



# **INFRAESTRUCTURA WORDPRESS**

IAW

Gabriela Mendoza  
2º ASIR

## Índice

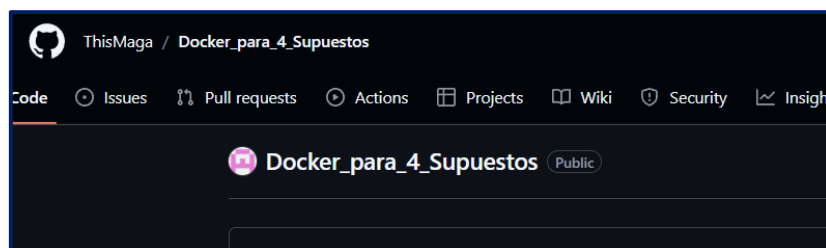
Pasos para crear la infraestructura .....	2
1. Repositorio en Github .....	2
2. Creamos la ruta de Wordpress.....	2
3. Iniciar los contenedores .....	5
4. Verificamos Wordpress .....	5
5. Para parar los servicios.....	8
6. Actualizar los commit en GitHub con los cambios realizados .....	9

# MANUAL DE IMPLANTACIÓN PARA WORDPRESS CON DOCKER

## Pasos para crear la infraestructura

### 1. Repositorio en Github

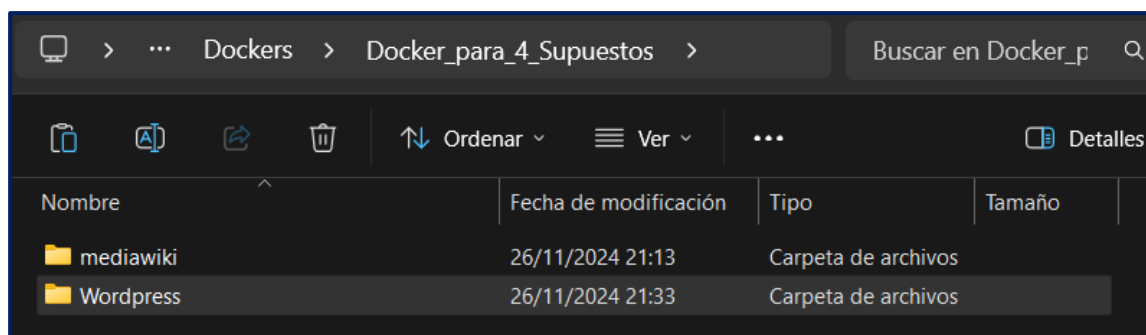
En esta ocasión utilizaremos el mismo repositorio, con diferencia que crearemos otra carpeta para crear su directorio.



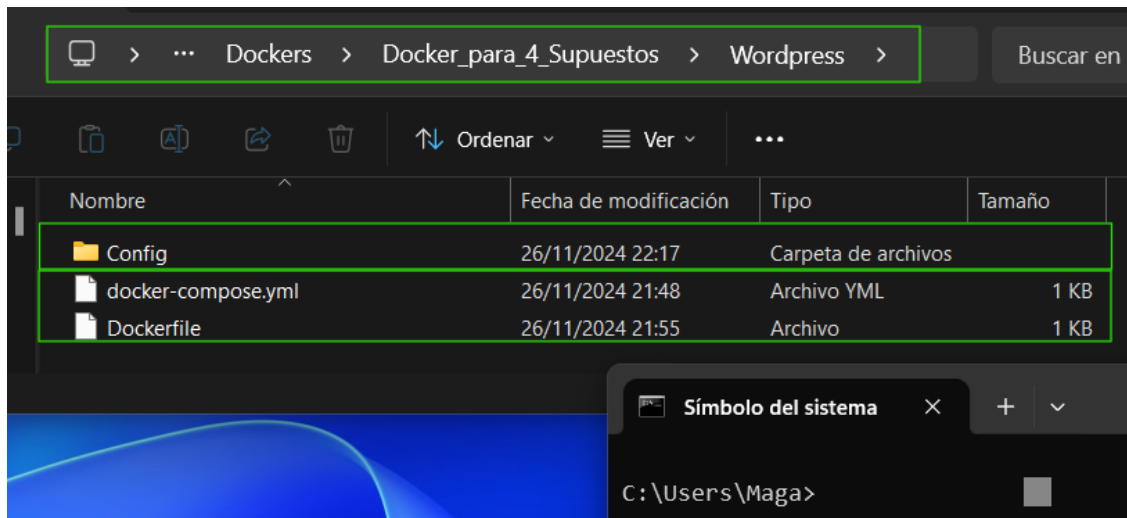
### 2. Creamos la ruta de Wordpress

Para crear la estructura, lo primero es crear un directorio en la ruta donde vamos a trabajar.

En mi caso la creare en el Disco **C:\Docker\Docker\_para\_4\_Supuestos**, creamos la carpeta Wordpress



En el agregaremos los siguientes ficheros:



Procedemos a abrir el VisualCode con la ruta para agregar el contenido a los ficheros.

- Fichero docker-compose.yml, que define los servicios necesarios para ejecutar WordPress (la aplicación y la base de datos).

```
docker-compose.yml X
Wordpress > docker-compose.yml
1  version: '3.8'
2
3  services:
4    wordpress:
5      image: wordpress:latest
6      container_name: wordpress
7      ports:
8        - "8081:80" # WordPress estará disponible en http://localhost:8081
9      environment:
10       - WORDPRESS_DB_HOST=db
11       - WORDPRESS_DB_USER=wordpress
12       - WORDPRESS_DB_PASSWORD=password
13       - WORDPRESS_DB_NAME=wordpress
14      volumes:
15       - wordpress_data:/var/www/html
16      depends_on:
17       - db
18      restart: always
19
20    db:
21      image: mariadb:latest
22      container_name: wordpress_db
23      environment:
24       - MYSQL_ROOT_PASSWORD=rootpassword
25       - MYSQL_DATABASE=wordpress
26       - MYSQL_USER=wordpress
27       - MYSQL_PASSWORD=password
28      volumes:
29       - db_data:/var/lib/mysql
30      restart: always
31
32  volumes:
33    wordpress_data:
34    db_data:
```

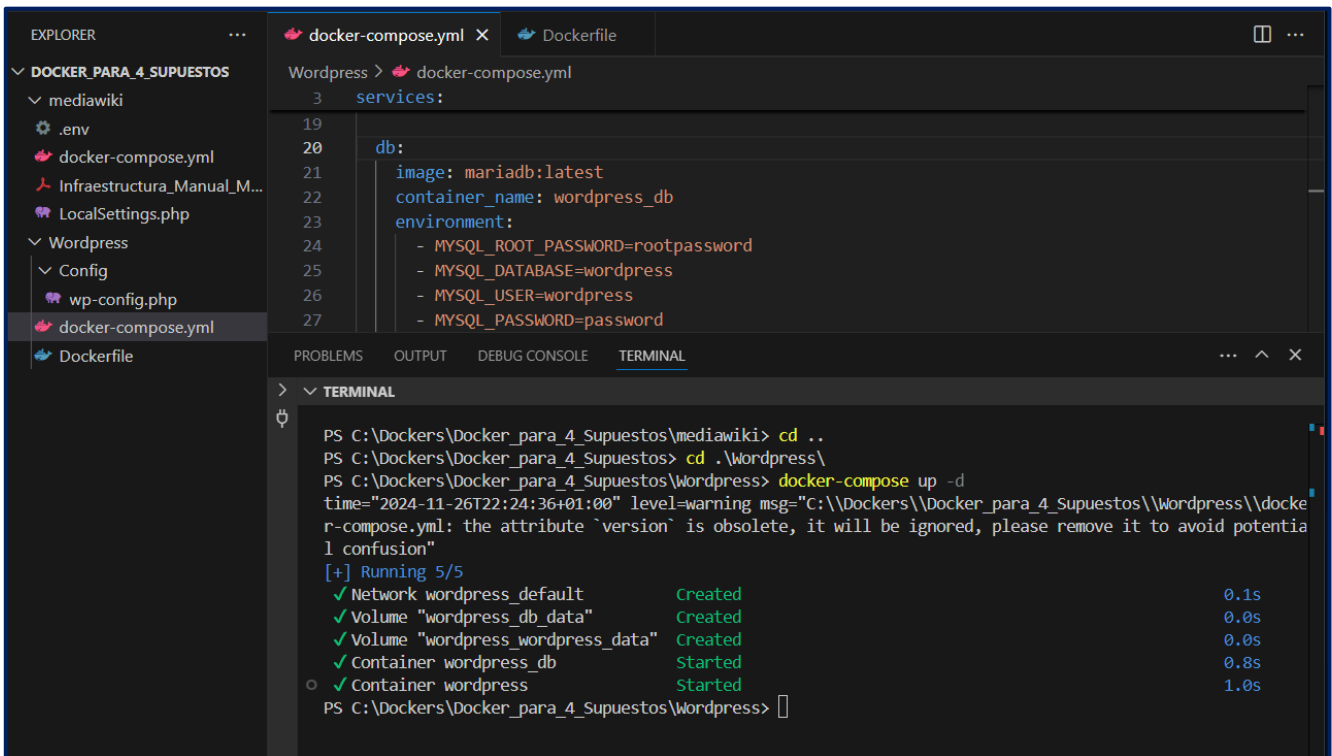
→ En caso de querer modificar el Wordpress, podemos tener el fichero de configuración

```
Wordpress > Dockerfile
1  FROM wordpress:latest
2
3  # Instalar dependencias adicionales (opcional)
4  RUN apt-get update && apt-get install -y \
5      vim \
6      curl
7
8  # Copiar configuraciones personalizadas (si es necesario)
9  # COPY ./config/wp-config.php /var/www/html/wp-config.php
10
```

- Una configuración adicional, es crear una carpeta llamada Config, donde se pueden guardar archivos personalizados, como wp-config.php. De ser necesario configurar parámetros avanzados, por ejemplo, se crea el archivo wp-config.php aquí y actualiza el Dockerfile para copiarlo al contenedor.

### 3. Iniciar los contenedores

- Nos movemos a la ruta de Wordpress y ejecutamos **docker-compose up -d**



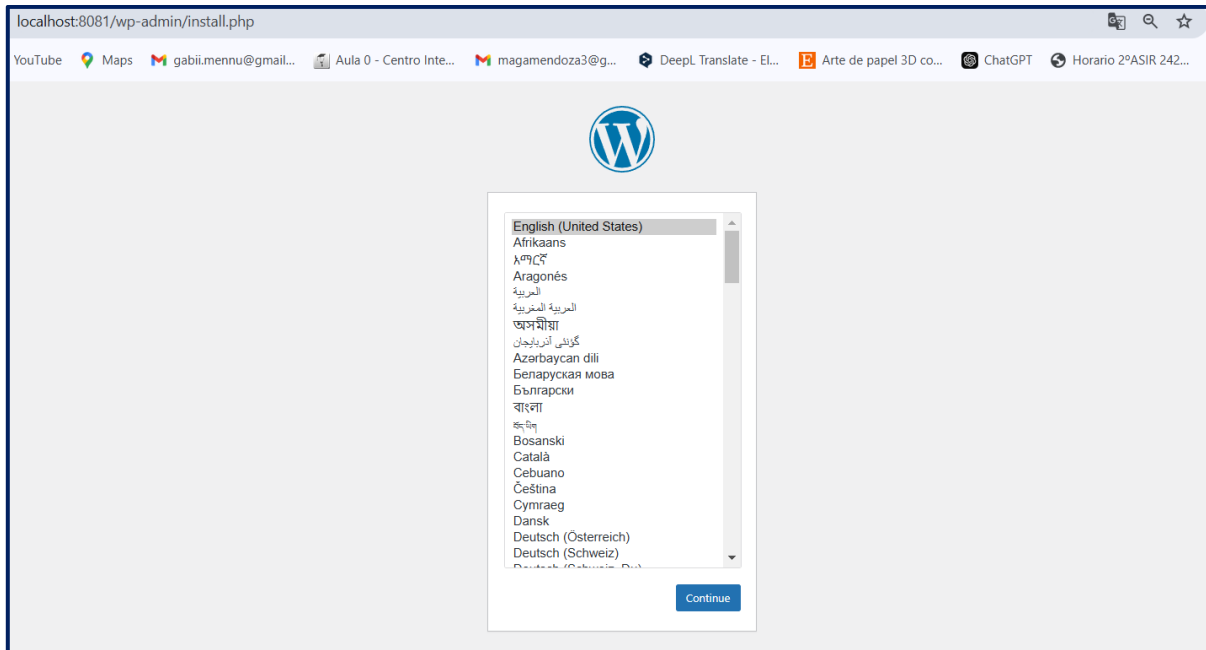
The screenshot shows a Visual Studio Code editor with two tabs: 'docker-compose.yml' and 'Dockerfile'. The 'docker-compose.yml' file is open, showing the 'services' section with a 'db' service configured with 'image: mariadb:latest', 'container\_name: wordpress\_db', and environment variables for MySQL root password, database name, user, and password. The 'Dockerfile' tab is also visible. Below the editor, the 'TERMINAL' panel shows the execution of the 'docker-compose up -d' command in the 'Wordpress' directory. The output indicates that the network 'wordpress\_default' and volumes 'wordpress\_db\_data' and 'wordpress\_wordpress\_data' were created, and the containers 'wordpress\_db' and 'wordpress' were started successfully.

```
Wordpress > docker-compose.yml
3 services:
19
20 db:
21   image: mariadb:latest
22   container_name: wordpress_db
23   environment:
24     - MYSQL_ROOT_PASSWORD=rootpassword
25     - MYSQL_DATABASE=wordpress
26     - MYSQL_USER=wordpress
27     - MYSQL_PASSWORD=password

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
> TERMINAL
PS C:\Dockers\Docker_para_4_Supuestos\mediawiki> cd ..
PS C:\Dockers\Docker_para_4_Supuestos> cd .\Wordpress\
PS C:\Dockers\Docker_para_4_Supuestos\Wordpress> docker-compose up -d
time="2024-11-26T22:24:36+01:00" level=warning msg="C:\\Dockers\\Docker_para_4_Supuestos\\Wordpress\\docke
r-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potentia
l confusion"
[+] Running 5/5
 ✓ Network wordpress_default      Created      0.1s
 ✓ Volume "wordpress_db_data"     Created      0.0s
 ✓ Volume "wordpress_wordpress_data" Created      0.0s
 ✓ Container wordpress_db        Started      0.8s
 ○ Container wordpress           Started      1.0s
PS C:\Dockers\Docker_para_4_Supuestos\Wordpress> 
```

### 4. Verificamos Wordpress

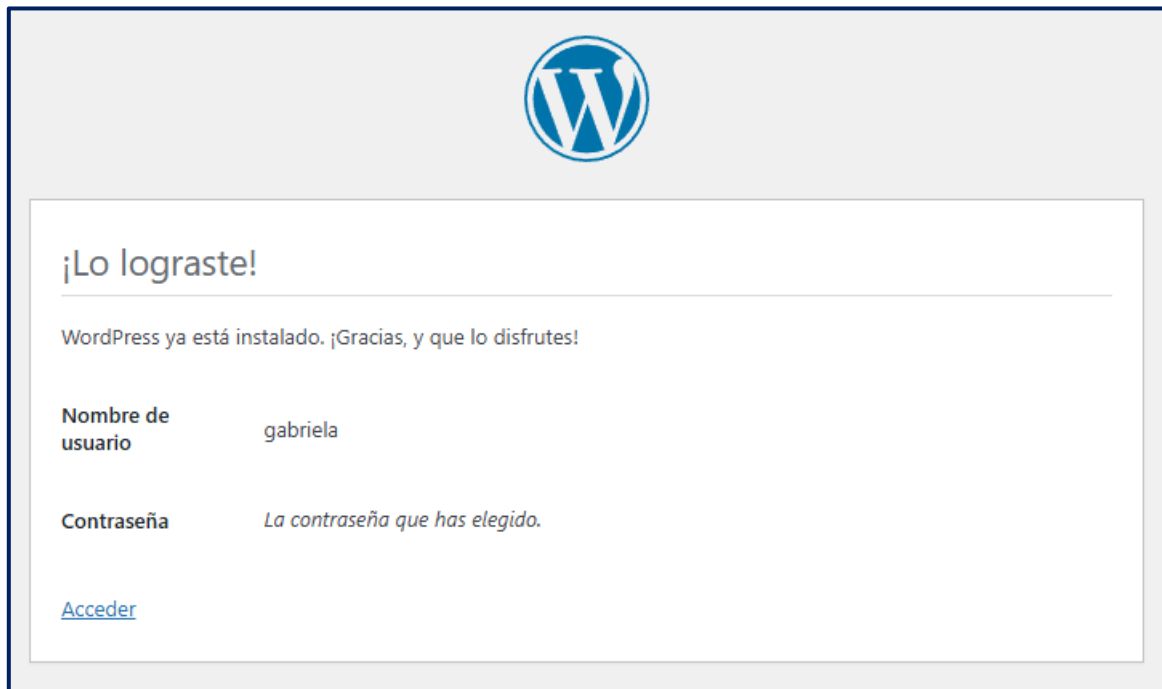
- Una vez, ejecutado el docker-compose nos vamos al navegador, y colocamos: <http://localhost:8081>, para verificar la descarga.



- En efecto, verificamos que tenemos la aplicación del Wordpress, para completar la instalación, seguimos los pasos de la pantalla.
- En este punto he añadido el usuario y contraseña de:
  - Gabriela
  - ediuePo1Ihn&(c0lCr

A screenshot of the WordPress installation page, specifically the 'Information needed' section. The page has a light gray header with the WordPress logo. Below the header, there's a section titled 'Hola' with a welcome message. The main section is titled 'Información necesaria' and contains a form with several fields: 'Título del sitio' (Site title) with the value 'Hola Mundo!!', 'Nombre de usuario' (Username) with the value 'gabriela', 'Contraseña' (Password) with the value 'ediuePo1Ihn&(c0lCr', and 'Tu correo electrónico' (Email) with the value 'magamendoza3@gmail.com'. There are also checkboxes for 'Visibilidad en los motores de búsqueda' (Search engines) and a button 'Instalar WordPress' at the bottom.

→ Procedemos a instalar Wordpress

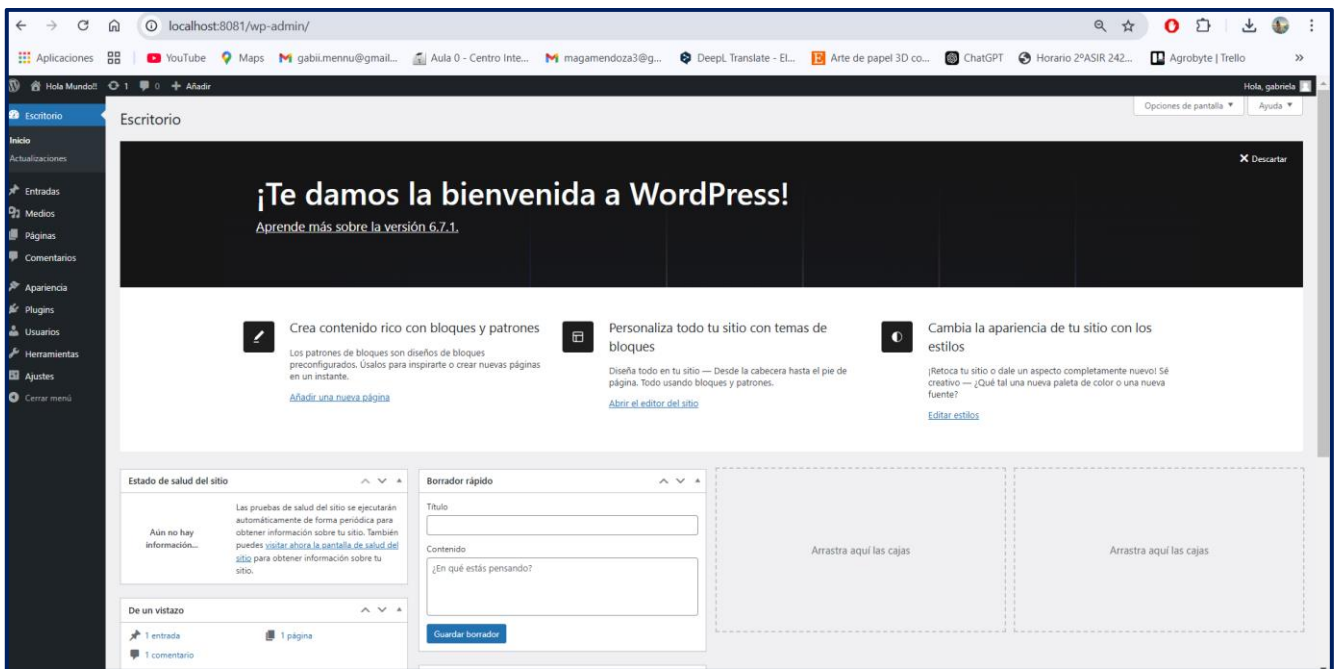


→ Accedemos con el usuario creado:

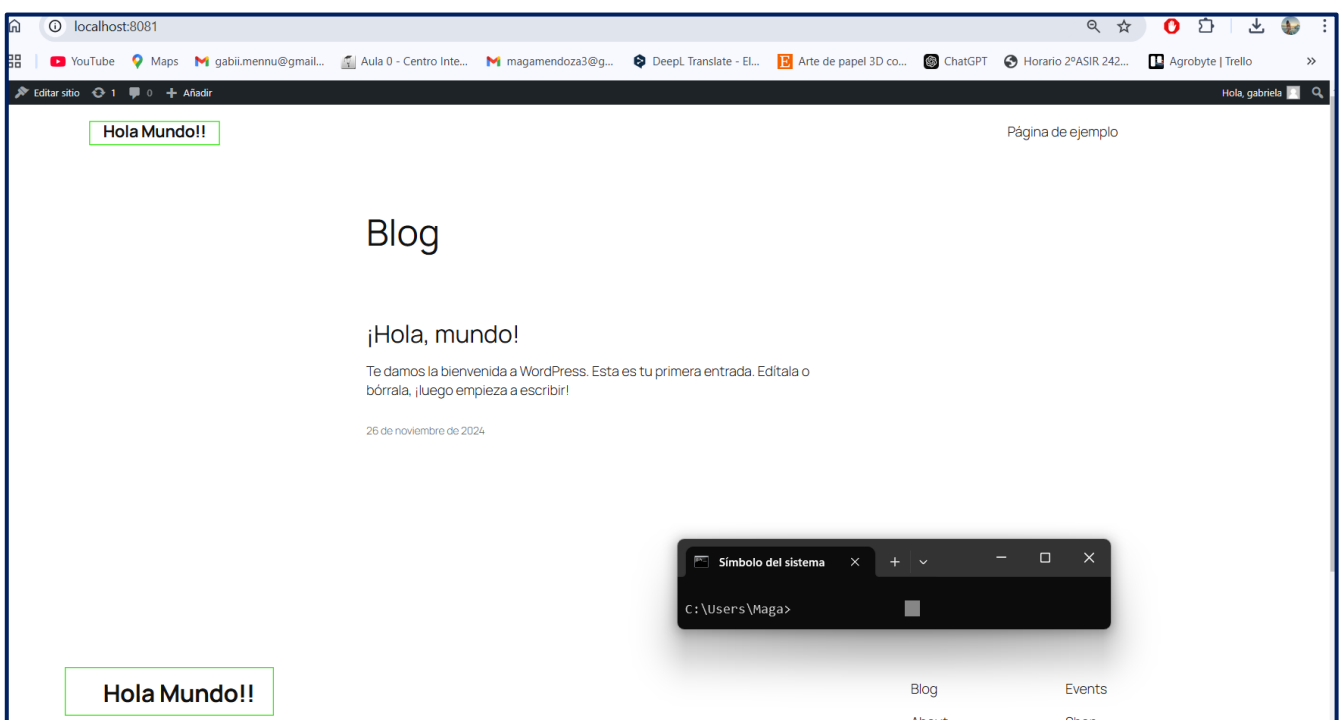


→ Comprobamos que tenemos acceso y podemos iniciar sesión



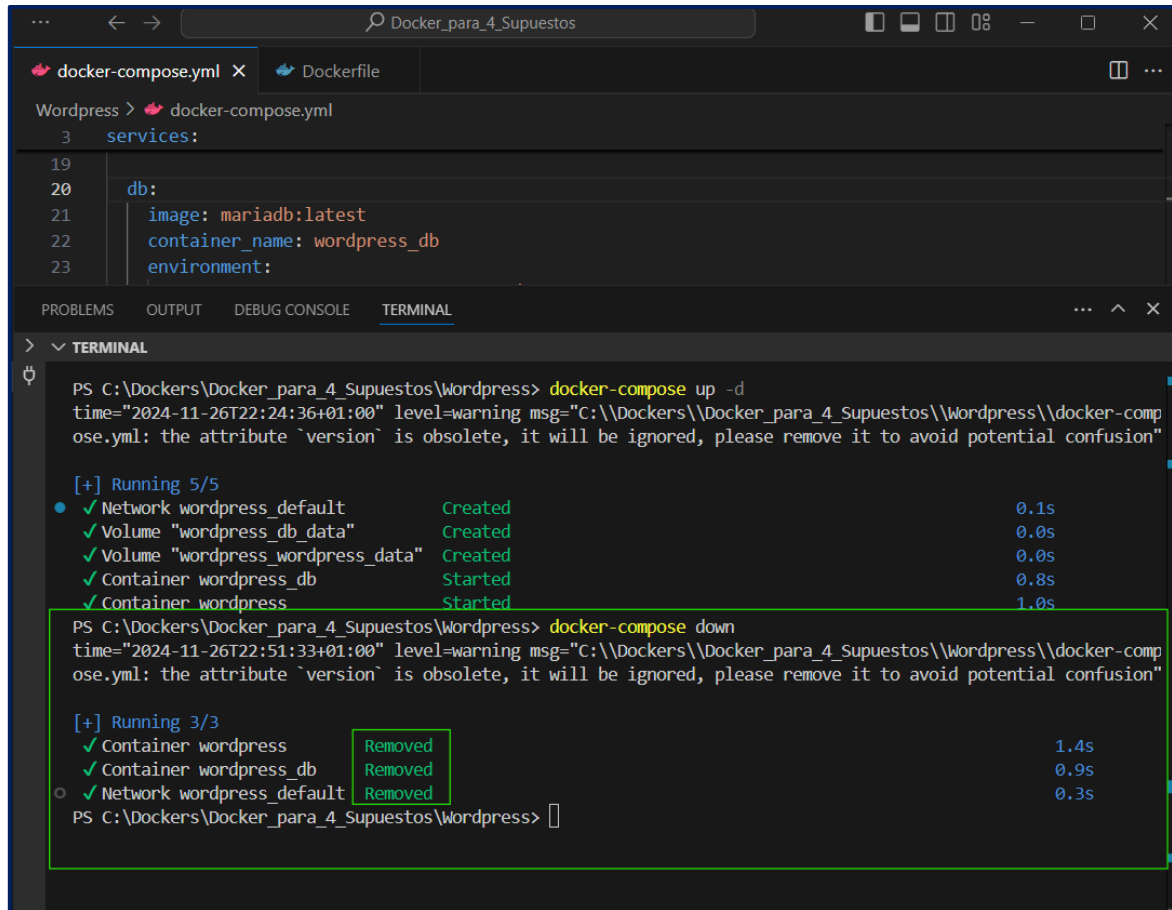


→ Buscamos la entrada con el título que hemos creado, el “Hola Mundo!!” Es el título que he creado anteriormente.



## 5. Para parar los servicios

Para detener los contenedores ejecutamos **docker-compose down**



The screenshot shows a VS Code editor with a file named `docker-compose.yml` open. The file content is as follows:

```
services:
  db:
    image: mariadb:latest
    container_name: wordpress_db
    environment:
```

Below the editor, the `TERMINAL` tab is active, showing the output of the `docker-compose up -d` and `docker-compose down` commands. The output for `up -d` shows the creation of a network, two volumes, and two containers. The output for `down` shows the removal of the same components. A green box highlights the `down` output.

```
PS C:\Dockers\Docker_para_4_Supuestos\Wordpress> docker-compose up -d
time="2024-11-26T22:24:36+01:00" level=warning msg="C:\\Dockers\\Docker_para_4_Supuestos\\Wordpress\\docker-comp
ose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"

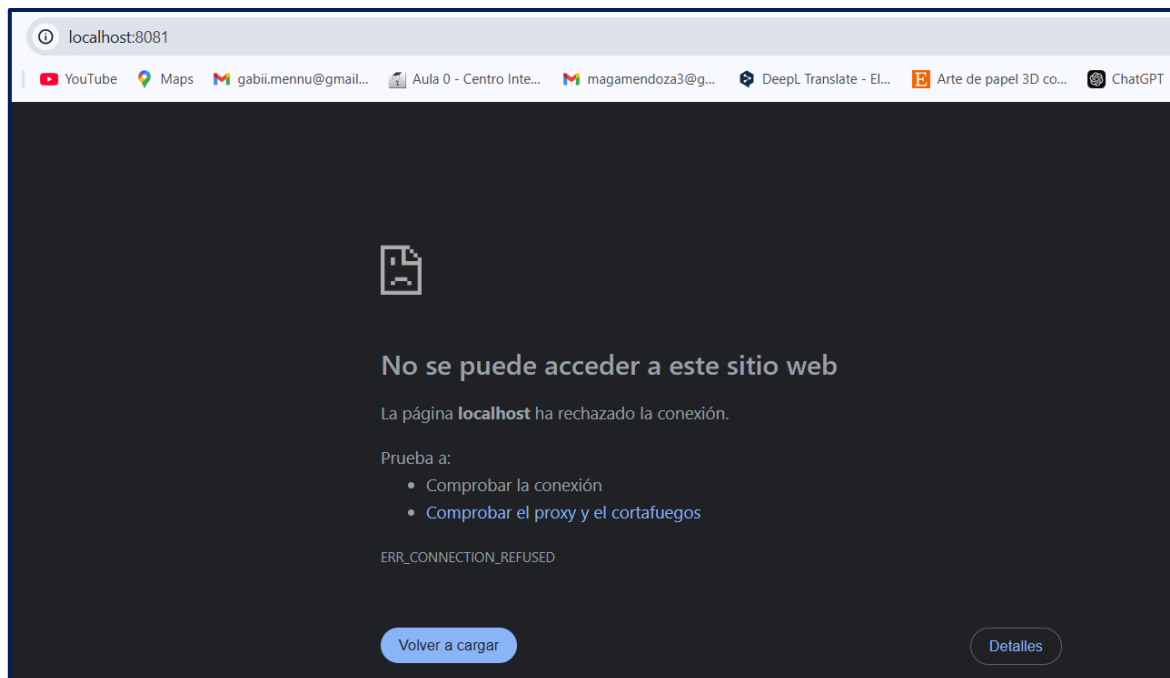
[+] Running 5/5
 ✓ Network wordpress_default          Created           0.1s
 ✓ Volume "wordpress_db_data"        Created           0.0s
 ✓ Volume "wordpress_wordpress_data" Created           0.0s
 ✓ Container wordpress_db            Started           0.8s
 ✓ Container wordpress               Started           1.0s

PS C:\Dockers\Docker_para_4_Supuestos\Wordpress> docker-compose down
time="2024-11-26T22:51:33+01:00" level=warning msg="C:\\Dockers\\Docker_para_4_Supuestos\\Wordpress\\docker-comp
ose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"

[+] Running 3/3
 ✓ Container wordpress               Removed           1.4s
 ✓ Container wordpress_db            Removed           0.9s
 ✓ Network wordpress_default          Removed           0.3s

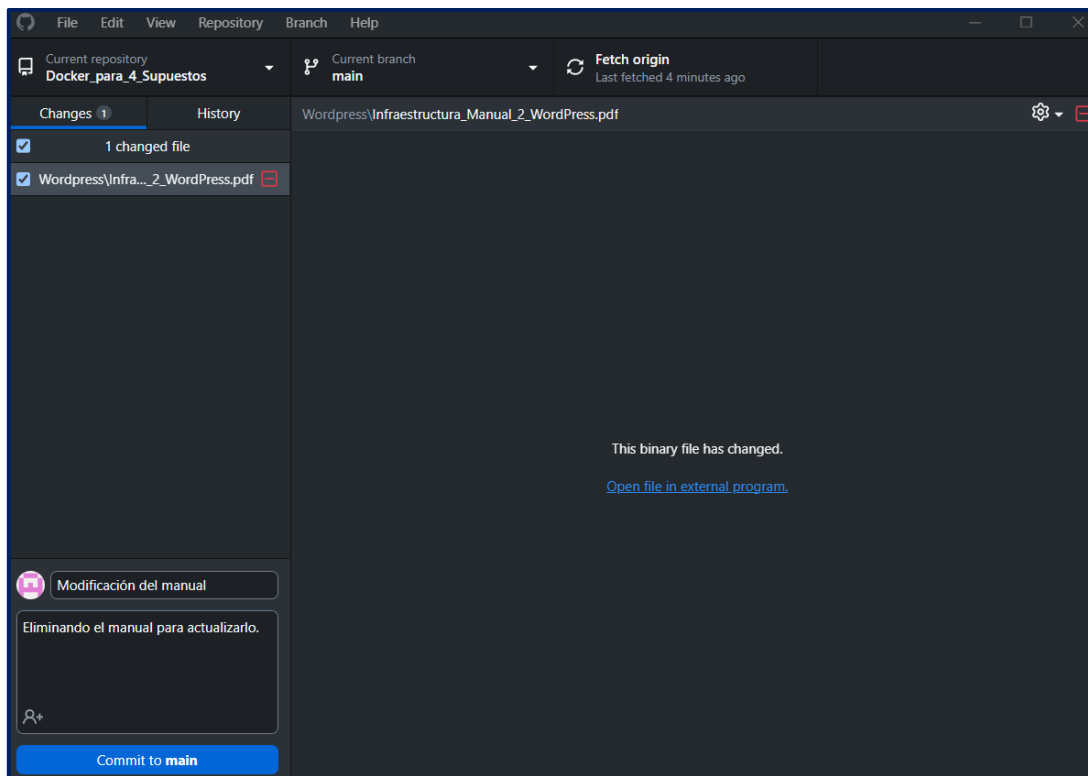
PS C:\Dockers\Docker_para_4_Supuestos\Wordpress>
```

→ Después de para el contenedor, vemos que la página de WordPress no carga.

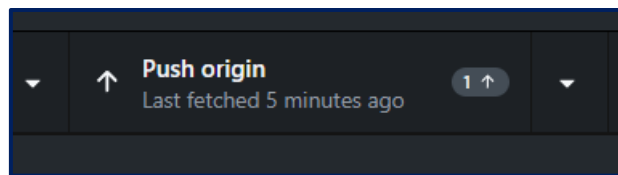


## 6. Actualizar los commit en GitHub con los cambios realizados

No, olvides actualizar los commit en github, con los cambios realizados en cualquier fichero.



→ Cargar, hacer un push



→ Verificar los cambios en GitHub

