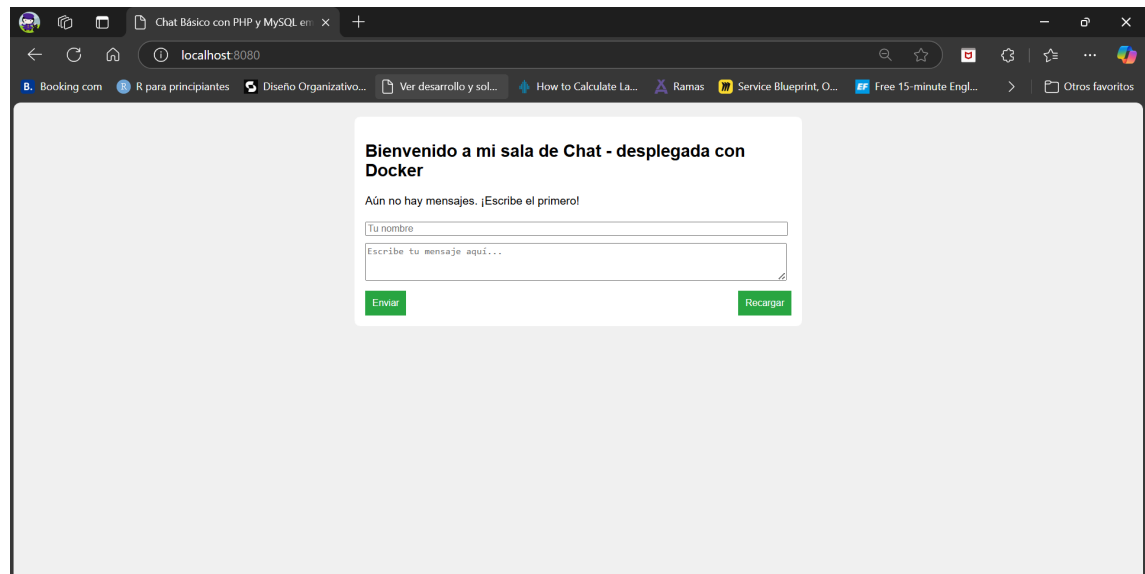
```
PS C:\Users\USUARIO\src> docker-compose up
[+] Running 2/2
  ✓ Network src_default Created0.1s
  - Container src-db-1 Creating 0.1s
  ✓ Container src-db-1 Created0.2s
[+] Running 3/4
  ✓ Network src_default Created0.1s
  ✓ Container src-db-1 Created0.2s
[+] Running 4/4
  ✓ Network src_default Created0.1s
```

```
PS C:\Users\USUARIO\src> docker ps
```

| CONTAINER ID | IMAGE                             | COMMAND                  | CREATED            | STATUS |
|--------------|-----------------------------------|--------------------------|--------------------|--------|
| 12a369da7b95 | src-web                           | "docker-php-entrypoi..." | About a minute ago | Up     |
| 9 seconds    | 0.0.0.0:8080->80/tcp              | src-web-1                |                    |        |
| 6f4ceacdc08f | phpmyadmin/phpmyadmin             | "/docker-entrypoint..."  | About a minute ago | Up     |
| 9 seconds    | 0.0.0.0:8081->80/tcp              | src-phpmyadmin-1         |                    |        |
| 2682d4fea868 | mysql:8.0                         | "docker-entrypoint.s..." | About a minute ago | Up     |
| 9 seconds    | 0.0.0.0:3306->3306/tcp, 33060/tcp | src-db-1                 |                    |        |

- Comprueba si puedes acceder a index.php vía navegador y trabajar con la aplicación (captura del navegador). Introduce mensajes nuevos.



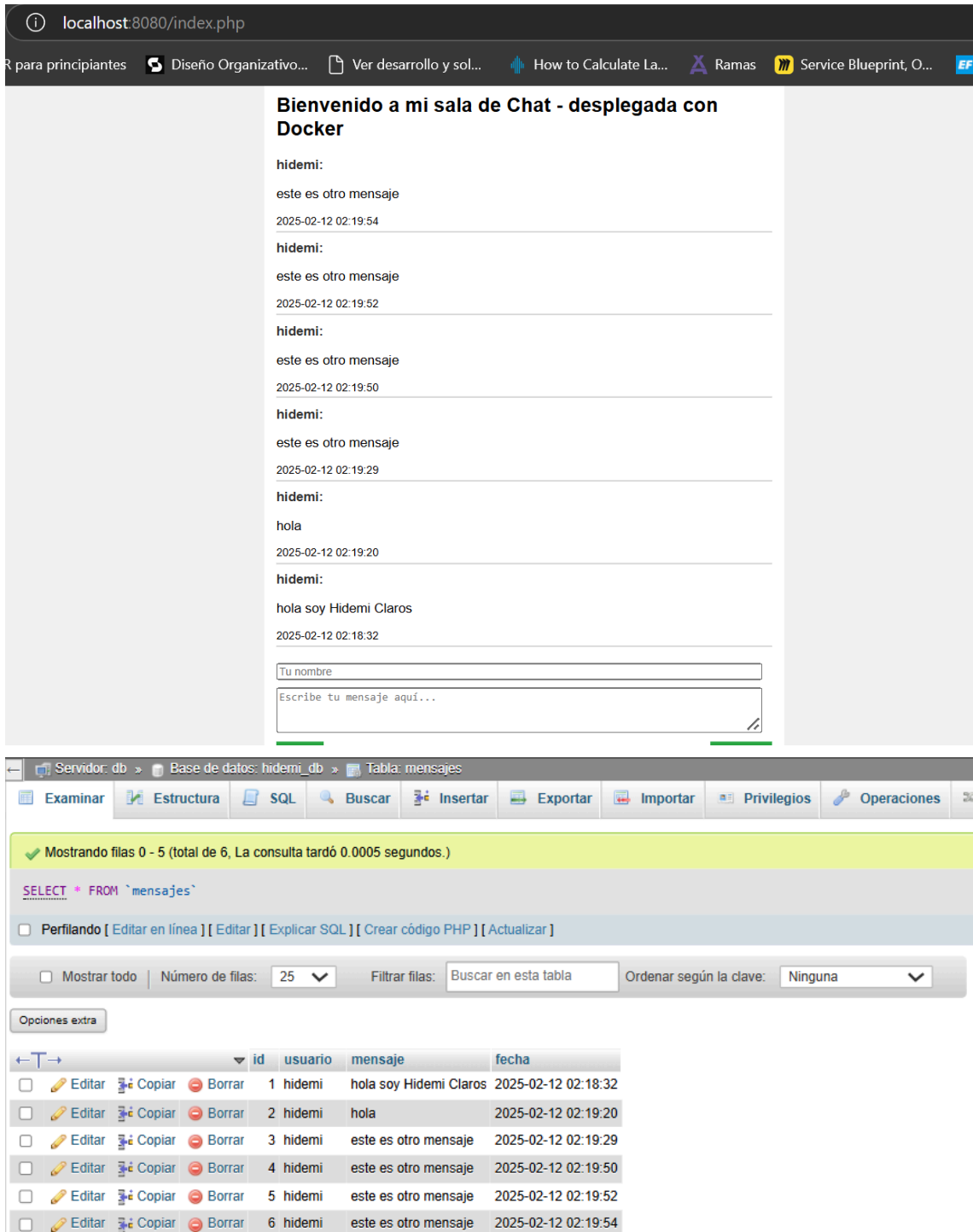


- f. Haz docker-compose down y luego otro up. Comprueba si los mensajes creados previamente siguen existiendo en la aplicación desplegada.

```
PS C:\Users\USUARIO\src> docker-compose do
wn
[+] Running 0/0
  - Container src-phpmyadmin-1  Stopping0.1
[+] Running 0/2
  - Container src-phpmyadmin-1  Stopping0.2
[+] Running 0/2
  - Container src-phpmyadmin-1  Stopping0.3
[+] Running 0/2
  - Container src-phpmyadmin-1  Stopping0.4
[+] Running 0/2
```

```
PS C:\Users\USUARIO\src> docker-compose up
[
[+] Running 2/2
  ✓ Network src_default          Created0.0s
  - Container src-db-1          Creating  0.1s
  ✓ Container src-db-1          Created0.1s
[+] Running 3/4
  ✓ Network src_default          Created0.0s

  ✓ Container src-db-1          Created0.1s
[+] Running 4/4
  ✓ Network src_default          Created0.0s
```



**Bienvenido a mi sala de Chat - desplegada con Docker**

hidemi:  
este es otro mensaje  
2025-02-12 02:19:54

hidemi:  
este es otro mensaje  
2025-02-12 02:19:52

hidemi:  
este es otro mensaje  
2025-02-12 02:19:50

hidemi:  
este es otro mensaje  
2025-02-12 02:19:29

hidemi:  
hola  
2025-02-12 02:19:20

hidemi:  
hola soy Hidemi Claros  
2025-02-12 02:18:32

Tu nombre  
Escribe tu mensaje aquí...

Servidor: db > Base de datos: hidemi\_db > Tabla: mensajes

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones

✓ Mostrando filas 0 - 5 (total de 6, La consulta tardó 0.0005 segundos.)

SELECT \* FROM `mensajes`

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

|                          | id | usuario | mensaje                | fecha               |
|--------------------------|----|---------|------------------------|---------------------|
| <input type="checkbox"/> | 1  | hidemi  | hola soy Hidemi Claros | 2025-02-12 02:18:32 |
| <input type="checkbox"/> | 2  | hidemi  | hola                   | 2025-02-12 02:19:20 |
| <input type="checkbox"/> | 3  | hidemi  | este es otro mensaje   | 2025-02-12 02:19:29 |
| <input type="checkbox"/> | 4  | hidemi  | este es otro mensaje   | 2025-02-12 02:19:50 |
| <input type="checkbox"/> | 5  | hidemi  | este es otro mensaje   | 2025-02-12 02:19:52 |
| <input type="checkbox"/> | 6  | hidemi  | este es otro mensaje   | 2025-02-12 02:19:54 |

El resultado de esta práctica es una pila LAMP montada con Docker.

### A entregar:

- Carpeta Actividad 1-5:
  - src/index.php (solo con phpinfo() )
  - src/dockerfile

- imagen creada (busca en /var/lib/docker)
- Carpeta Actividad 6-8:
  - src/dockerfile
  - src/db\_test.php
  - src/docker-compose.yml (con apache, php, mysql y la app web)
  - imagen creada (busca en /var/lib/docker)
  - Contenido de src/datadir
- Carpeta Actividad 9-10:
  - src/dockerfile
  - src/index.php (adjunto modificado)
  - src/docker-compose.yml (con apache, php, mysql, PHPMyAdmin y la app web)
  - imagen creada (busca en /var/lib/docker)
  - Contenido de src/datadir
- Documento con las capturas y las explicaciones solicitadas, en orden y claramente enumeradas.

#### Recursos:

- <https://docs.docker.com/>
- [https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql)
- [https://hub.docker.com/\\_/phpmyadmin](https://hub.docker.com/_/phpmyadmin)

#### Recordatorio:

- Docker image = aplicación y sus recurso preparados para desplegar
- Docker container = imagen desplegada en entorno virtualizado
- Dockerfile = Para poder construir una imagen
- Docker-compose.yml = para poder arrancar multiples contenedores en conjunto (en una misma red)

#### Condiciones de entrega:

- La práctica se **debe** entregar de forma **individual**, cada uno debe presentar sus propias respuestas. Sin embargo, se puede trabajar en equipo.
- Se debe entregar un documento de texto (.pdf, .docx, .odt, etc.) con los ejercicios correctamente ordenados, identificados y **numerados**.
- En cada página del documento debe aparecer el nombre completo del alumno.
- La nota comprenderá un valor numérico entre 0 y 10.
- **La fecha límite de entrega es la indicada en Google Classroom.**
- **Se podrá entregar hasta 72 horas más tarde de la fecha límite pero con una penalización sobre su puntuación (no será posible aspirar al 10).**