

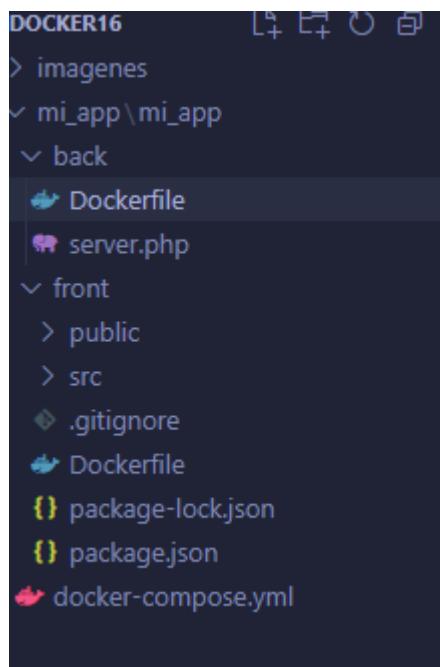
Actividad Docker. Conectividad frontend-backend

1. Se entregan los ficheros de una aplicación frontend con react que muestra un formulario donde el usuario puede escribir su nombre y tras pulsar el botón de envío se realiza una petición AJAX que envía mediante POST los datos del formulario hacia un fichero server.php (se entrega también) que los recoge y devuelve un saludo.

Se debe tener en cuenta que:

- a) Se debe crear un fichero docker-compose.yml que permita el despliegue automático de los servicios frontend y backend
- b) El servicio del backend debe exponer el puerto 80 del contenedor (puerto por defecto del contenedor php) al puerto 8000 de la máquina Windows
- c) El servicio frontend debe exponer el puerto 80 del contenedor (puerto por defecto del contenedor nginx) al puerto 80 de la máquina Windows
- d) El servicio frontend debe construir una imagen multistage (2 etapas) con un fichero Dockerfile que permita:
 - Etapa 1
 - Usar la imagen de node:18-alpine
 - Establecer el directorio de trabajo en /app
 - Copiar los ficheros package*.json de la aplicación
 - Instalar las dependencias
 - Copiar los ficheros de la aplicación
 - Ejecutar npm run build (crea la subcarpeta build con la construcción de la aplicación)
 - Etapa 2
 - Usar la imagen de nginx
 - Copiar desde la etapa anterior los ficheros de la carpeta /app/build (de la etapa 1) a la carpeta /usr/share/nginx/html (de la etapa 2)
- e) El servicio de backend debe construir una imagen con Dockerfile que permita:
 - Usar la imagen de php:7.4-apache
 - Copiar el fichero server.php al directorio /var/www/html
- f) Una vez completado el despliegue se debe verificar el correcto funcionamiento accediendo a <http://localhost> y probando el envío de datos mediante el formulario del frontend

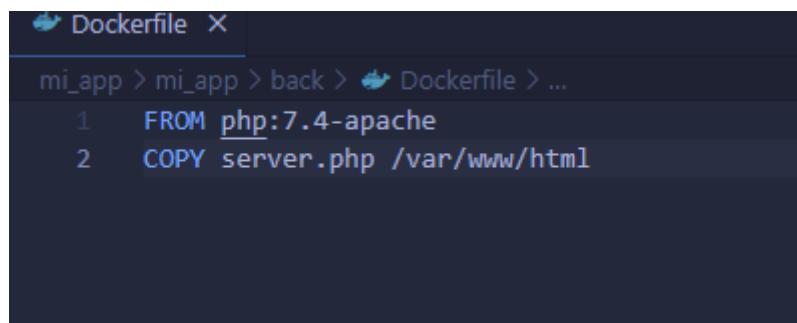
1.- Estructura:



2.- Dockerfile del frontend:

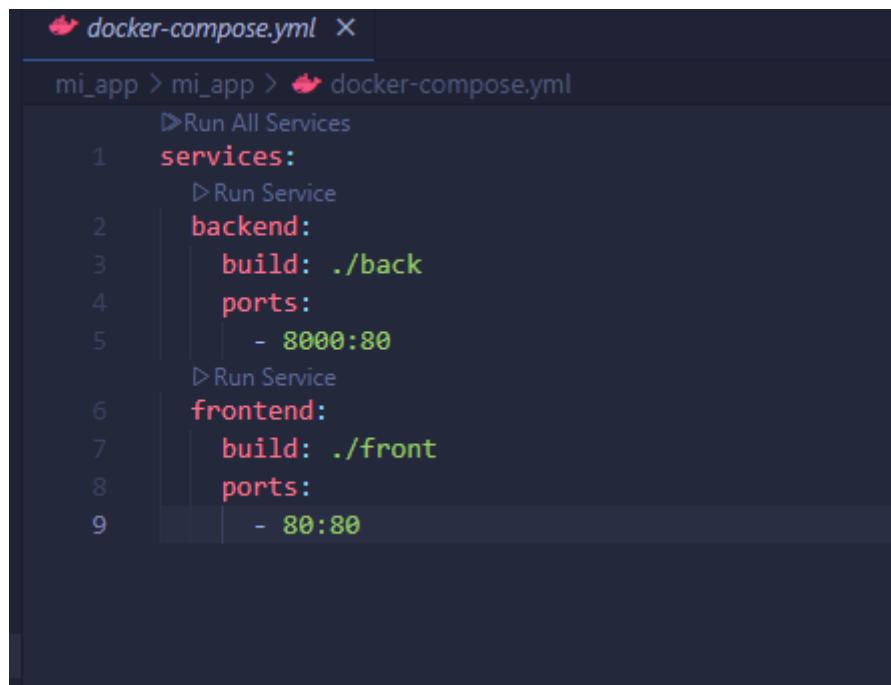
```
FROM node:18-alpine as react
WORKDIR /app
COPY package*.json .
RUN npm install
COPY . .
RUN npm run build
FROM nginx
COPY --from=react /app/build /usr/share/nginx/html
```

3.- Dockerfile del backend:



```
mi_app > mi_app > back > Dockerfile > ...
1 FROM php:7.4-apache
2 COPY server.php /var/www/html
```

4.- docker-compose.yml:



```
mi_app > mi_app > docker-compose.yml
    ▷Run All Services
1 services:
2     ▷Run Service
3         backend:
4             build: ./back
5             ports:
6                 - 8000:80
7             ▷Run Service
8                 frontend:
9                     build: ./front
10                    ports:
11                        - 80:80
```

5.- Resultado:

