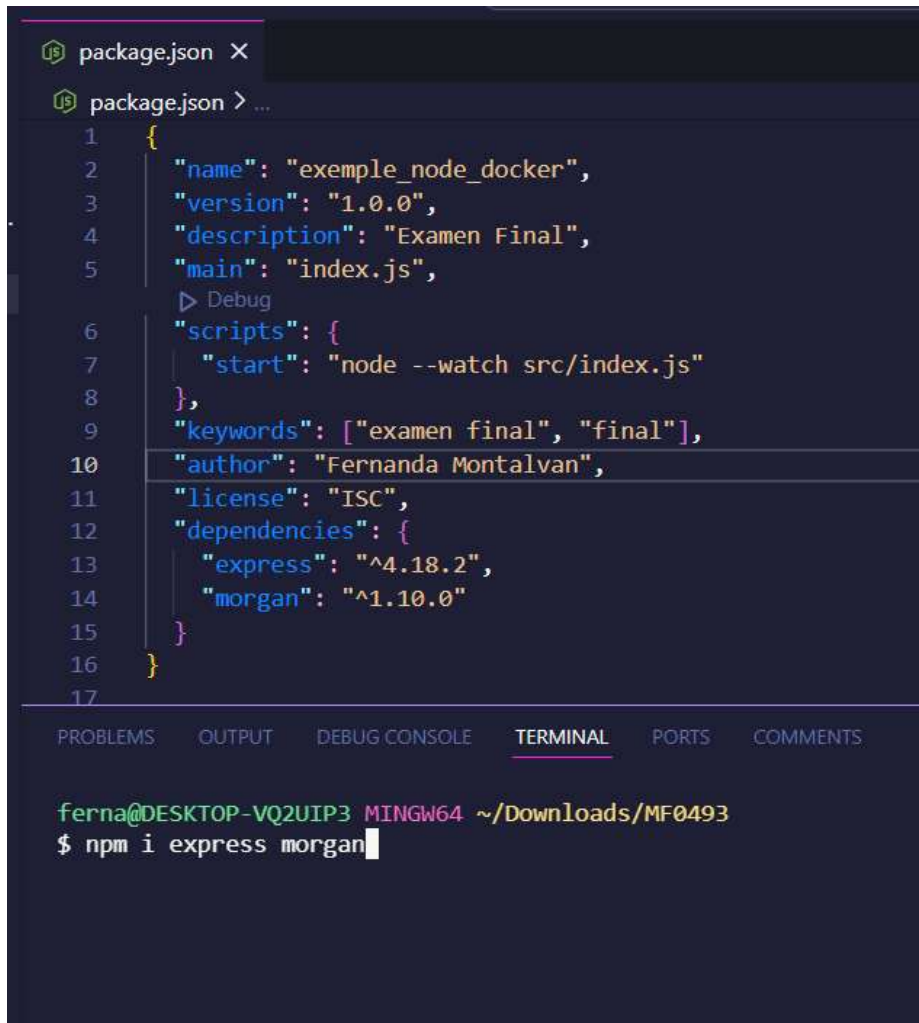


Rellenamos nuestro package.json e instalamos los módulos de Express y Morgan (he instalado la versión de morgan -D



The image shows a VS Code editor window with a file named `package.json` open. The file contains the following JSON:

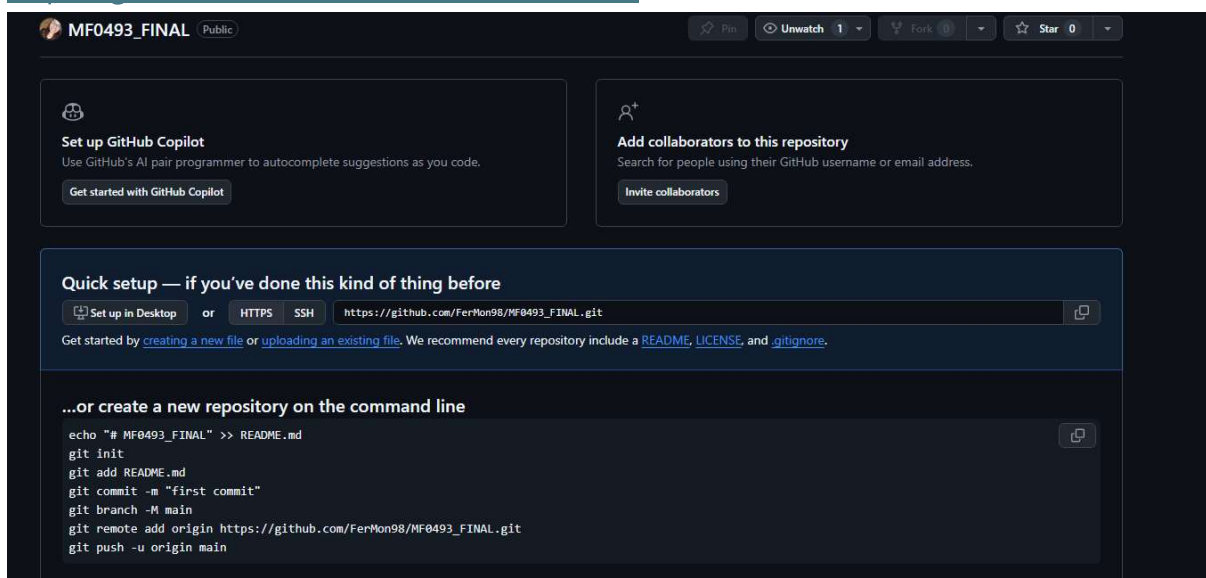
```
{
  "name": "exemple_node_docker",
  "version": "1.0.0",
  "description": "Examen Final",
  "main": "index.js",
  "scripts": {
    "start": "node --watch src/index.js"
  },
  "keywords": ["examen final", "final"],
  "author": "Fernanda Montalvan",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2",
    "morgan": "^1.10.0"
  }
}
```

Below the editor, the `TERMINAL` tab is active, showing a command prompt with the following text:

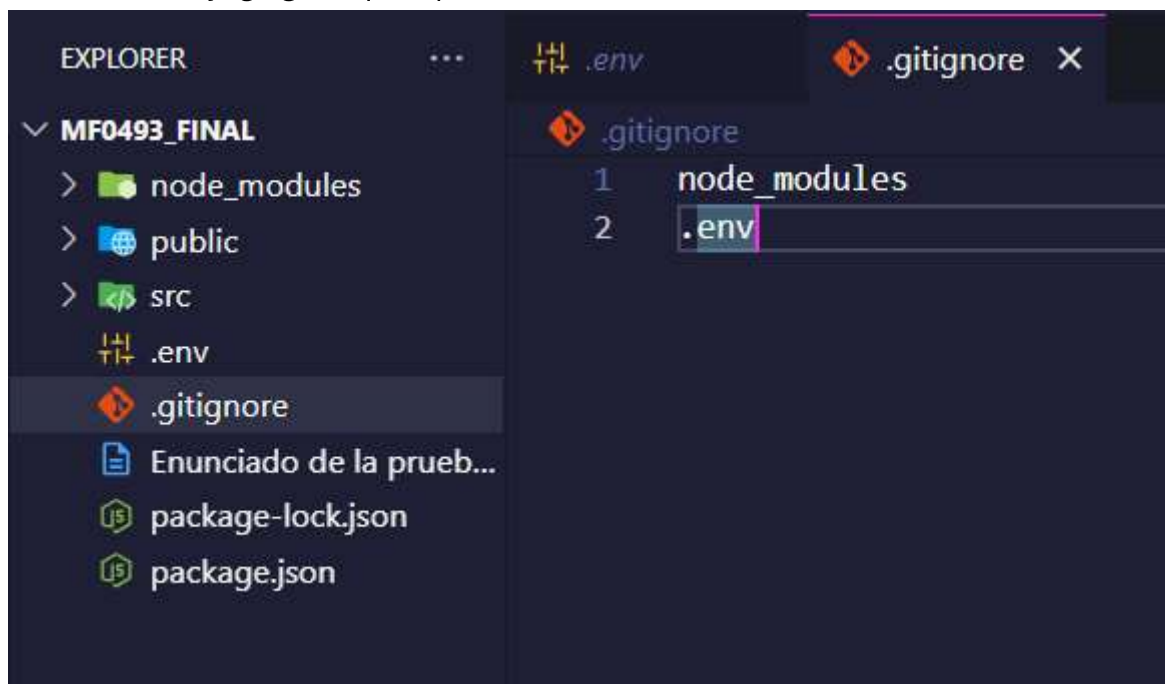
```
ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493
$ npm i express morgan
```

Creamos el repositorio de Git al que estaremos subiendo los archivos

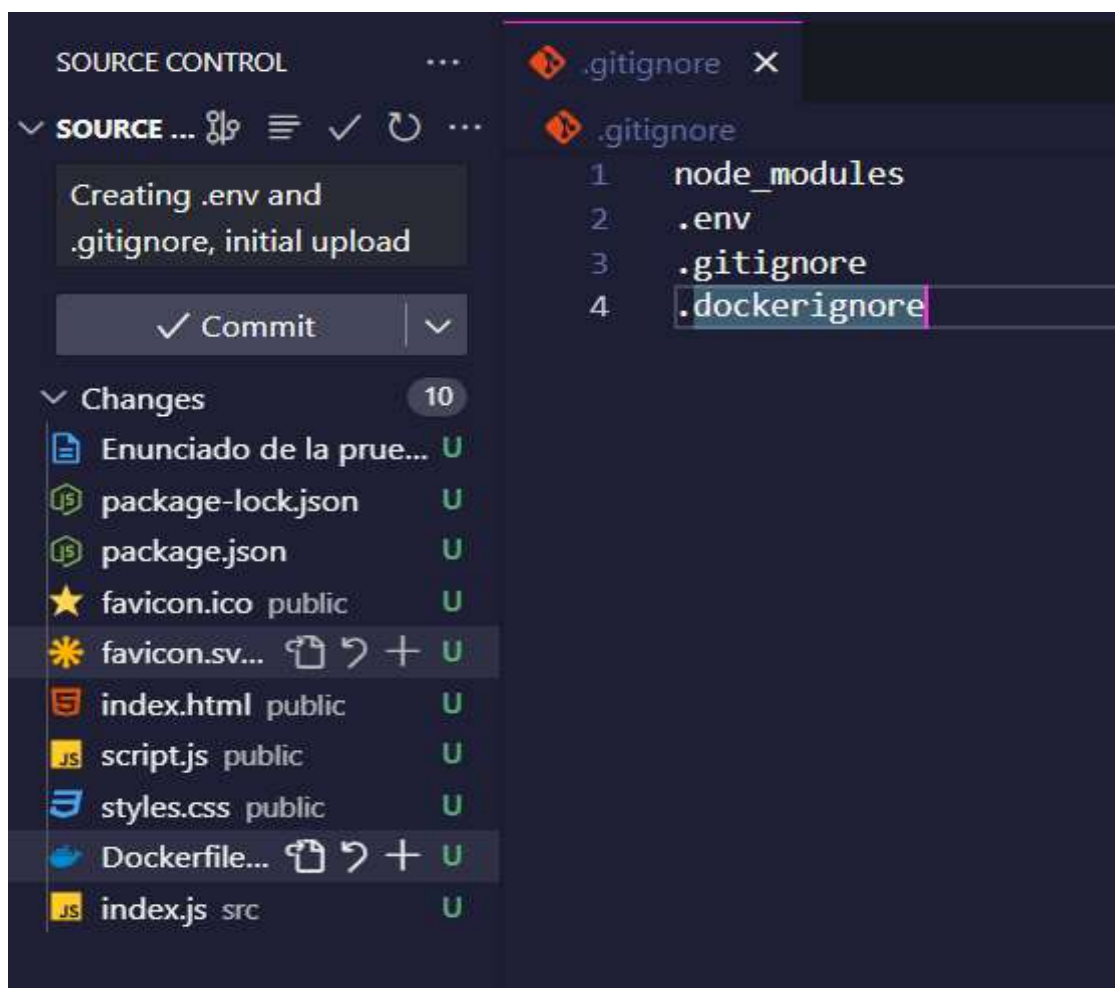
https://github.com/FerMon98/MF0493_FINAL



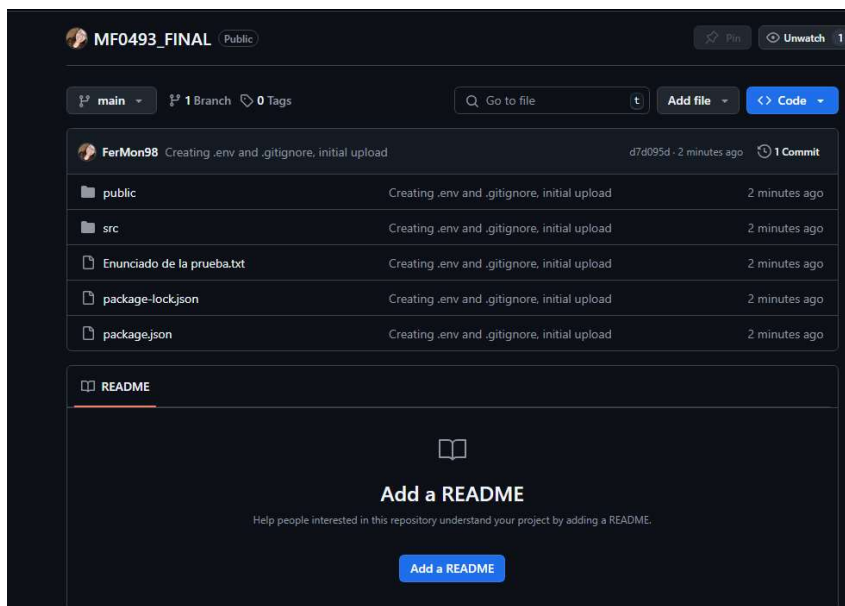
Creamos .env y .gitignore para poder Subir nuestros archivos.



Primer commit para poder subir los archivos incluyendo .dockerignore dentro de los archivos



Comprobamos que los archivos se han subido correctamente



Nos aseguramos de declarar PORT en el archivo .env (6000) y nos instalamos dotenv

```
src > JS index.js > ...
1 // AQUÍ ARRANCAMOS EL SERVIDOR
2 const express = require('express')
3 const app = express()
4
5 const {join} = require('path')
6
7 //Accedemos a dotenv para obtener variables de entorno
8 require('dotenv').config({ path: '../.env' });
9
10
11 // Per poder veure per consola les peticions http que van arribant
12 const morgan = require('morgan') 13.8k (gzipped: 5.1k)
13 app.use(morgan('dev'))
14
15 const port = process.env.PORT || 3000
16
17 app.use(express.static('public'))
18 // const router = require('./routes/index.js')
19 // app.use(router)
20 // sintaxi alternativa
21 // app.use(require('./routes/index.js'))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

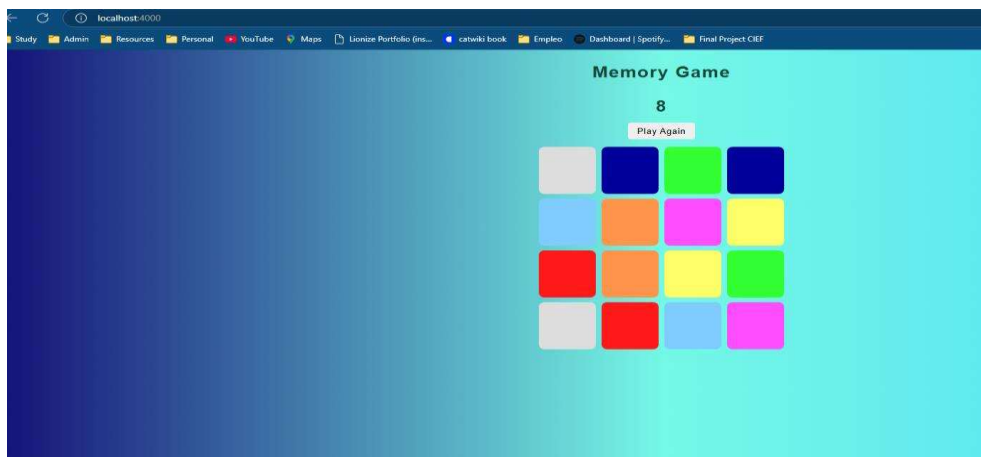
ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$ npm install dotenv
```

Comprobamos que corre en el puerto adecuado y que la app se despliega correctamente

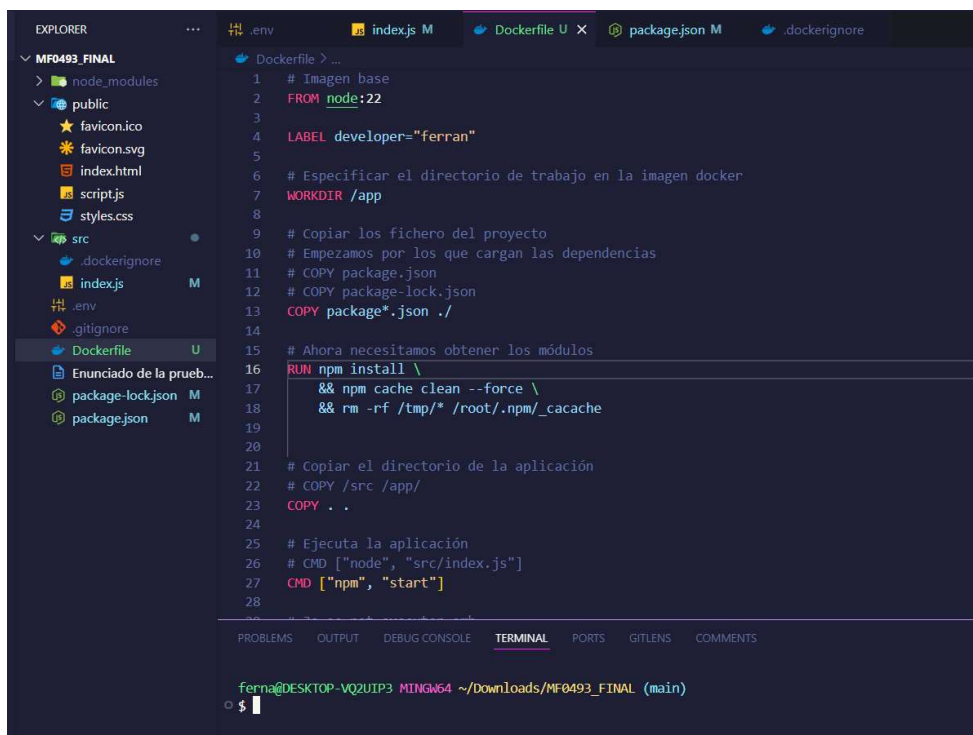


```
TERMINAL
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

TERM_PROGRAM: 'vscode',
TERM_PROGRAM_VERSION: '1.95.3',
TMP: 'C:\\Users\\ferna\\AppData\\Local\\Temp',
Restarting 'src/index.js'
Servidor web arrancat a http://localhost:4000
█
```



Preparamos el entorno de Docker, el archivo de dockerfile tiene que estar en la ruta del directorio, añadimos código para limpiar cache (además del resto de comandos para subir la app

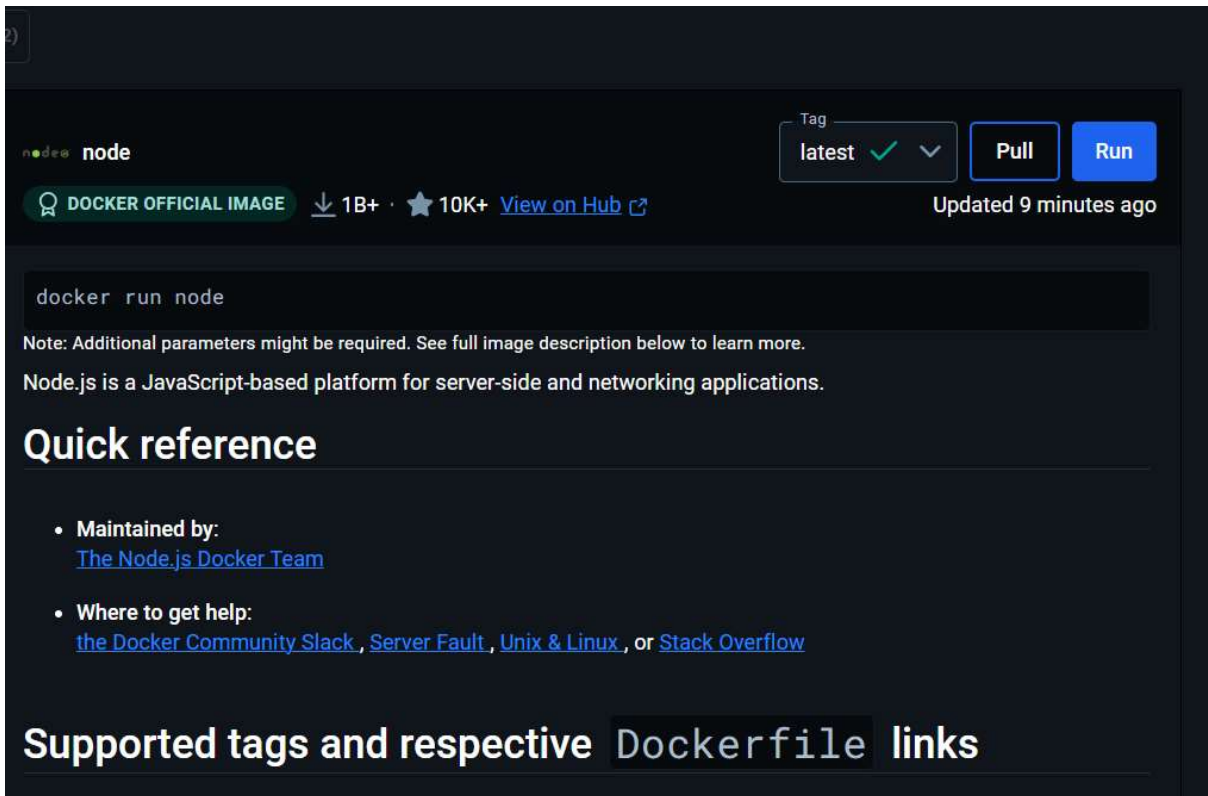


```
EXPLORER
MF0493_FINAL
├── node_modules
├── public
│   ├── favicon.ico
│   ├── favicon.svg
│   ├── index.html
│   ├── script.js
│   └── styles.css
├── src
│   ├── .dockerignore
│   ├── index.js
│   ├── .env
│   ├── .gitignore
│   ├── Dockerfile
│   ├── Enunciado de la prueba...
│   ├── package-lock.json
│   └── package.json
└── ...

Dockerfile
1 # Imagen base
2 FROM node:22
3
4 LABEL developer="ferran"
5
6 # Especificar el directorio de trabajo en la imagen docker
7 WORKDIR /app
8
9 # Copiar los fichero del proyecto
10 # Empezamos por los que cargan las dependencias
11 # COPY package.json
12 # COPY package-lock.json
13 COPY package*.json ./
14
15 # Ahora necesitamos obtener los módulos
16 RUN npm install \
17     && npm cache clean --force \
18     && rm -rf /tmp/* /root/.npm/_cacache
19
20
21 # Copiar el directorio de la aplicación
22 # COPY /src /app/
23 COPY . .
24
25 # Ejecuta la aplicación
26 # CMD ["node", "src/index.js"]
27 CMD ["npm", "start"]
28

TERMINAL
ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$
```

Nos aseguramos que tenemos descargado node



The screenshot shows the Docker Hub interface for the 'node' image. At the top, it says 'node' with a 'Tag' dropdown set to 'latest'. There are 'Pull' and 'Run' buttons. Below this, it indicates 'DOCKER OFFICIAL IMAGE' with download and star counts (1B+ and 10K+), a 'View on Hub' link, and 'Updated 9 minutes ago'. A code block shows the command 'docker run node'. A note states: 'Note: Additional parameters might be required. See full image description below to learn more. Node.js is a JavaScript-based platform for server-side and networking applications.' Below this is a 'Quick reference' section with two bullet points: 'Maintained by: The Node.js Docker Team' and 'Where to get help: the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow'. At the bottom, a section titled 'Supported tags and respective Dockerfile links' is partially visible.

Nos aseguramos que estamos loggeados en la terminal y corremos nuestro comando build para crear la imagen

```
ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$ docker login --username ferzk98 --password [REDACTED]
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded

ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$ docker build -t mf0493 .
[+] Building 26.1s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 795B
=> [internal] load metadata for docker.io/library/node:20
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:20@sha256:f99a6bd6fb137cfc683e1ff60e7d67aa8d5fcf45a6a64c01c37b4f433a6238b6
=> => resolve docker.io/library/node:20@sha256:f99a6bd6fb137cfc683e1ff60e7d67aa8d5fcf45a6a64c01c37b4f433a6238b6
=> => sha256:3232912c81c7f69d27f6ae8193e237b4d36fe328663ee85f30892a6184b0b737 448B / 448B
=> => sha256:999cfbe722a60f1e73b8da038c54addbb3b17d6c85ad326a7d691cd74b40b375 1.25MB / 1.25MB
=> => sha256:85b5e5802bc805d7fb83e52bce19ab35b371f4ae0768a110580f9cee82992278 48.23MB / 48.23MB
=> => sha256:59e9bd962a2cc4982a37a1d06331fd3b9e68d5917af60ceef39fb4b81fdc1a44 3.33kB / 3.33kB
=> => extracting sha256:59e9bd962a2cc4982a37a1d06331fd3b9e68d5917af60ceef39fb4b81fdc1a44
=> => extracting sha256:85b5e5802bc805d7fb83e52bce19ab35b371f4ae0768a110580f9cee82992278
=> => extracting sha256:999cfbe722a60f1e73b8da038c54addbb3b17d6c85ad326a7d691cd74b40b375
=> => extracting sha256:3232912c81c7f69d27f6ae8193e237b4d36fe328663ee85f30892a6184b0b737
=> [internal] load build context
=> => transferring context: 2.58MB
=> [2/5] WORKDIR /app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install && npm cache clean --force && rm -rf /tmp/* /root/.npm/_cacache
=> [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:f2b2234b8aa51152d73ad4725ca0546300994c5aafcaa6c32980fed51f5318bf
=> => exporting config sha256:22788eedc8766a4d68188637387afe72c36fdfeb1412dfa3089f70cfff347d26c
=> => exporting attestation manifest sha256:4cd05602e710642efdb50732a95d4ab7bfcdb7517c61d2e06659d084d088bcb9
=> => exporting manifest list sha256:23533516374a09f667bd2a4fc688e87119d91b71b4ab41e5f3aa6228e94e65a6
=> => naming to docker.io/library/mf0493:latest
=> => unpacking to docker.io/library/mf0493:latest
```

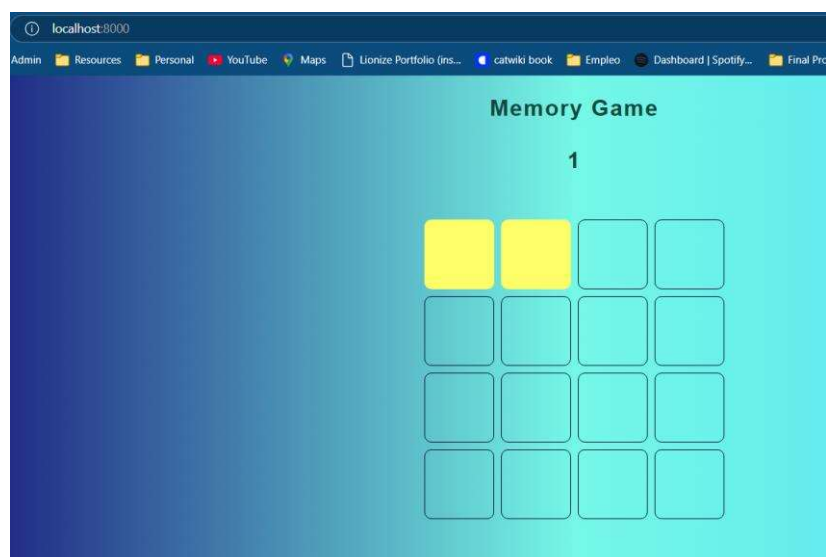
Creamos el contenedor y verificamos que el realmente funciona

```
erna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
docker build --no-cache -t ferzk98/mf0493 .
[+] Building 158.2s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 834B
=> [internal] load metadata for docker.io/library/node:23.1-slim
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 86B
=> [1/5] FROM docker.io/library/node:23.1-slim@sha256:952c222213ccf0179745b518025
=> => resolve docker.io/library/node:23.1-slim@sha256:952c222213ccf0179745b518025
=> [internal] load build context
=> => transferring context: 1.28kB
=> CACHED [2/5] WORKDIR /app
=> [3/5] COPY package*.json ./
=> [4/5] RUN npm install      && npm cache clean --force      && rm -rf /tmp/* /root
=> [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:6093836eed35c8e999111c2411124af4fb6cfc1a4a3b6a4cc
=> => exporting config sha256:70aca29e12e68030f955634c95bfb11d53679635fceb1fa5b28
=> => exporting attestation manifest sha256:4e16bbf709af1270c3d9b6b0f4d68340160e8
=> => exporting manifest list sha256:b522d2359112a9507c8efc7c7decfdd214a65a28f68b
=> => naming to docker.io/ferzk98/mf0493:latest
=> => unpacking to docker.io/ferzk98/mf0493:latest

erna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
docker run -it -p 8000:8000 ferzk98/mf0493

exemple_node_docker@1.0.0 start
node --watch src/index.js

servidor web arrancat a http://localhost:8000
ET / 200 16.636 ms - 514
ET /styles.css 200 1.968 ms - 1220
ET /script.js 200 3.300 ms - 3314
ET /favicon.svg 200 1.164 ms - 2917
```



Una vez comprobado creamos nuestro repositorio en docker hub
https://hub.docker.com/repository/docker/ferzk98/mf0493_fernanda_montalvan/genera , creamos nuestro tag y hacemos push

GeneralTagsBuildsCollaboratorsWebhooksSettings

ferzk98/mf0493_fernanda_montalvan

Created 5 minutes ago

Práctica final

This repository does not have a category

INCOMPLETE

Docker commands

To push a new tag to this repository:

docker push ferzk98/mf0493_fernanda_montalvan:tagname

Public View

Tags

INCOMPLETE

This repository is empty. Push some images to it to see them appear here.

Automated builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

Upgrade

```
ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$ docker tag ferzk98/mf0493:latest ferzk98/mf0493_fernanda_montalvan:v1

ferna@DESKTOP-VQ2UIP3 MINGW64 ~/Downloads/MF0493_FINAL (main)
$ docker push ferzk98/mf0493_fernanda_montalvan:v1
The push refers to repository [docker.io/ferzk98/mf0493_fernanda_montalvan]
199f138f14e4: Pushed
2d429b9e73a6: Mounted from library/node
85c5cddd0333: Pushed
d212f687a707: Pushed
3a45e26a1412: Pushed
1317a471b254: Pushed
593492f73912: Pushed
e7beb2f2f925: Pushed
fb3e878bb7fb: Pushed
9ac6b3d3d592: Pushed
v1: digest: sha256:b522d2359112a9507c8efc7c7decfdd214a65a28f68bf32b36f538c7c2386e71 size: 856
```

ferzk98/mf0493_fernanda_montalvan

Last pushed 1 minute ago

Práctica final

This repository does not have a category

INCOMPLETE

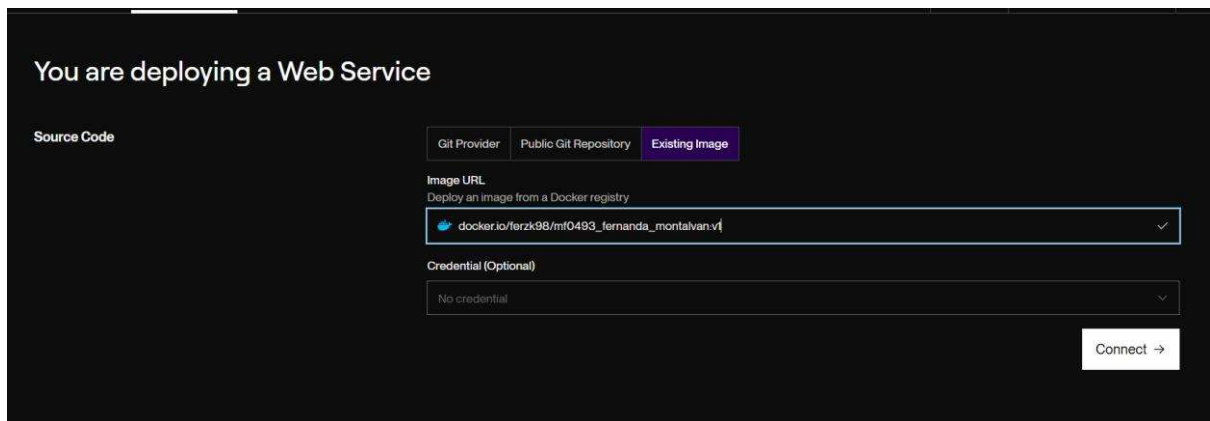
Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	a minute ago	a minute ago

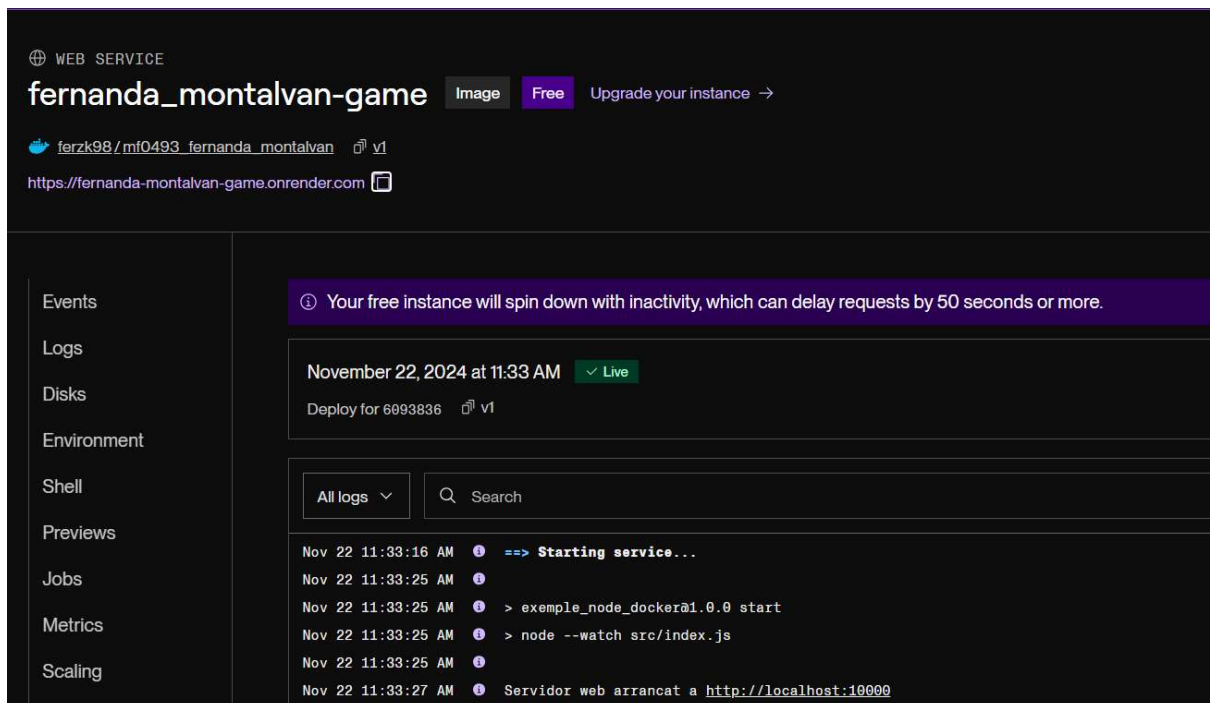
[See all](#)

Ahora que ya hemos actualizado nuestro repositorio estamos listos para subir a Render, recordemos de hacer commit en Git también, en nuestro Dashboard de Render, elegimos Web Service, y nos conectamos a nuestro repositorio, nos aseguramos de elegir el plan que queremos (en nuestro caso el gratis) y de que estamos en la zona horario correcta (Frankfurt, EU Central), también hay que cambiar el nombre si queremos que despliegue un nombre diferente al del repositorio. Dado que es un proyecto sencillo y no tenemos variables de entorno podemos simplemente darle en Deploy Web Service



The screenshot shows the 'You are deploying a Web Service' interface in Render. It has a dark theme. At the top, it says 'You are deploying a Web Service'. Below this, there are three tabs: 'Git Provider', 'Public Git Repository', and 'Existing Image'. The 'Existing Image' tab is selected. Underneath, there's a section for 'Image URL' with the instruction 'Deploy an image from a Docker registry'. A text input field contains 'docker.io/ferzk98/mt0493_fernanda_montalvan.v1'. Below that is a 'Credential (Optional)' section with a dropdown menu set to 'No credential'. A 'Connect' button with a right arrow is at the bottom right.

Una vez desplegado deberíamos de poder ver nuestro proyecto así.



The screenshot shows the Render dashboard for a service named 'fernanda_montalvan-game'. The top bar includes a globe icon, 'WEB SERVICE', the service name, and buttons for 'Image', 'Free', and 'Upgrade your instance'. Below this, it shows the Docker image 'ferzk98/mt0493_fernanda_montalvan' and the version 'v1', along with the URL 'https://fernanda-montalvan-game.onrender.com'. A sidebar on the left lists navigation options: Events, Logs, Disks, Environment, Shell, Previews, Jobs, Metrics, and Scaling. The main area displays a status message: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' Below this, it shows the deployment date 'November 22, 2024 at 11:33 AM' and a 'Live' status. A 'Deploy for 6093836' button is also visible. At the bottom, there's a log viewer with a search bar and a list of logs. The logs show the service starting at 11:33:16 AM, followed by commands to start the Docker container and run the application, and finally, the web server starting at 11:33:27 AM on localhost:10000.

Y accediendo al link que nos da render, deberíamos de poder ver nuestra app desplegada <https://fernanda-montalvan-game.onrender.com>

Memory Game

