

Actividades Docker. Node con React

1. Se pide usar el contenedor de **node:18-alpine** y sobre él crear una imagen a partir de un Dockerfile que permita dockerizar la aplicación React que se entrega en el ZIP adjunto

Previamente se debe crear una carpeta para aplicación y copiar dentro de ella los ficheros de la aplicación

El fichero Dockerfile debe realizar las siguientes acciones:

- Descargar la imagen de node que se pide
- Establecer el directorio de trabajo en **/app** mediante la instrucción **WORKDIR**
- Copiar **packages.json** al directorio anterior
- Instalar las dependencias con **npm install**
- Instalar con **npm i -g serve** el servidor web
- Copiar los ficheros de la aplicación en **/app**
- Construir la aplicación con **npm run build**
- Exponer el puerto 3000
- Lanzar la aplicación con el comando ejecutando **serve -s dist** mediante **CMD**

Se muestra captura de pantalla de la configuración del fichero vite.config.js para que trabaje con el puerto 3000 tanto en modo desarrollo como en producción

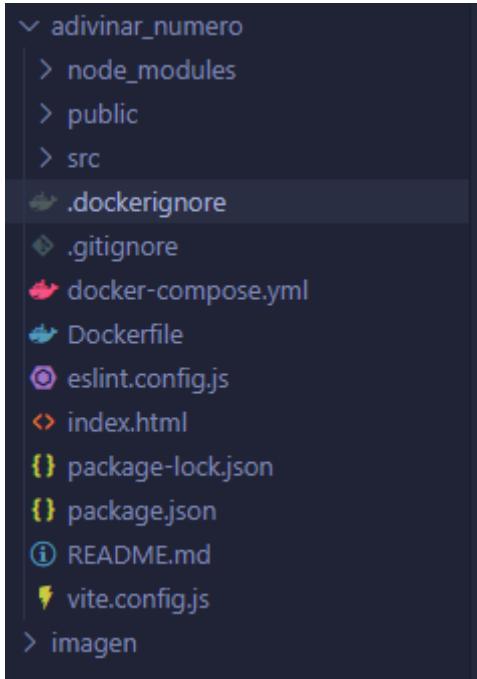
```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// https://vite.dev/config/
export default defineConfig({
  plugins: [react()],
  base: "/",
  preview: {
    port: 3000,
    strictPort: true,
  },
  server: {
    port: 3000,
    strictPort: true,
    host: true,
    origin: "http://0.0.0.0:3000",
  },
})
```

La creación del contenedor se realizará mediante docker-compose.yml que permitirá vincular o mapear el puerto 3000 del contenedor con el puerto 3000 de la maquina Windows (este último puerto puede modificarse lógicamente). Este fichero creará también el servicio **react** y el contenedor de nombre **app_react**

Se debe también mostrar una captura de pantalla del acceso a <http://localhost:3000>

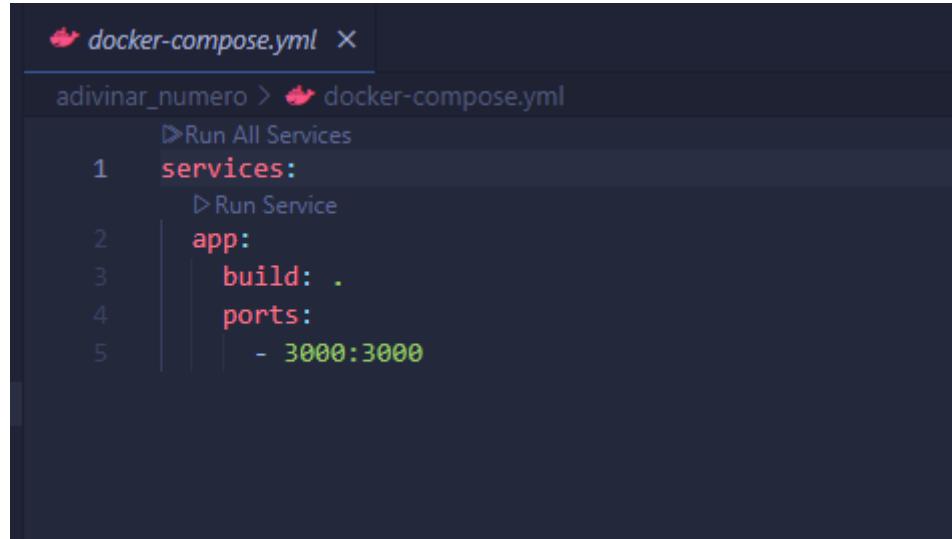
1.- Estructura:



2.- Dockerfile:

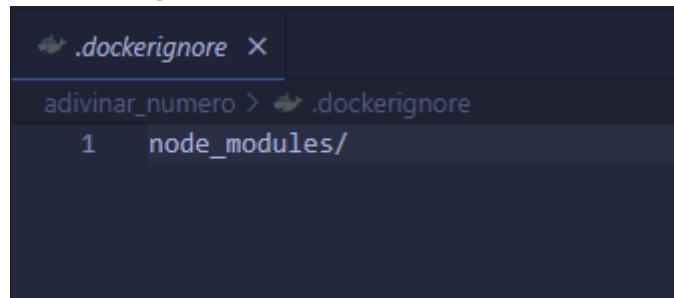
```
Dockerfile
adivinar_numero > Dockerfile > ...
1  FROM node:18-alpine
2  WORKDIR /usr/src/app
3  COPY package.json .
4  RUN npm install
5  RUN npm i -g serve
6  COPY . .
7  RUN npm run build
8  EXPOSE 3000
9  CMD [ "serve", "-s", "dist" ]
```

3.- Docker-compose:



```
docker-compose.yml
adivinar_numero > docker-compose.yml
  1   services:
  2     app:
  3       build: .
  4       ports:
  5         - 3000:3000
```

4.- .dockerignore para no copiar el node_modules:



```
.dockerignore
adivinar_numero > .dockerignore
  1   node_modules/
```

5.- Resultado:

