

## **Reto Docker**

Natalia Flórez Guzmán

Gestión de la Configuración

José Ignacio López Vélez

Universidad de Medellín

2025

### Paso 1 crear una carpeta que se llame node-docker-app.

Crear el directorio para el proyecto con mkdir y luego entramos a esa carpeta, correr este comando para generar un package.json

```
Microsoft Windows [Versión 10.0.26100.6584]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\NataliaF>cd Downloads

C:\Users\NataliaF\Downloads>mkdir node-docker-app

C:\Users\NataliaF\Downloads>cd node-docker-app

C:\Users\NataliaF\Downloads\node-docker-app>npm init -y
Wrote to C:\Users\NataliaF\Downloads\node-docker-app\package.json:

{
  "name": "docker-app",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

### Paso 2

Creamos un archivo llamado app.js y se agrega el código

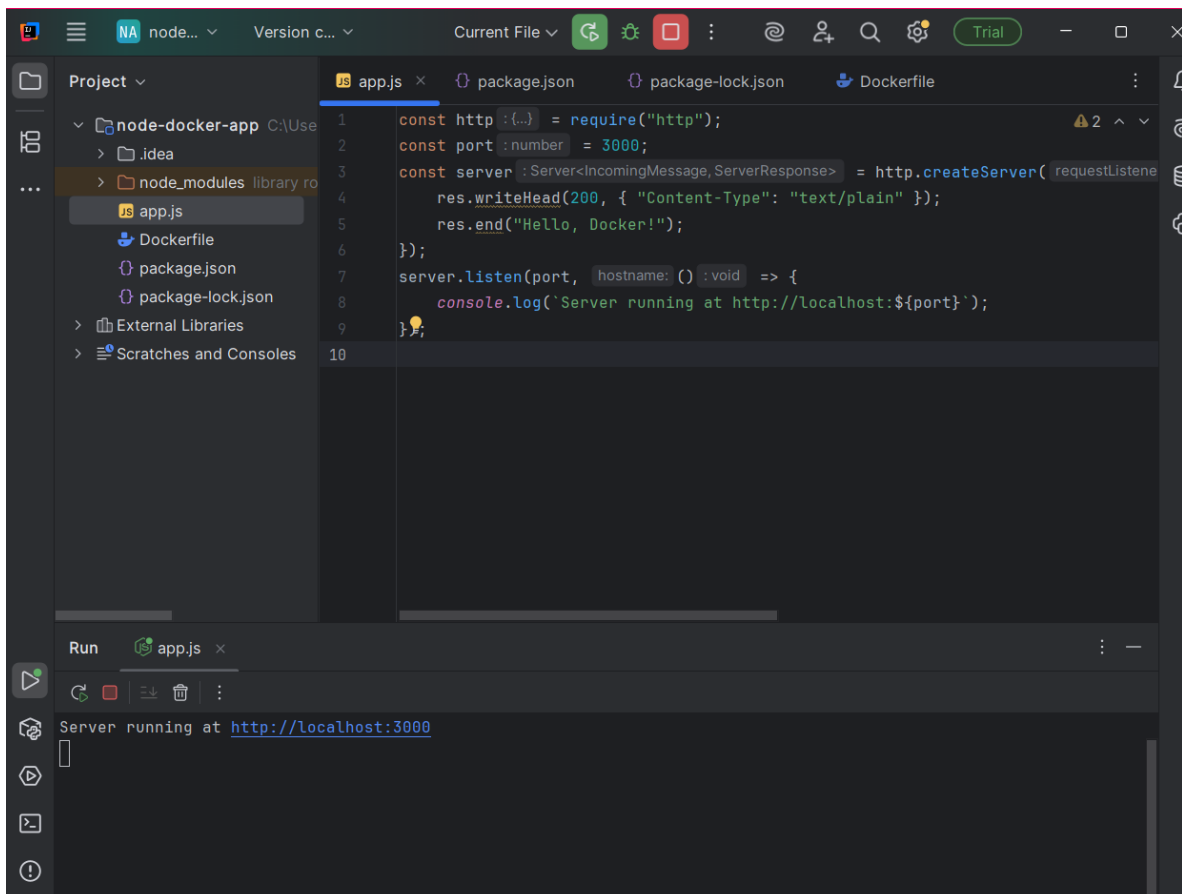
```
const http = require("http");
```

```
const port = 3000;
```

```
const server = http.createServer((req, res) => {
  res.writeHead(200, { "Content-Type": "text/plain" });
  res.end("Hello, Docker!");
});
```

```
server.listen(port, () => {  
  console.log(`Server running at http://localhost:${port}`);  
});
```

```
C:\Users\NataliaF\Downloads\node-docker-app>notepad app.js  
C:\Users\NataliaF\Downloads\node-docker-app>|
```

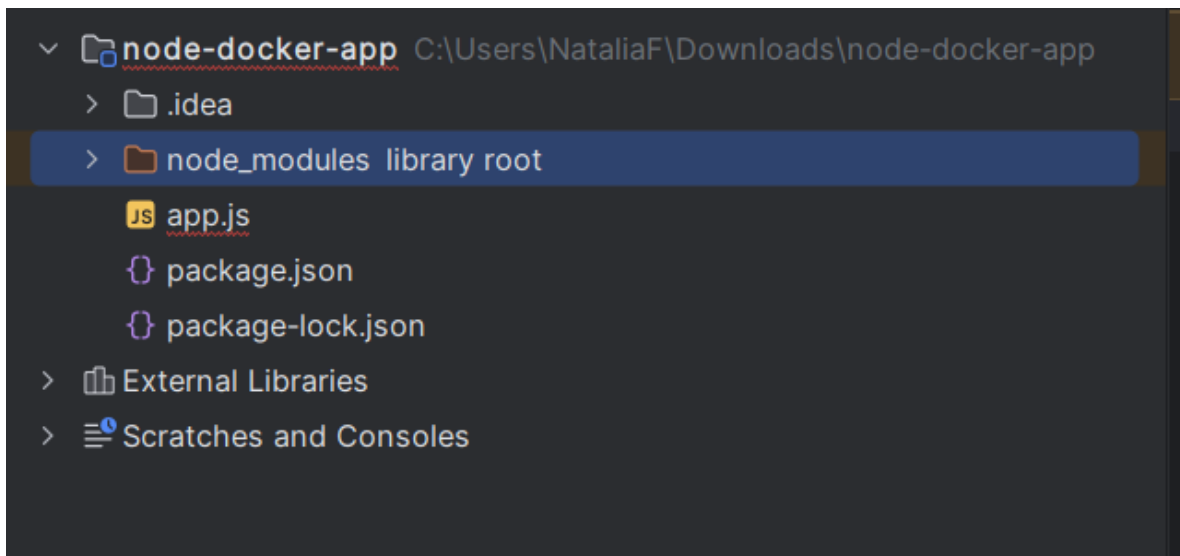


### Paso 3

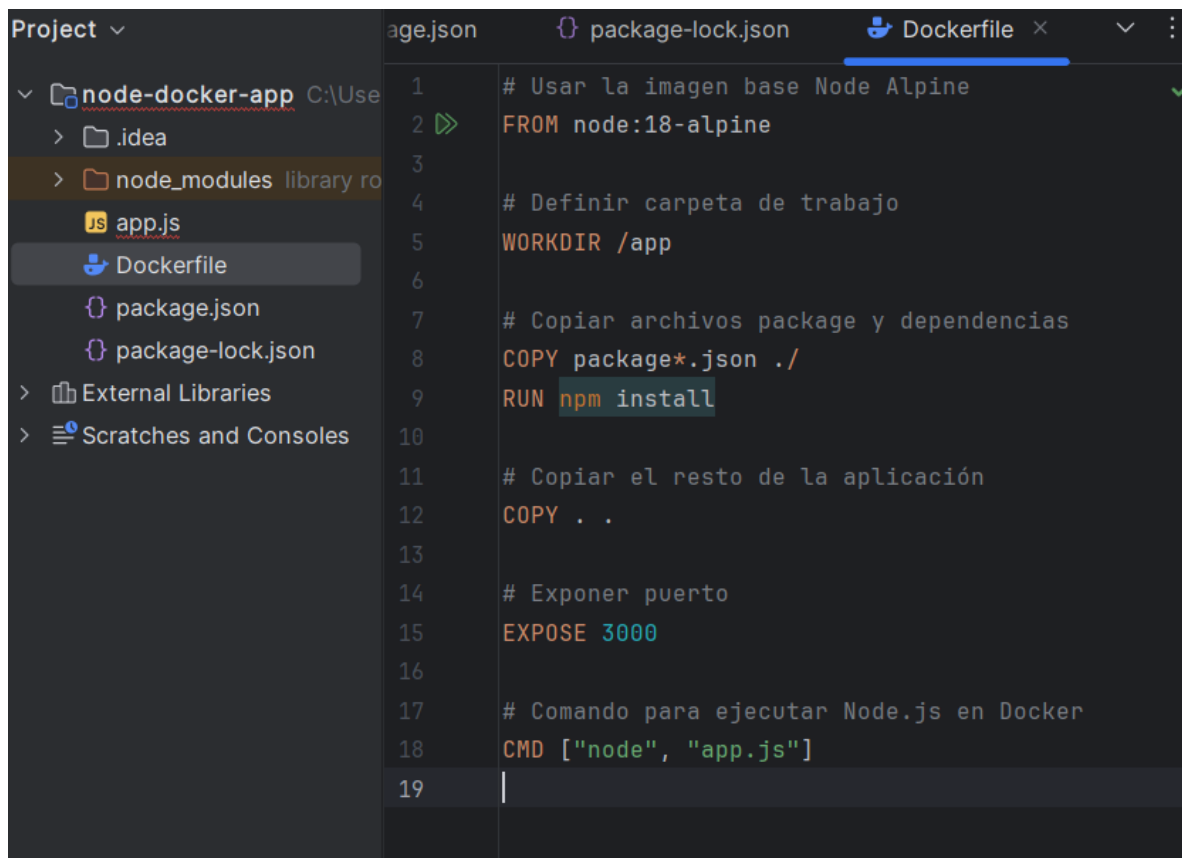
Instalamos las dependencias necesarias con `npm install express`

```
C:\Users\NataliaF\Downloads\node-docker-app>npm install express  
  
added 68 packages, and audited 69 packages in 4s  
  
16 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
  
C:\Users\NataliaF\Downloads\node-docker-app>|
```

Al instalar las dependencias se creó una carpeta node\_modules



Creamos un archivo Dockerfile en la raíz del proyecto y pegar lo dado en la guía



The screenshot shows an IDE with a project named 'node-docker-app'. The file explorer on the left lists files: app.js, Dockerfile, package.json, and package-lock.json. The Dockerfile is open in the editor, showing the following content:

```
1 # Usar la imagen base Node Alpine
2 FROM node:18-alpine
3
4 # Definir carpeta de trabajo
5 WORKDIR /app
6
7 # Copiar archivos package y dependencias
8 COPY package*.json ./
9 RUN npm install
10
11 # Copiar el resto de la aplicación
12 COPY . .
13
14 # Exponer puerto
15 EXPOSE 3000
16
17 # Comando para ejecutar Node.js en Docker
18 CMD ["node", "app.js"]
19
```

**FROM** = imagen base.

**WORKDIR** = directorio de trabajo dentro del contenedor.

**COPY** = copia archivos del host al contenedor.

**RUN** = ejecuta comandos dentro del contenedor.

**CMD** = indica qué ejecutar al iniciar el contenedor.

#### Paso 4 Construimos la imagen ejecutando en la raíz del proyecto

sudo docker build -t node-docker-app .

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas caract
erísticas y mejoras. https://aka.ms/PSWindows

PS C:\Users\NataliaF\Downloads\node-docker-app> sudo docker build -t nod
e-docker-app .
Sudo está deshabilitado en este equipo. Para habilitarlo, vaya a Página...
Configuración del desarrollador en la aplicación Configuración
PS C:\Users\NataliaF\Downloads\node-docker-app> sudo docker build -t nod
e-docker-app .
PS C:\Users\NataliaF\Downloads\node-docker-app> |
```

### Paso : Ejecutar el contenedor y abrir el navegador

