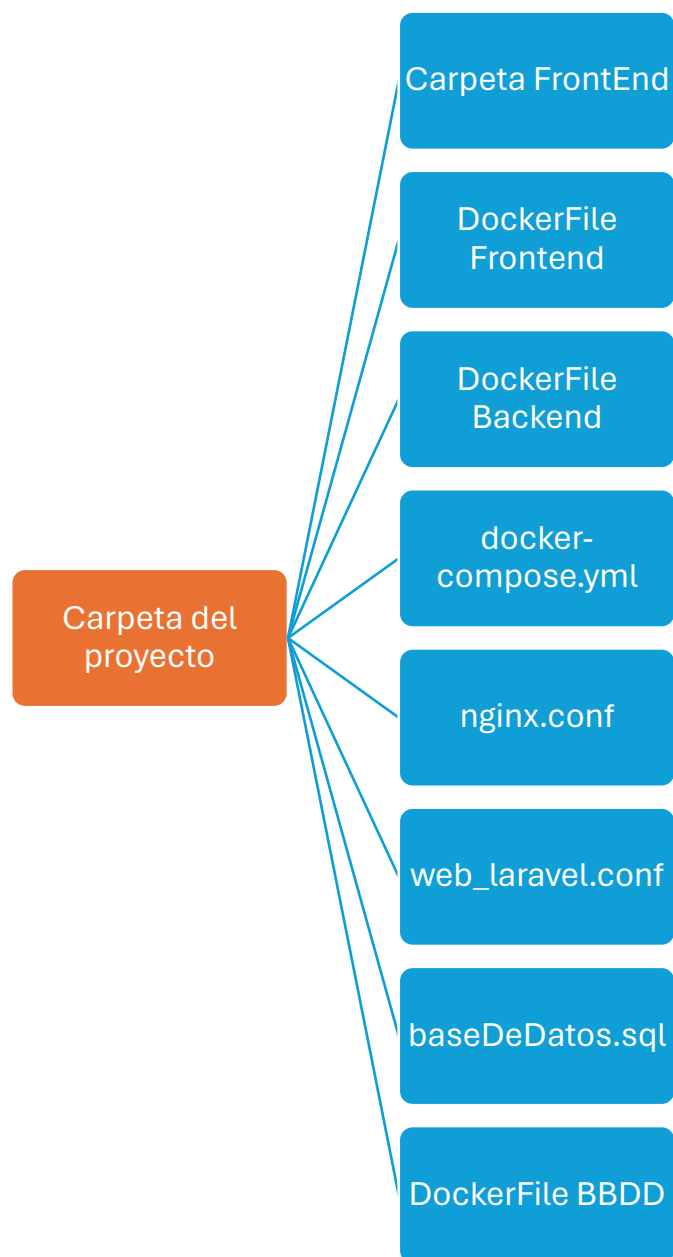


PUESTA EN MARCHA. DESPLIEGUE

4.1 Guía del despliegue del proyecto

1) Nuestra estructura de carpetas debe ser la siguiente:



2) Crear DockerFile para el backend.

Ten en cuenta a la hora de clonar el repositorio poner tu nombre de usuario, tu Personal Access Token que podrás encontrar o crear en tu cuenta de GitHub y el repositorio de backend de donde lo quieres clonar.

```
Dockerfile-backend > ...
1 # Usa la imagen base de PHP con Apache
2 FROM php:8.2-apache
3
4 # Establece el directorio de trabajo
5 WORKDIR /var/www/html
6
7 # Actualiza los paquetes e instala dependencias necesarias
8 RUN apt-get update && \
9     apt-get install -y git zlib1g-dev libzip-dev && \
10     rm -rf /var/lib/apt/lists/*
11
12 # Instala extensiones PHP necesarias
13 RUN docker-php-ext-install zip pdo pdo_mysql
14
15 # Clona el repositorio
16 RUN git clone https://'nombre usuario GitHub':'Personal Access Token'@github.com/'repositorio de backend que quieres clonar'
17
18 # Copia el archivo de entorno
19 COPY ./CityExplorer_Backend/.env /var/www/html/cityExplorer-backend
20
21 # Instala Composer
22 RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
23
24 # Cambia el directorio de trabajo al repositorio clonado
25 WORKDIR /var/www/html/cityExplorer-backend
26
27 # Actualiza e instala dependencias del proyecto con Composer
28 RUN composer update && composer install
29
30 # Cambia la propiedad y permisos de los directorios necesarios
31 RUN chown -R www-data:www-data storage bootstrap/cache && \
32     chmod -R 775 storage && \
33     chmod -R 775 bootstrap/cache && \
34     chmod -R 777 public
35
36 # Desactiva el sitio por defecto de Apache
37 RUN a2dissite 000-default.conf
38
39 # Copia la configuración personalizada de Apache y la habilita
40 COPY web_laravel.conf /etc/apache2/sites-available
41 RUN a2ensite web_laravel.conf && a2enmod rewrite
42
43 # Expone el puerto 80 para HTTP
44 EXPOSE 80
45
46 # Comando para iniciar Apache en primer plano
47 CMD ["apache2-foreground"]
48
```

3) Crear DockerFile para el frontend.

Ten en cuenta a la hora de clonar el repositorio poner tu nombre de usuario, tu Personal Access Token que podrás encontrar o crear en tu cuenta de GitHub y el repositorio de frontend de donde lo quieres clonar.

```
DockerFile-frontend > ...
1 FROM node:20.11.1
2
3 WORKDIR /app
4
5 # Instala git y nginx en una sola capa
6 RUN apt-get update && \
7     apt-get install -y git nginx && \
8     rm -rf /var/lib/apt/lists/*
9
10 # Clona el repositorio
11 RUN git clone https://'nombre usuario de GitHub':'Personal Access Token'@github.com/'repositorio de front que quieres clonar'
12
13 WORKDIR /app/cityExplorer-frontend
14
15 # Instala dependencias y construye el proyecto
16 RUN npm install && \
17     npm install @rollup/rollup-linux-x64-gnu && \
18     npm run build
19
20 # Configura nginx
21 COPY nginx.conf /etc/nginx/sites-available/nginx.conf
22 RUN ln -s /etc/nginx/sites-available/nginx.conf /etc/nginx/sites-enabled
23
24 # Copia archivos construidos al directorio web de nginx
25 RUN cp -r /app/cityExplorer-frontend/dist/* /var/www/html/
26
27 # Exponer el puerto 80
28 EXPOSE 80
29
30 # Comando para iniciar nginx
31 CMD ["nginx", "-g", "daemon off;"]
32
```

4) Crear DockerFile para la base de datos

Ten en cuenta cambiar el nombre 'baseDeDatos.sql' por tu base de datos. Este archivo tiene que estar correctamente ubicado tal y como se ve en el esquema inicial.

```
DockerFile-db > FROM
1 FROM mysql:latest
2 EXPOSE 80
3
```

5) Crear docker-compose.yml

```

docker-compose.yml
1  version: '3.3'
2
3  services:
4    db:
5      container_name: CityExplorer_mysql_db
6      build:
7        dockerfile: DockerFile-db
8        context: .
9      ports:
10     - "3306:80"
11     environment:
12       MYSQL_DATABASE: city_explorer
13       MYSQL_USER: maximo
14       MYSQL_PASSWORD: Maximo.2001
15       MYSQL_ROOT_PASSWORD: root
16     restart: always
17
18     laravel:
19       container_name: CityExplorer_Backend
20       build:
21         dockerfile: DockerFile-backend
22         context: .
23       volumes:
24         - laravel_data:/var/www/html
25       ports:
26         - "8000:80"
27       restart: always
28       depends_on:
29         - db
30
31     react:
32       container_name: CityExplorer_Frontend
33       build:
34         dockerfile: DockerFile-frontend
35         context: .
36       volumes:
37         - react_data:/var/www/html
38       ports:
39         - "80:80"
40       restart: always
41       depends_on:
42         - laravel
43
44   volumes:
45     laravel_data:
46     react_data:
47

```

6) Crear nginx.conf

Este archivo define las directrices básicas para cómo Nginx debe manejar las solicitudes HTTP para un dominio específico (cityexplorer.com). Nginx es un servidor web popular y también se puede usar como proxy inverso, balanceador de carga, y más.

```
nginx.conf
1
2 server {
3     listen 80;
4     server_name maximonp24.iesmontenaranco.com www.maximonp24.iesmontenaranco.com;
5     root /var/www/html;
6     location / {
7         try_files $uri /index.html;
8     }
9
10 }
```

7) Crear web_laravel.conf

Este archivo de configuración (web_laravel.conf) se utiliza para configurar un host virtual en Apache para el sitio web cityexplorer.com, especificando cómo se deben manejar las solicitudes y dónde se encuentran los archivos del sitio. Es particularmente relevante para aplicaciones Laravel, ya que señala el directorio 'public' como el punto de entrada principal.

```
web_laravel.conf
1 <VirtualHost *:80>
2     ServerAdmin maximo@novoa.com
3     ServerName maximonp24.iesmontenaranco.com
4     DocumentRoot /var/www/html/cityExplorer-backend/public/
5
6     <Directory /var/www/html/cityExplorer-backend/public/>
7         Options Indexes FollowSymLinks
8         AllowOverride All
9         Require all granted
10    </Directory>
11
12    ErrorLog /var/log/apache2/error.log
13    CustomLog /var/log/apache2/access.log combined
14 </VirtualHost>
```

8) En nuestra carpeta de front debemos tener un archivo .env donde guardaremos la IP donde hará referencia nuestras peticiones en una

variable. Es recomendable tenerlo en este archivo para así no cambiarlo en todas las peticiones que hagamos

```
CityExplorer_Frontend > .env
You, 1 second ago | 2 authors (You and one other)
1 VITE_APP_PETICION_IP = maximonp24.iesmontenaranco.com:8000
2
3
4
5
6
7
```

- 9) En la carpeta de back tenemos por defecto este archivo .env ya creado. Debe de estar configurado de esta manera:

```
CityExplorer_Backend > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:BiOD9kuBnTvk4W5uit/9G2gj6ed2GyjM48bHEYBkMRg=
4 APP_DEBUG=true
5 APP_URL=https://maximonp24.iesmontenaranco.com:8000
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 #conexion localhost
12 # DB_CONNECTION=mysql
13 # DB_HOST=127.0.0.1
14 # DB_PORT=3306
15 # DB_DATABASE=city_explorer
16 # DB_USERNAME=maximo
17 # DB_PASSWORD=Maximo.2001
18
19
20 #conexion AWS
21 #DB_CONNECTION=mysql
22 #DB_HOST=cityexplorer.cl0acek06xsc.us-east-1.rds.amazonaws.com
23 #DB_PORT=3306
24 #DB_DATABASE=city_explorer
25 #DB_USERNAME=maximo
26 #DB_PASSWORD=Maximo.2001
27
28
29 #conexion dockerfile-db
30 DB_CONNECTION=mysql
31 DB_HOST=db
32 DB_PORT=3306
33 DB_DATABASE=city_explorer
34 DB_USERNAME=maximo
35 DB_PASSWORD=Maximo.2001
```

- 10) Comprimimos la carpeta del proyecto en un zip.

- 11) Descomprimos la carpeta en nuestro programa cliente SSH (PuTTY, Termius), en el servidor donde queremos desplegarlo. Existen muchas

opciones: un servidor propio, un servidor conjunto, AWS, etc. En este caso, un servidor personalizado del IES Monte Naranco para los alumnos de este ciclo.

Utilizamos el comando `unzip *nombre de la carpeta*`

12) Entramos en la carpeta descomprimida con `cd *nombre de la carpeta*`

13) Iniciamos sesión en Docker con el comando `docker login`

14) Escribimos el comando `docker-compose up -d`.

Esto lo que hace es iniciar los servicios definidos en el archivo `docker-compose.yml` en segundo plano (modo detached)

15) Entramos en nuestro contenedor del back con `docker exec -it CityExplorer_Backend /bin/bash`

16) Al ser nuestra primera vez, migramos nuestros datos a la base de datos: `php artisan migrate:fresh --seed`

17) Una vez conseguimos efectuar los pasos anteriores, pegamos en la URL del buscador de nuestro navegador la IP que nos hayan proporcionado ya sea nuestro tutor, nosotros mismos o AWS (recomendable usar IP elástica en este caso).

18) Le damos a buscar y ¡LISTO!