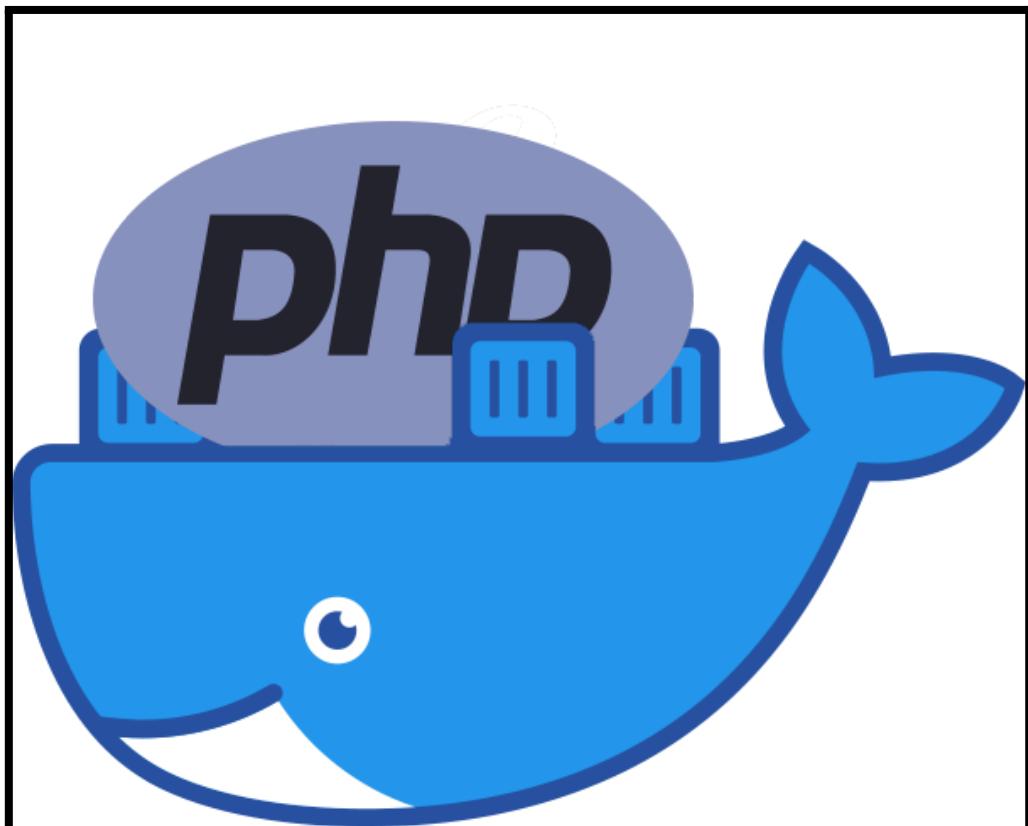


Despliegue de una aplicación PHP

Con mariadb



<i>Nombre y Apellido</i>	Laura San Román
<i>Curso</i>	2º - DAW
<i>Fecha</i>	29 / 01 / 2026
<i>Asignatura</i>	Despliegue de aplicaciones web

SERVIDOR WEB VIRTUALIZADO

La estructura es:

Carpeta llamada servidor-daw

- site1-html
 - DockerFile
 - public-html
 - index.html
- site2-php
 - DockerFile
 - public-html
 - insert.php
 - delete.php
- site3-db

Sabiendo la estructura que tiene, pasamos a la configuración de las imágenes y contenedores.

CONFIGURACIÓN DEL SITE 1 - HTML

Paso 1. Entrar a la cmd -> cd C:\Users\dawmi\servidor-daw\site1-html

Paso 2. Crear un .txt llamado DockerFile donde se guardará la siguiente información

```
-> FROM httpd:2.4  
COPY ./public-html/ /usr/local/apache2/htdocs/
```

Y se guardará como *todos los archivos para que no tenga extensión ya que sino, no funciona.

Paso 3. Una vez hecho eso, se construye -> docker build -t site1-html .

Paso 4. Ejecutar el contenedor con el puerto 7001:80

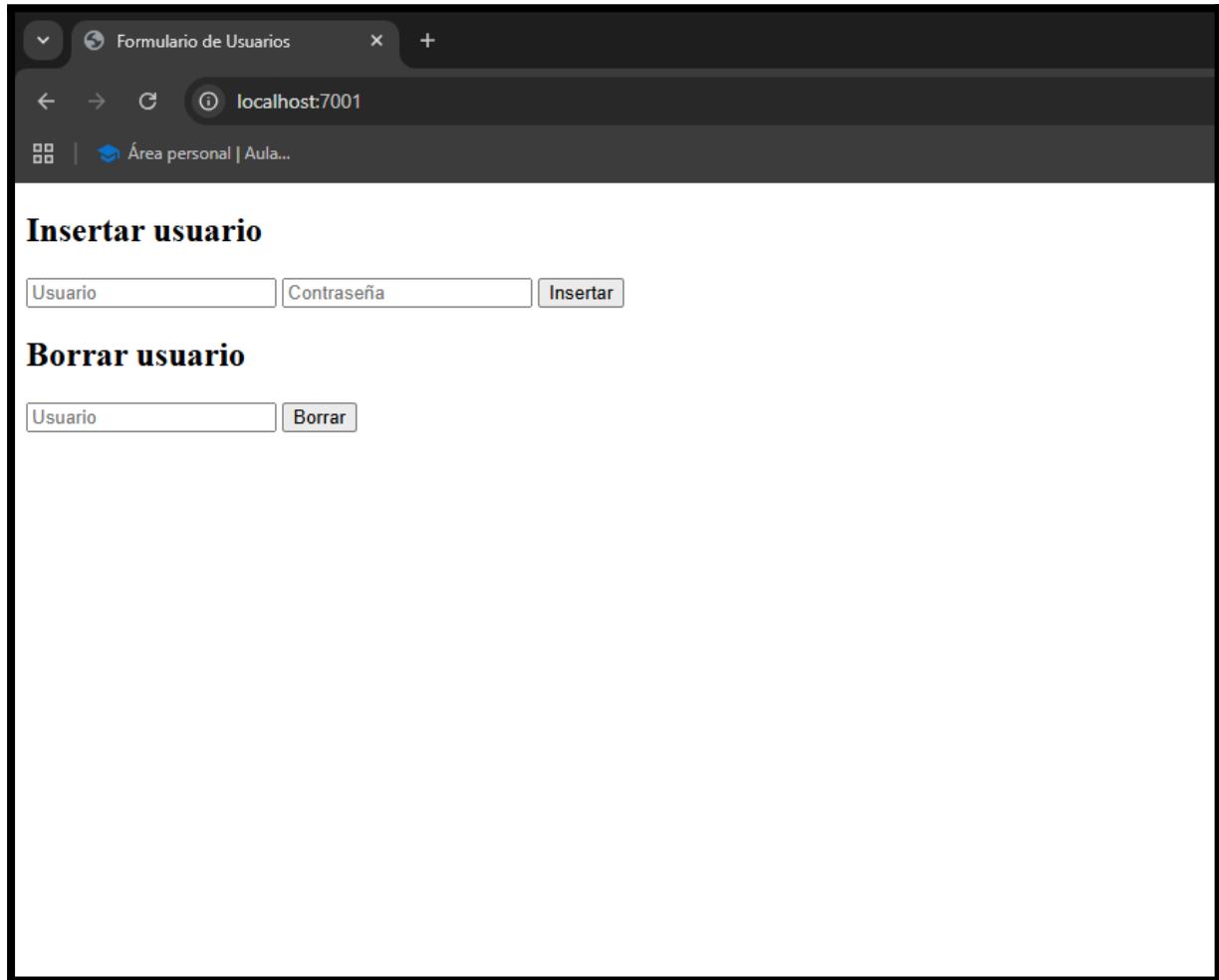
```
-> docker run -dit --name site1-html -p 7001:80 site1-html
```

Paso 5. Para comprobar que va, ir al navegador (mejor FireFox), y poner

```
-> http://localhost:7001
```

Y tendrá que verse el formulario puesto en nuestro index.html.

Tal que así:



index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Formulario de Usuarios</title>
7  </head>
8  <body>
9      <h2>Insertar usuario</h2>
10     <form action="http://localhost:7002/insert.php" method="POST">
11         <input type="text" name="username" placeholder="Usuario" required autocomplete="off">
12         <input type="password" name="password" placeholder="Contraseña" required autocomplete="off">
13         <button type="submit">Insertar</button>
14     </form>
15
16     <h2>Borrar usuario</h2>
17     <form action="http://localhost:7002/delete.php" method="POST">
18         <input type="text" name="username" placeholder="Usuario" required autocomplete="off">
19         <button type="submit">Borrar</button>
20     </form>
21 </body>
22 </html>
```

En el action de cada formulario, se pone el localhost que corresponda con su php.

CONFIGURACIÓN DEL SITE 2 - PHP

Paso 1. Entrar a la cmd -> C:\Users\dawmi\servidor-daw\site2-php

Paso 2. Crear un .txt llamado DockerFile donde se guardará la siguiente información

```
-> FROM php:8.2-apache  
RUN docker-php-ext-install mysqli  
COPY ./public-html/ /var/www/html/
```

Y se guardará como *todos los archivos para que no tenga extensión ya que sino, no funciona.

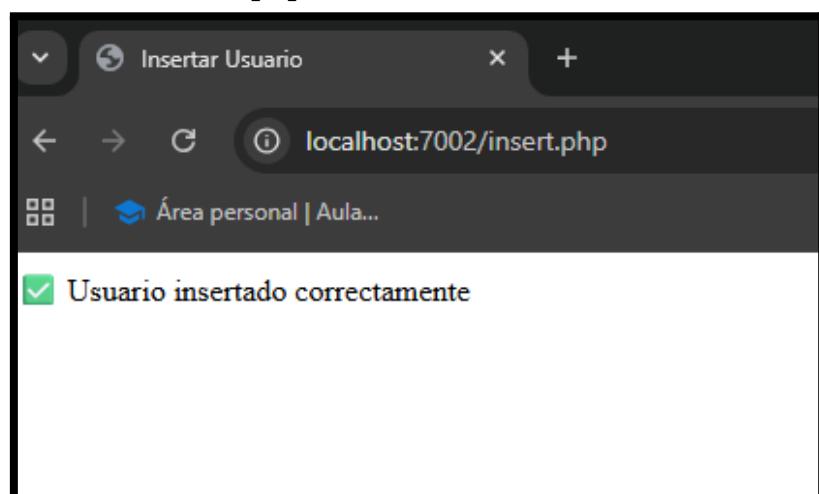
Paso 3. Una vez hecho eso, se construye -> docker build -t site2-php .

Paso 4. Ejecutar el contenedor con el puerto 7002:80

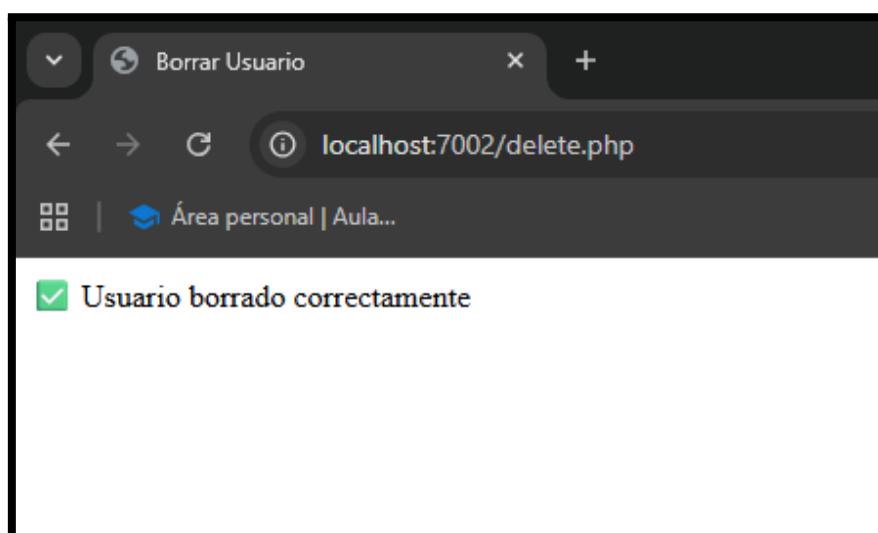
```
-> docker run -dit --name site2-php -p 7002:80 site2-php
```

Paso 5. Sabremos que funciona cuando desde nuestro formulario creado, lo llenemos y si le damos a insertar te llevará a la siguiente página:

-> <http://localhost:7002/insert.php>



Y si, le damos a eliminar un usuario, te llevará a esta:



-> <http://localhost:7002/delete.php>

insert.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Insertar Usuario</title>
7  </head>
8  <body>
9      <?php
10         $host = "host.docker.internal";
11         $puerto = 3307;
12         $usuario = "usuario";
13         $password = "password";
14         $base_datos = "users";
15
16         // Conexión
17         $conn = new mysqli($host, $usuario, $password, $base_datos, $puerto);
18         if ($conn->connect_error) { die("Error de conexión: " . $conn->connect_error); }
19
20         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
21             $username = mysqli_real_escape_string($conn, $_POST['username']);
22             $pass = mysqli_real_escape_string($conn, $_POST['password']);
23
24             $sql = "INSERT INTO users (username, password) VALUES ('$username', '$pass')";
25             if ($conn->query($sql) === TRUE) {
26                 echo "✓ Usuario insertado correctamente";
27             } else {
28                 echo "✗ Error al insertar: " . $conn->error;
29             }
30         }
31         $conn->close();
32     ?>
33  </body>
34  </html>
```

delete.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Borrar Usuario</title>
7  </head>
8  <body>
9      <?php
10         $host = "host.docker.internal";
11         $puerto = 3307;
12         $usuario = "usuario";
13         $password = "password";
14         $base_datos = "users";
15
16         // Conexión
17         $conn = new mysqli($host, $usuario, $password, $base_datos, $puerto);
18         if ($conn->connect_error) { die("Error de conexión: " . $conn->connect_error); }
19
20         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
21             $username = mysqli_real_escape_string($conn, $_POST['username']);
22
23             $sql = "DELETE FROM users WHERE username='$username'";
24             if ($conn->query($sql) === TRUE) {
25                 echo "✓ Usuario borrado correctamente";
26             } else {
27                 echo "✗ Error al borrar: " . $conn->error;
28             }
29         }
30         $conn->close();
31     ?>
32  </body>
33  </html>
```

CONFIGURACIÓN DEL SITE 3 - MARIADB

Paso 1. Entrar a la cmd -> cd C:\Users\dawmi\servidor-daw\site3-db

Paso 2. Ejecutar el contenedor de MariaDB

```
-> docker run -dit --name mariadb-daw -p 3307:3306 -e  
MARIADB_ROOT_PASSWORD=root -e MARIADB_DATABASE=users -e  
MARIADB_USER=usuario -e MARIADB_PASSWORD=password mariadb:latest
```

Paso 3. Crear la base de datos si no está creada

```
-> docker exec -i mariadb-daw mariadb -u root -proot -e "USE users; CREATE TABLE  
IF NOT EXISTS users (username VARCHAR(50) PRIMARY KEY, password  
VARCHAR(50));"
```

Paso 4. Para ver la base de datos, se hace desde phpMyAdmin, para ello, hay que ejecutarlo y se hace directamente un “pull”, sin necesidad de hacer “pull” en docker.

```
-> docker run -d --name phpmyadmin -p 5004:80 -e  
PMA_HOST=host.docker.internal -e PMA_PORT=3307 phpmyadmin/phpmyadmin
```

```
C:\Users\dawmi\servidor-daw\site3-db>docker run -d --name phpmyadmin -p 5004:80 -e PMA_HOST=host.docker.internal -e PMA_PORT=3307 phpmyadmin/phpmyadmin  
Unable to find image 'phpmyadmin/phpmyadmin:latest' locally  
latest: Pulling from phpmyadmin/phpmyadmin  
b5a8243c1a5e: Pull complete  
273a09302c4a: Pull complete  
30b083700c69: Pull complete  
fddb92e888a7: Pull complete  
004f06ab2f6c: Pull complete  
0befbf457381d: Pull complete  
a139c2f3234a: Pull complete  
dd53cf9bf4cf: Pull complete  
00ef78e422f0: Pull complete  
6571cfdbe5b2: Pull complete  
8d83c968ca9a: Pull complete  
0248257cbd51: Pull complete  
24403a1f6855: Pull complete  
e1cf44d6017a: Pull complete  
a8c81cca4b8b: Pull complete  
8c7716127147: Pull complete  
2489d5e860a7: Pull complete  
4f4fb700ef54: Pull complete  
749b92ea0995: Pull complete  
271012172ea0: Pull complete  
4eed3454c20c: Pull complete  
Digest: sha256:42a200db07b4e70fbf32c594ad4521cf16399b8e54bbb5adceae98e7566dfbeb  
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest  
a8b0ccb5096bcd5654dc93c6fe135a2ced25e386391fe15d2c52ce60b9bc1a4f  
  
C:\Users\dawmi\servidor-daw\site3-db>
```

Paso 5. Entrar a <http://localhost:5004> donde puedes poner:

Usuario - usuario

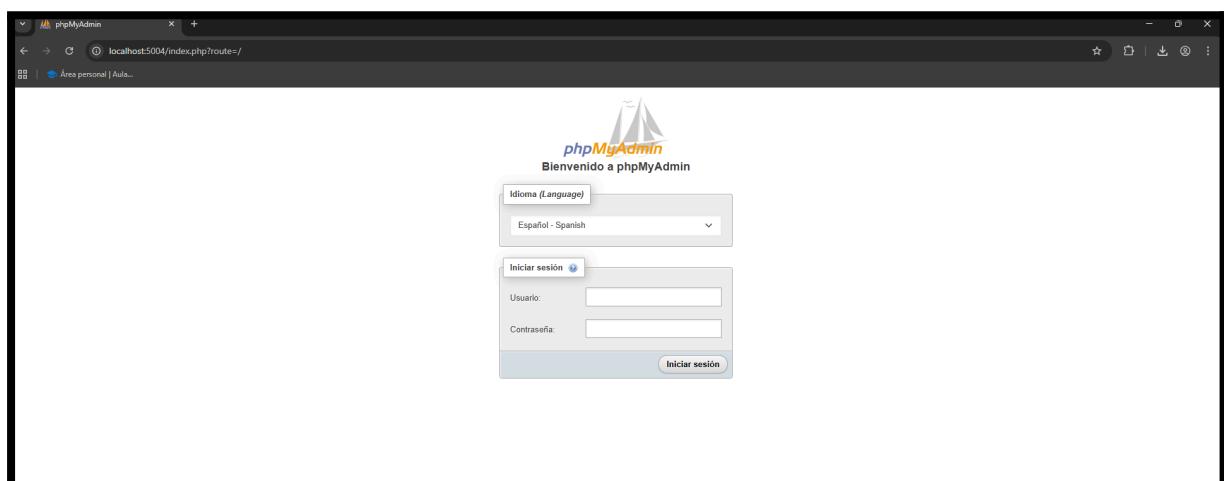
Contraseña - password

O bien entrar como:

Usuario - root

Contraseña - root

Y ahí veremos la base de datos, donde podremos saber si se insertan y eliminan los datos introducidos desde nuestro index.html



Aquí ponemos los datos ya dichos.

A screenshot of the phpMyAdmin interface showing the "users" table in the "information_schema" database. The left sidebar shows the database structure with "information_schema" expanded, revealing "users", "Nueva", "users", "Columnas", and "Índices". The main area has tabs for "Examinar", "Estructura", "SQL", "Buscar", and "Insertar", with "Examinar" selected. A green message box says "Mostrando filas 0 - 1 (total de 2, La consulta tardó 0.0029 segundos.)". Below it is a SQL query: "SELECT * FROM `users`". There are buttons for "Perfilando" and "Editar en línea". At the bottom, there are buttons for "Mostrar todo", "Número de filas: 25", and "Filtrar filas: Buscar". The table itself has columns "username" and "password". Two rows are listed: one for "dani" with password "7890" and another for "laura" with password "1234".

Podemos borrar desde la propia tabla o desde el index.html (localhost:7001)

The screenshot shows two consecutive pages from a web application.

Page 1: Insertar usuario

A form with three fields: "Usuario" (with value "dani"), "Contraseña" (empty), and a "Insertar" button.

Page 2: Borrar usuario

A form with one field "dani" and a "Borrar" button.

The screenshot shows a confirmation message after a user has been deleted.

Borrar Usuario

A message: "✓ Usuario borrado correctamente" (User deleted correctly).

Y comprobamos que se haya borrado.

The screenshot shows the "users" table in phpMyAdmin.

Table Structure:

username	password
laura	1234

SQL Query:

```
SELECT * FROM `users`
```

Y así se vería nuestro docker:

	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	● phpmyadmin	a8b0cbb5096b	phpmyadm	5004:80 ↗	<input type="checkbox"/> ⋮ trash
<input type="checkbox"/>	● site1-html	2f63ad12392b	site1-html	7001:80 ↗	<input type="checkbox"/> ⋮ trash
<input type="checkbox"/>	● site2-php	6607b0836e71	site2-php	7002:80 ↗	<input type="checkbox"/> ⋮ trash
<input type="checkbox"/>	● mariadb-daw	de6784ec8e60	mariadb:lat	3307:3306 ↗	<input type="checkbox"/> ⋮ trash