

MANUAL PARA IMPLEMENTAR WORDPRESS EN DOCKER

ÍNDICE

1. [Requisitos previos](#)
 2. [Creación del entorno de trabajo](#)
 3. [Archivo de configuración \(docker-compose.yml\)](#)
 4. [Inicio de los contenedores](#)
 5. [Verificación de la instalación](#)
 6. [Gestión de los contenedores](#)
 7. [Configuración opcional: phpMyAdmin](#)
-

1. REQUISITOS PREVIOS

Antes de comenzar, asegúrate de tener instalado en tu máquina:

- **Docker**
 - **Docker Compose**
 - Acceso a la terminal o línea de comandos.
 - Conocimientos básicos de Docker y manejo de contenedores.
-

2. CREACIÓN DEL ENTORNO DE TRABAJO

Abre una terminal y crea un directorio para el proyecto:

```
mkdir wordpress-docker  
cd wordpress-docker
```

Crea los directorios de volúmenes necesarios:

```
mkdir wordpress
```

3. ARCHIVO DE CONFIGURACIÓN (docker-compose.yml)

Dentro del directorio `wordpress-docker`, crea un archivo llamado `docker-compose.yml`:

```
leo@DESKTOP-Q8N983J:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/WordPress$ nano docker-compose.yml
```

Escribe el siguiente contenido:

```

1  version: '3.9'
2
3  ∨ services:
4  ∨   wordpress:
5      image: wordpress:latest
6      container_name: wordpress_container
7  ∨   ports:
8      - "8080:80"
9  ∨   environment:
10      WORDPRESS_DB_HOST: db
11      WORDPRESS_DB_USER: wordpress
12      WORDPRESS_DB_PASSWORD: wordpress_password
13      WORDPRESS_DB_NAME: wordpress_db
14  ∨   volumes:
15      - ./wordpress:/var/www/html
16  ∨   depends_on:
17      - db
18
19  ∨   db:
20      image: mysql:5.7
21      container_name: wordpress_db
22  ∨   environment:
23      MYSQL_DATABASE: wordpress_db
24      MYSQL_USER: wordpress
25      MYSQL_PASSWORD: wordpress_password
26      MYSQL_ROOT_PASSWORD: root_password
27  ∨   volumes:
28      - db_data:/var/lib/mysql
29
30  ∨ volumes:
31      db_data:
32
33

```

4. INICIO DE LOS CONTENEDORES

Para iniciar los servicios definidos en el archivo `docker-compose.yml`, ejecuta:

```

leo@DESKTOP-Q0N9833:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/WordPress$ docker-compose up -d

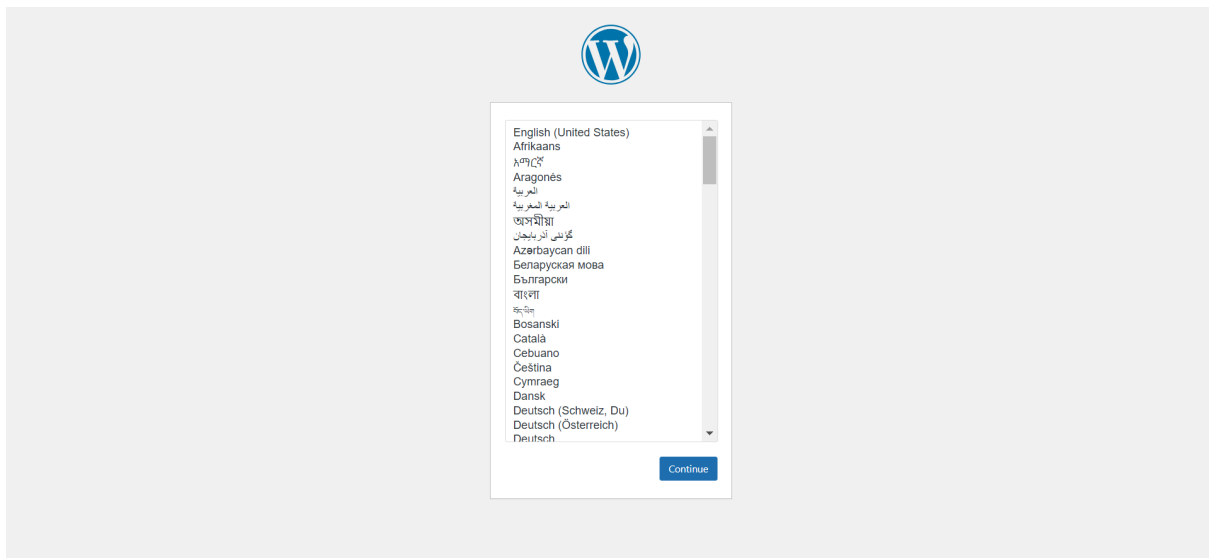
```

1.

- Este comando descargará las imágenes de Docker necesarias (si no están en tu sistema), creará los contenedores y los ejecutará en segundo plano.

5. VERIFICACIÓN DE LA INSTALACIÓN

1. Abre un navegador web.
2. Visita la URL: `http://localhost:8080`
2. Deberías ver la página de configuración de WordPress.
3. Sigue los pasos en pantalla para completar la instalación:
 - Selecciona el idioma.



- Introduce el nombre del sitio, usuario, contraseña y correo electrónico.

Hola

¡Este es el famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario

Contraseña

EboLeXRWYI&iR\$cggñ

Ocultar

Fuerte

Tu correo electrónico

Visibilidad en los motores de búsqueda

☐ Pedir a los motores de búsqueda que no indexen este sitio

Depende de los motores de búsqueda atender esta petición o no.

Instalar WordPress

6. GESTIÓN DE LOS CONTENEDORES

Listar contenedores activos

Para verificar qué contenedores están ejecutándose:

```
leo@DESKTOP-Q0N983J:/mnt/c/Users/User/Documents/Asir_2/Aplicaciones Web/XAMPP/htdocs/docker-projects/wordpress$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
AMES					
1e2870187e4e	wordpress:latest	"docker-entrypoint.s..."	5 seconds ago	Up 4 seconds	0.0.0.0:8080->80/tcp, :::8080->80/tcp
ordpress_container					
37e6451c7cd8	mysql:5.7	"docker-entrypoint.s..."	7 seconds ago	Up 5 seconds	3306/tcp, 33060/tcp
ordpress_db					
b3eab1d2cd7e	sdelements/lets-chat	"npm start"	22 hours ago	Up 23 minutes	5222/tcp, 0.0.0.0:80->8080/tcp, :::80->8080/tcp
etschat					
7dc8e736fb16	mongo:3.4	"docker-entrypoint.s..."	22 hours ago	Up 23 minutes	27017/tcp
ongo					

Detener los contenedores

Para detener todos los servicios:

```
leo@DESKTOP-Q0N983J:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/WordPress$ docker-compose down
Stopping wordpress_container ... done
Stopping wordpress_db ... done
Removing wordpress_container ... done
Removing wordpress_db ... done
Removing network wordpress_default
```

Ver logs de un contenedor

Para revisar los logs de un contenedor específico (por ejemplo, el de WordPress):

```
docker logs wordpress_container
```

Reiniciar un contenedor

Si necesitas reiniciar solo un contenedor:

```
leo@DESKTOP-Q0N983J:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/WordPress$ docker restart wordpress_container
wordpress_container
```

7. CONFIGURACIÓN OPCIONAL: PHPMYADMIN

Si deseas agregar **phpMyAdmin** para gestionar la base de datos de forma gráfica:

Edita el archivo `docker-compose.yml` y añade lo siguiente al final:

```
phpmyadmin:
  image: phpmyadmin/phpmyadmin
  container_name: phpmyadmin_container
  ports:
    - "8081:80"
  environment:
    PMA_HOST: db
    MYSQL_ROOT_PASSWORD: root_password
  depends_on:
    - db
```

Guarda el archivo y vuelve a levantar los contenedores:

```
docker-compose up -d
```

1.

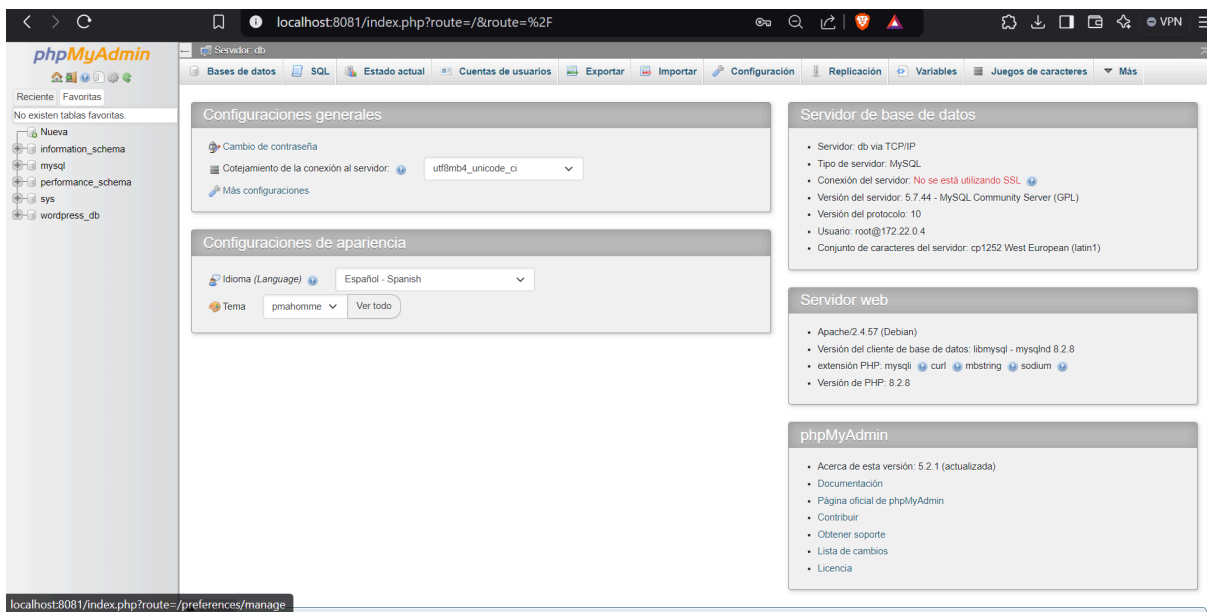
Accede a phpMyAdmin en tu navegador:

`http://localhost:8081`

2.

3. Usa las siguientes credenciales para iniciar sesión:

- **Servidor:** `db`
- **Usuario:** `root`
- **Contraseña:** `root_password`



8. NOTAS IMPORTANTES

- **Persistencia de datos:**
Los datos del sitio web (archivos de WordPress) y de la base de datos se almacenan en los volúmenes:
 - Archivos de WordPress: `./wordpress`
 - Datos de MySQL: `db_data`
- **Cambios en los puertos:**
Si `8080` o `8081` ya están ocupados en tu máquina, cámbialos en el archivo `docker-compose.yml`.
- **Actualización de WordPress:**
Si deseas actualizar WordPress, asegúrate de usar la última versión de la imagen (`wordpress:latest`).

