

# Implantación de Backend y Frontend en Local

AWS - EBS

Gabriela Mendoza

2º ASIR  
IAW

## ÍNDICE

<b>Implementación de backend y frontend en Local .....</b>	<b>2</b>
1. Creación de base de datos .....	2
2. Creación del backend .....	3
3. Creación del frontend.....	4

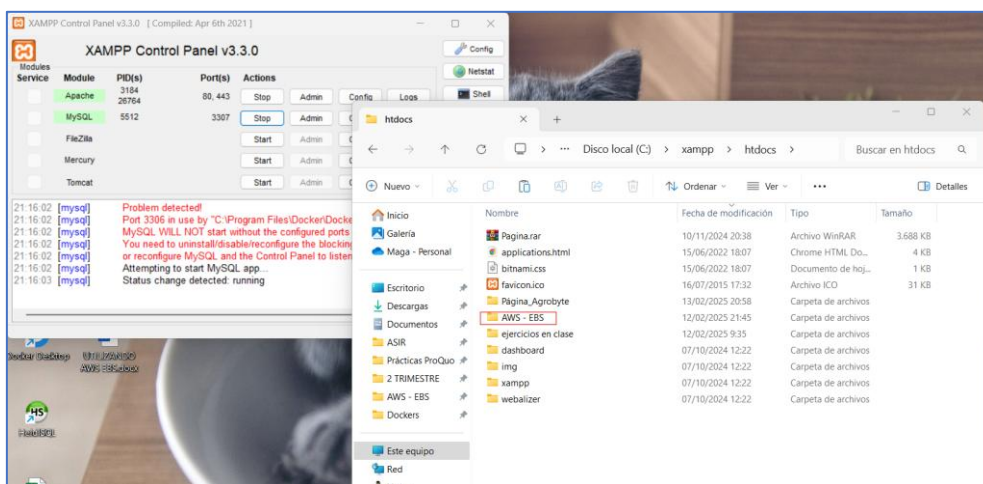
# Implementación de backend y frontend en Local

En esta sección, se describe la implementación del backend y frontend de una aplicación en un entorno local. El backend es responsable de la lógica del servidor y la gestión de los datos, mientras que el frontend se encarga de la interfaz de usuario y la experiencia visual. Antes de comenzar con la implementación en la nube, es necesario configurar y preparar ambos componentes localmente, empezando por la creación de la base de datos donde se almacenará la información de la página web. A continuación, se detallará el proceso paso a paso para lograrlo.

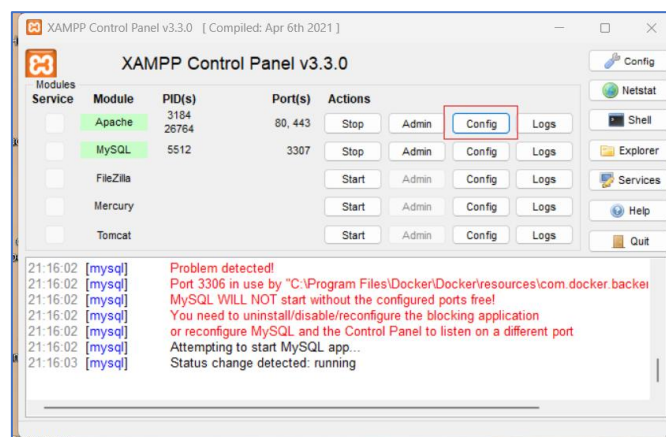
## 1. Creación de base de datos

Para crear la base de datos, ubicaremos la ruta de nuestro directorio, dentro de xampp, htdocs.

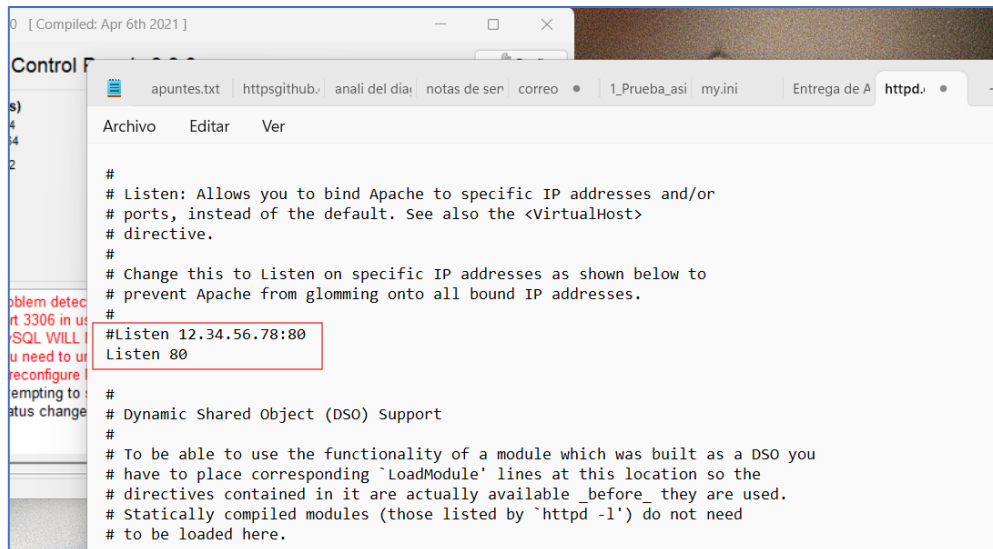
Abrimos el xampp e iniciamos apache y MySQL.



Visualizamos el Listen del archivo de configuración de apache.



Vemos que tenemos el 80

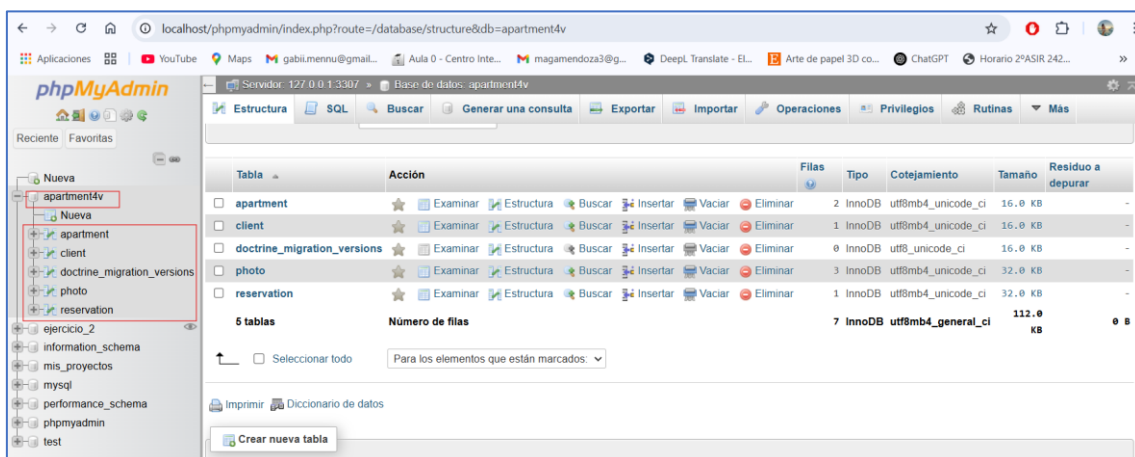


```
[Compiled: Apr 6th 2021]

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
```

Seguidamente, abrimos MySql desde xampp para crear la base de datos.



Una vez tengamos la base de datos, procedemos a enlazar con el dockerfile. Para ello debemos tener la aplicación de Docker corriendo y abrir el vscode.

## 2. Creación del backend

Comprobamos los puertos, es decir que no estén siendo usados por otros contenedores, y en caso de estar ocupados en otros contenedores, podemos cambiar el puerto o detener los contenedores que no necesitamos, el cual es mi caso.

Después de detener todos los contenedores, ejecutamos el *docker-compose up -d* para crear la imagen del contenedor.

The screenshot shows the VS Code interface with the Explorer panel on the left displaying the file structure of the 'AWS - EBS' project. The 'Backend' folder is expanded, showing files like 'bin', 'config', 'migrations', 'public', 'src', 'var', 'vendor', '.env', '.gitignore', 'composer.json', 'composer.lock', 'docker-compose.yml', 'Dockerfile', and 'symfony.lock'. The 'docker-compose.yml' file is selected and its content is visible in the main editor. The 'Dockerfile' file is also visible in the Explorer panel. The terminal panel at the bottom shows the output of the 'docker-compose up -d' command, indicating that the 'backend-db-1' and 'backend-backend-1' containers have started successfully.

```
3 services:
4   backend:
10    environment:
12      - DATABASE_URL=mysql://root:password@db:3307/symfony
13    volumes:
14      - ./var/www/html
15    depends_on:
16      - db
17
18 db:
19   image: mariadb:latest
20   environment:
21     MYSQL_ROOT_PASSWORD: password
22     MYSQL_DATABASE: symfony
23   ports:
24     - "3308:3307"
```

```
PS C:\xampp\htdocs\AWS - EBS\Backend> docker-compose up -d
time="2025-02-16T22:34:25+01:00" level=warning msg="C:\xampp\htdocs\AWS - EBS\Backend\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 2/2
  ✓ Container backend-db-1 Started
  ✓ Container backend-backend-1 Started
PS C:\xampp\htdocs\AWS - EBS\Backend>
```

### 3. Creación del frontend

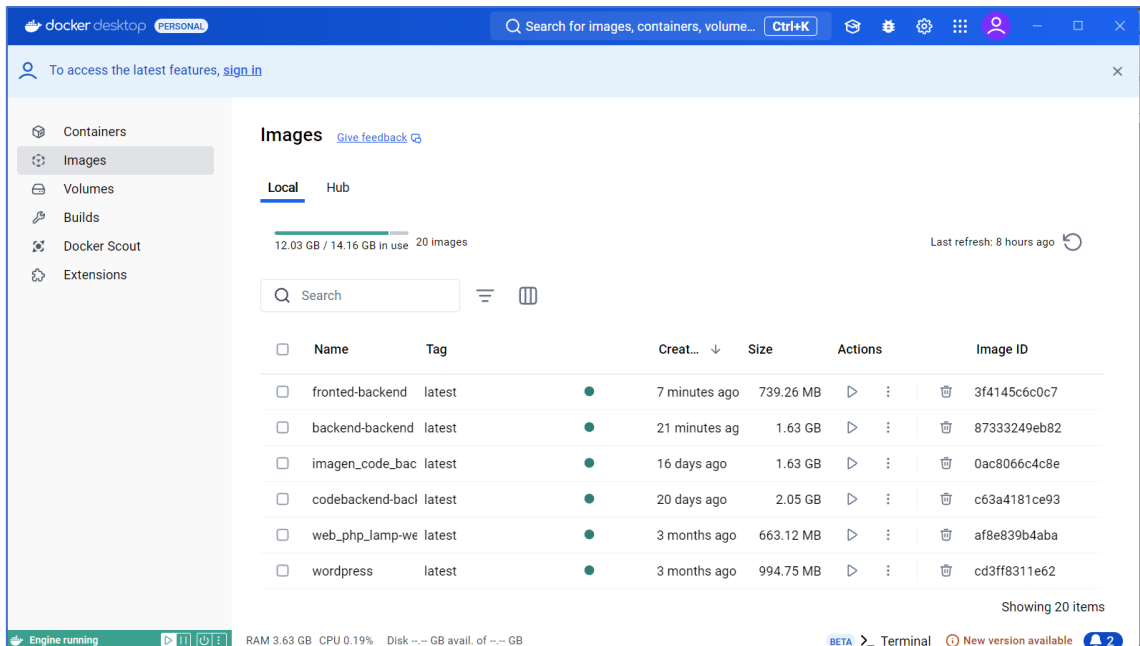
Ejecutamos el mismo proceso con el Frontend

The screenshot shows the VS Code interface with the Explorer panel on the left displaying the file structure of the 'AWS - EBS' project. The 'Frontend' folder is expanded, showing files like 'Code', 'Dockerfile', 'Dockerfile.zip', 'Apuntes EBS.txt', and 'init.sql'. The 'docker-compose.yml' file is selected and its content is visible in the main editor. The 'Dockerfile' file is also visible in the Explorer panel. The terminal panel at the bottom shows the output of the 'docker-compose up -d' command, indicating that the 'fronted-db-1' and 'fronted-backend-1' containers have started successfully. The output also shows the export of the manifest and the resolution of the provenance for the metadata file.

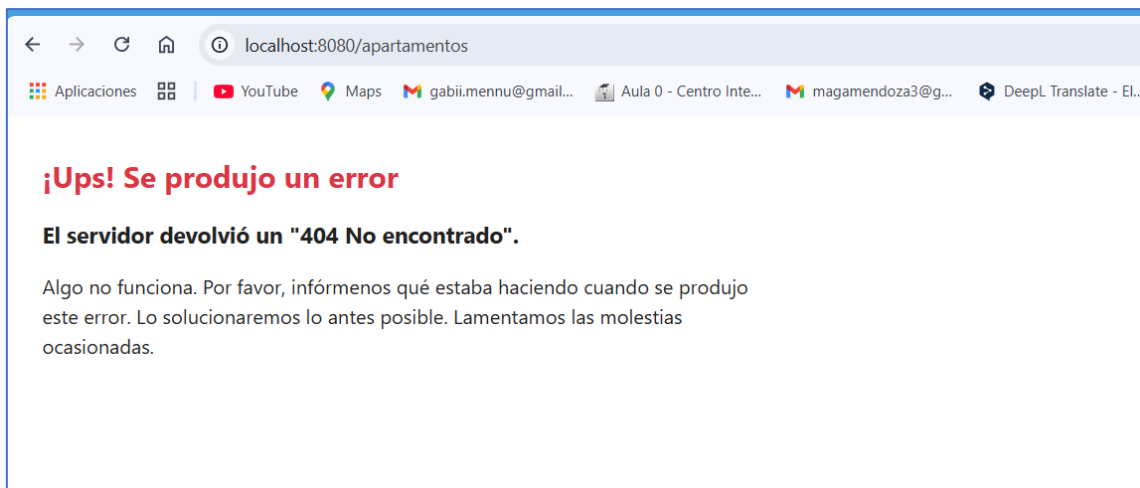
```
3 services:
4   backend:
10    environment:
12      - DATABASE_URL=mysql://root:password@db:3307/symfony
13    volumes:
14      - ./var/www/html
15    depends_on:
16      - db
17
18 db:
19   image: mariadb:latest
20   environment:
21     MYSQL_ROOT_PASSWORD: password
22     MYSQL_DATABASE: symfony
23   ports:
24     - "3308:3307"
```

```
PS C:\xampp\htdocs\AWS - EBS\Frontend> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     STATUS    PORTS                               NAMES
7c08e6495668   backend-backend  "docker-php-entryptoi..." 14 minutes ago Up 11 minutes 0.0.0.0:8080->80/tcp   backend-backend-1
4d578afa7a40   mariadb:latest  "docker-entrypoint.s..." 14 minutes ago Up 11 minutes 0.0.0.0:3306->3306/tcp backend-db-1
PS C:\xampp\htdocs\AWS - EBS\Frontend>
```

Comprobamos las imágenes en Docker.



Para finalizar comprobamos la pagina web, con un localhost:8080/apartamentos, que es el puerto que hemos usado.



Visto en clase, nos daba un error al ejecutar la web. Tanto con el puerto como sin él.

