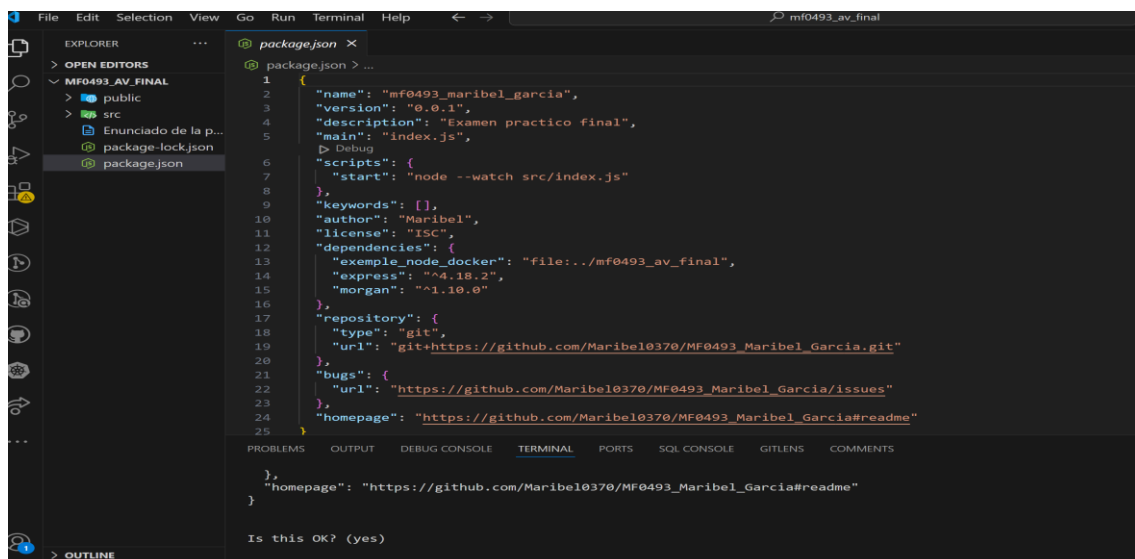
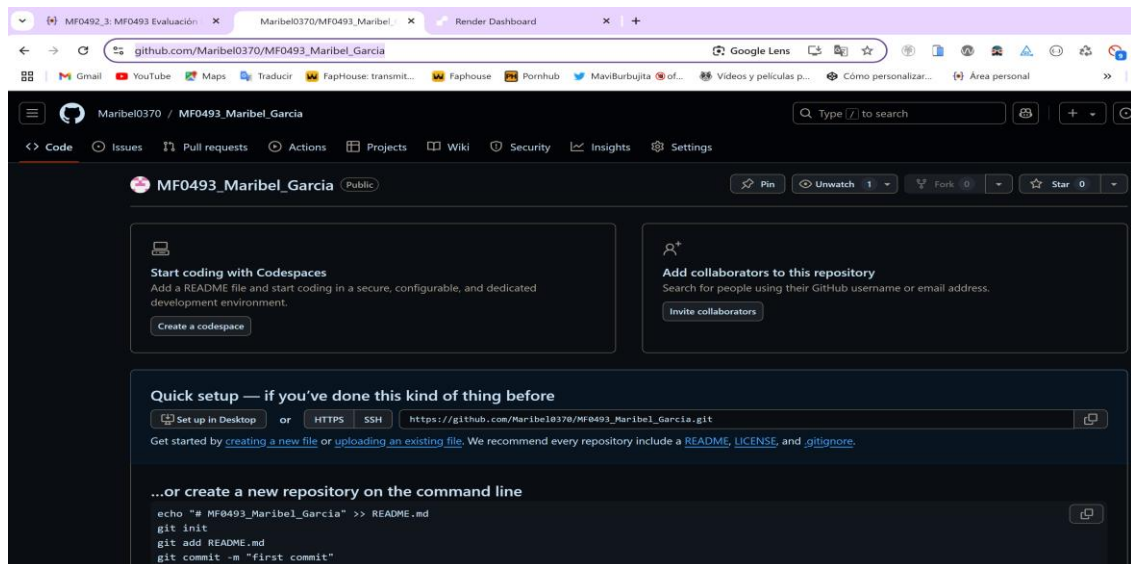
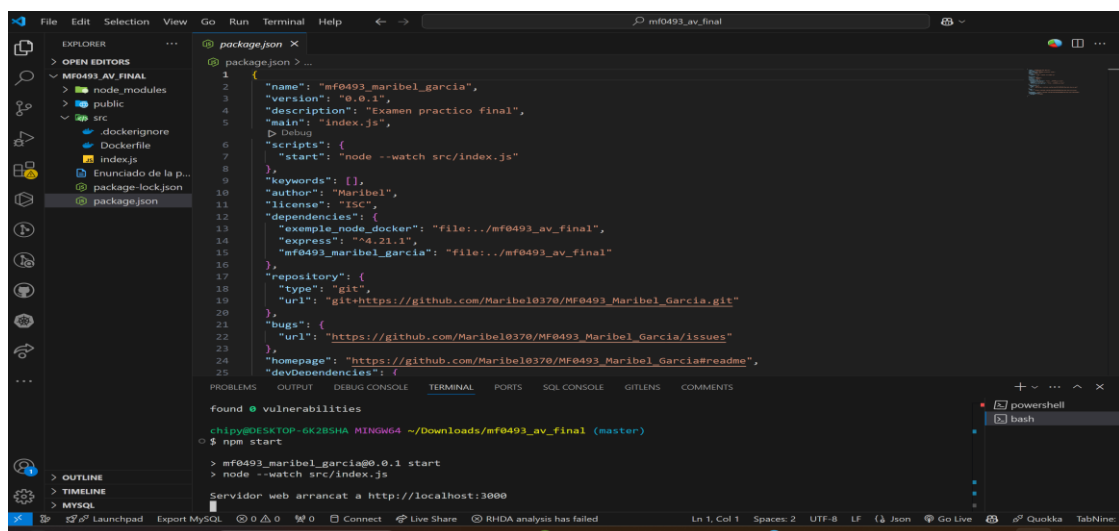


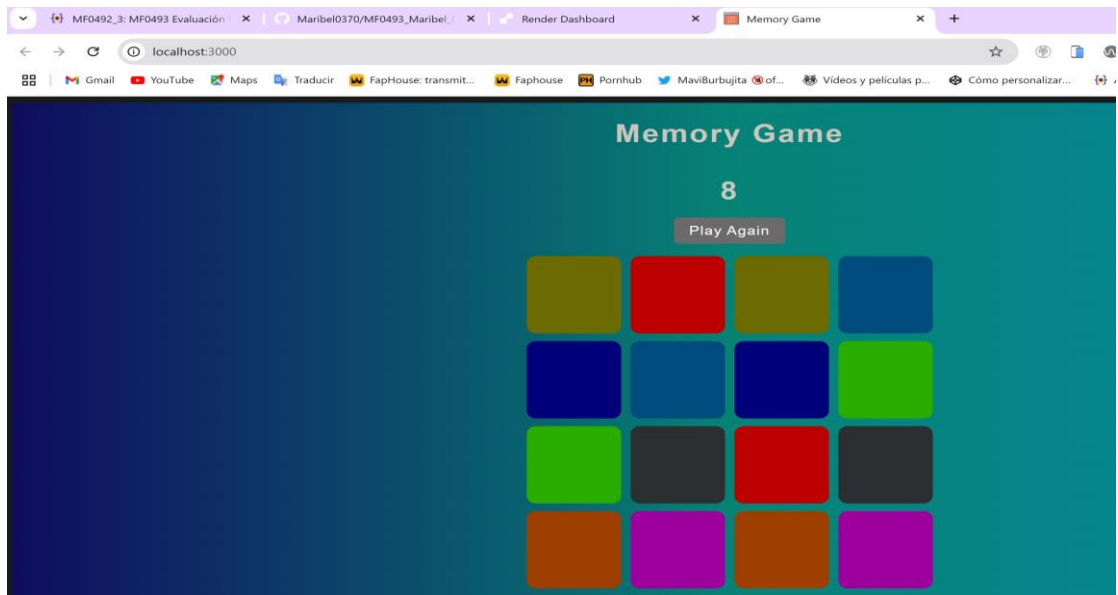
Creo un repositorio en Github y se pone en el archivo package.json



Se instalan las dependencias Express y Morgan y se comprueba que la aplicación está funcionando



La aplicación funciona correctamente



Se crean los archivos Dockerfile, .Docckerignore y .gitignore

```

1 # Identificar qué imagen queremos utilizar
2 FROM node:23.0-slim
3
4 # Indicar el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar los archivos de package.json y package-lock.json
8 COPY package*.json ./
9
10 # Instalar las dependencias
11 RUN npm install \
12     && npm cache clean --force \
13     && rm -rf /tmp/* /root/.npm_cacache
14
15 # Copiar el resto de los archivos
16 COPY . .
17
18 # Exponer el puerto en el que la aplicación se ejecutará
19 EXPOSE 3000
20
21 # Comando para ejecutar la aplicación
22 CMD ["npm", "start"]
  
```

```

Node.js v23.0.0

chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$ node -v
v23.0.0
chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$
  
```

```

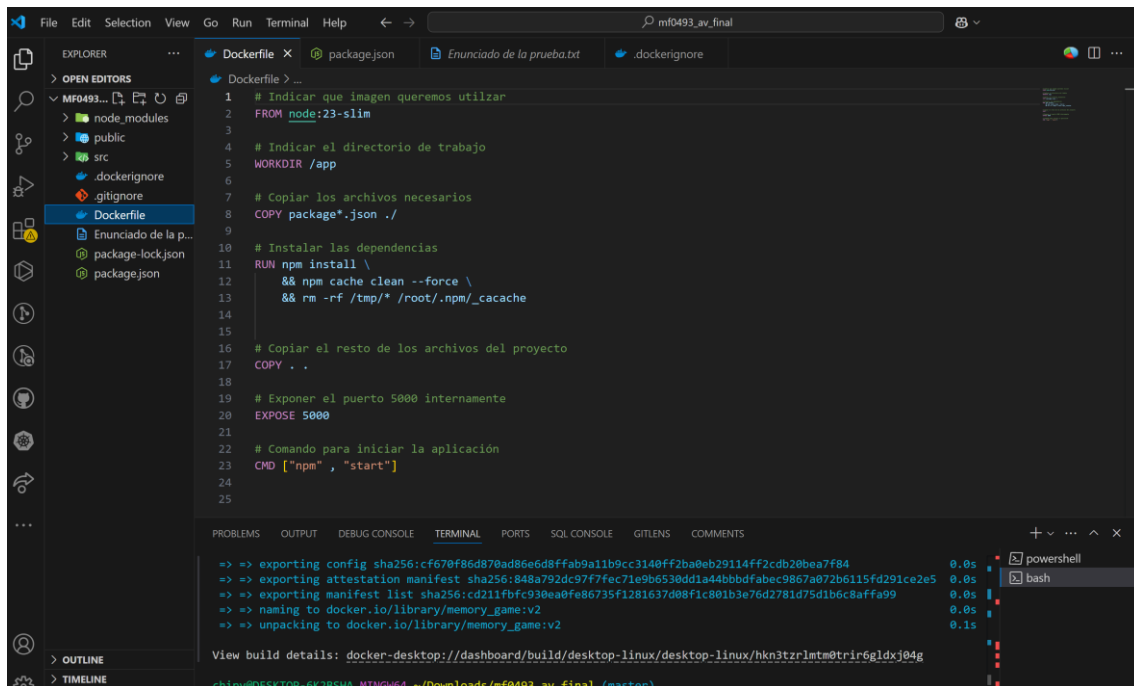
1 {
2   "name": "mf0493_maribel_garcia",
3   "version": "0.0.1",
4   "description": "Examen practico final",
5   "main": "index.js",
6   "scripts": {
7     "start": "node src/app.js",
8     "dev": "node --watch src/app.js",
9     "test": "echo \\\"Error: no test specified\\\" && exit 1"
10  },
11   "keywords": [],
12   "author": "Maribel",
13   "license": "ISC",
14   "dependencies": {
15     "example_node_docker": "file:../mf0493_av_final",
16     "express": "^4.21.1",
17     "mf0493_maribel_garcia": "file:../mf0493_av_final"
18  },
19   "repository": {
20     "type": "git",
21     "url": "git+https://github.com/Maribel0370/MF0493_Maribel_Garcia.git"
22  },
23   "bugs": {
24     "url": "https://github.com/Maribel0370/MF0493_Maribel_Garcia/issues"
25  },
26 }
  
```

```

Node.js v23.0.0

chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$ node -v
v23.0.0
chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$
  
```

Se crea la imagen en Docker

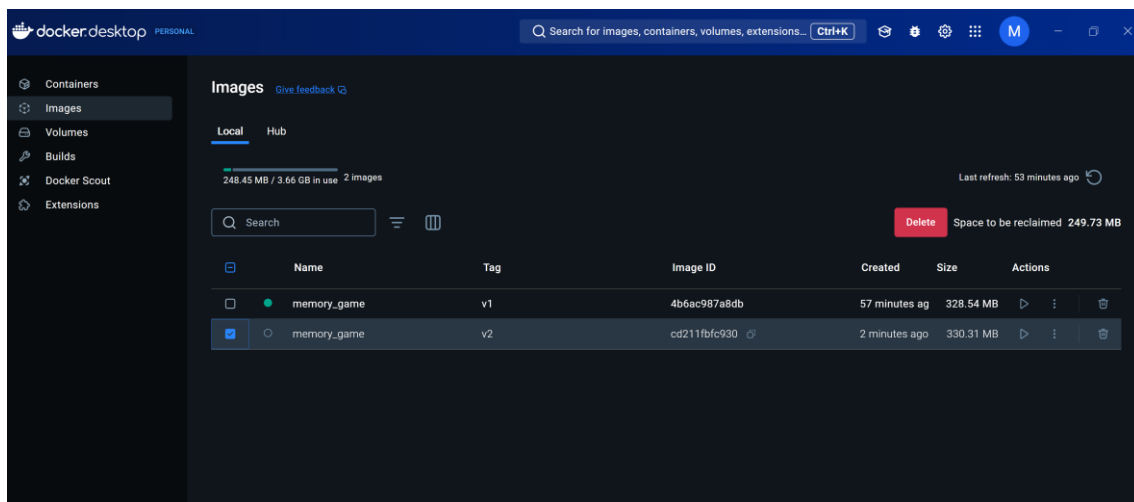


The image shows a Visual Studio Code editor with a Dockerfile open. The Dockerfile contains instructions to build a Docker image for a Node.js application. The terminal at the bottom shows the output of the build process, including the export of the image and the installation of dependencies.

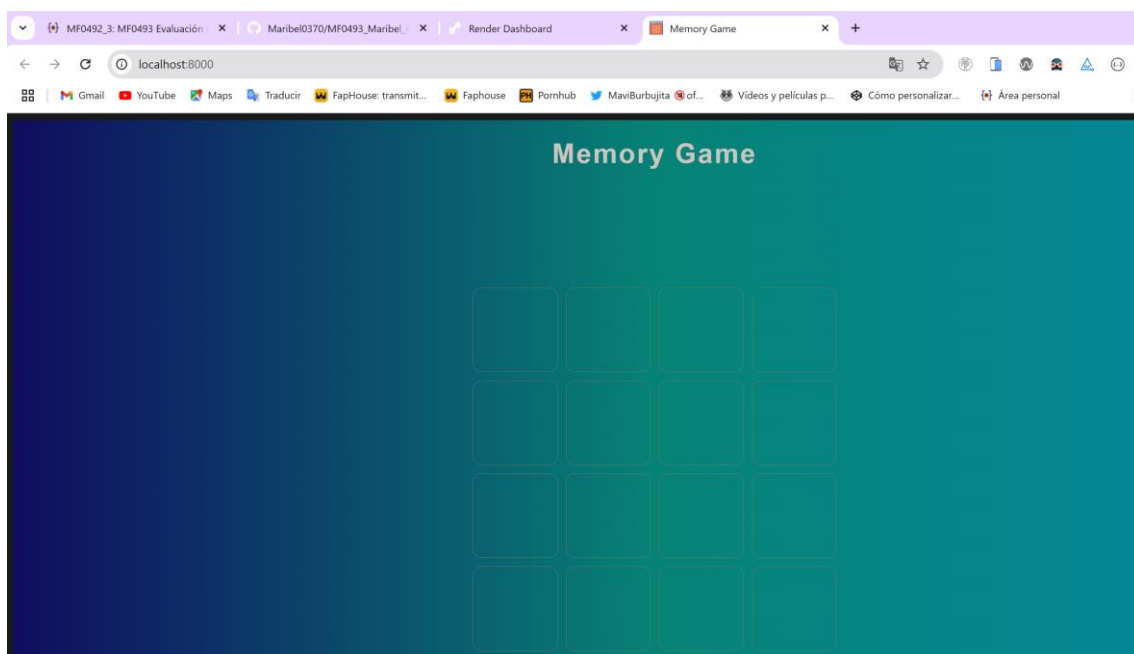
```
1 # Indicar que imagen queremos utilizar
2 FROM node:23-slim
3
4 # Indicar el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar los archivos necesarios
8 COPY package*.json ./
9
10 # Instalar las dependencias
11 RUN npm install \
12     && npm cache clean --force \
13     && rm -rf /tmp/* /root/.npm/_cacache
14
15 # Copiar el resto de los archivos del proyecto
16 COPY . .
17
18 # Exponer el puerto 5000 internamente
19 EXPOSE 5000
20
21 # Comando para iniciar la aplicación
22 CMD ["npm", "start"]
23
24
25
```

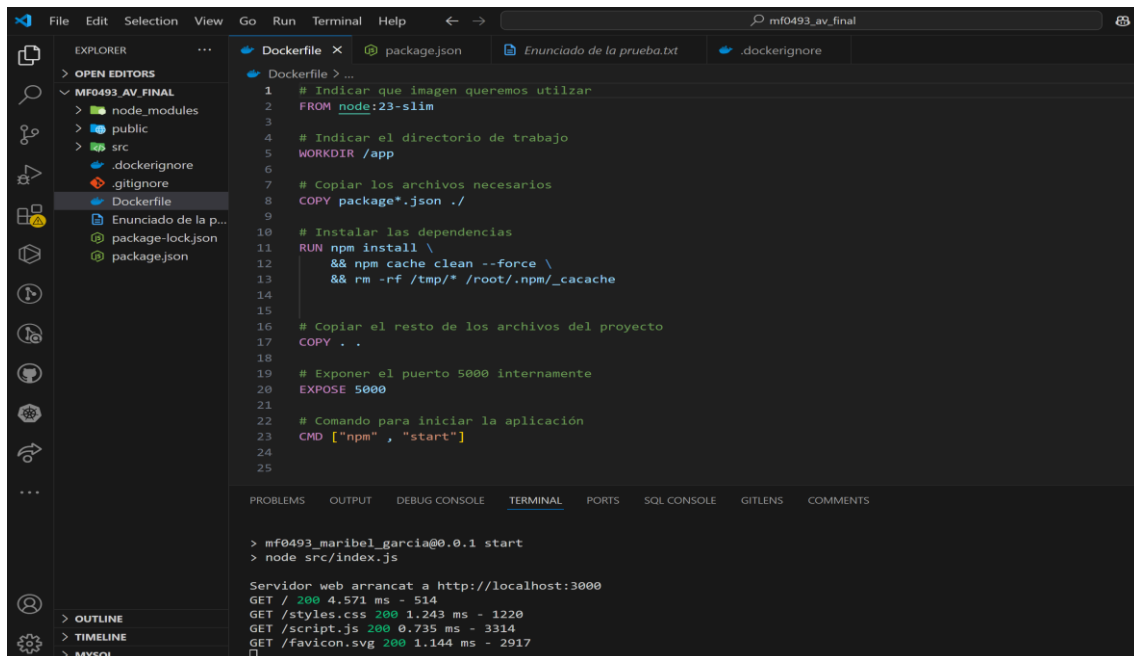
Terminal output:

```
=> => exporting config sha256:cf670f86d870ad86e6d8ffab9a11b9cc3140ff2ba0eb29114ff2c2b20bea7f84 0.0s
=> => exporting attestation manifest sha256:848a792dc97f7fec71e9b6530dd1a44bbdbfabec9867a072b6115fd291ce2e5 0.0s
=> => exporting manifest list sha256:cd211fbfc930ea0fe86735f1281637d08f1c801b3e76d2781d75d1b6c8affa99 0.0s
=> => naming to docker.io/library/memory_game:v2 0.0s
=> => unpacking to docker.io/library/memory_game:v2 0.1s
```



Se crea el contenedor para la imagen



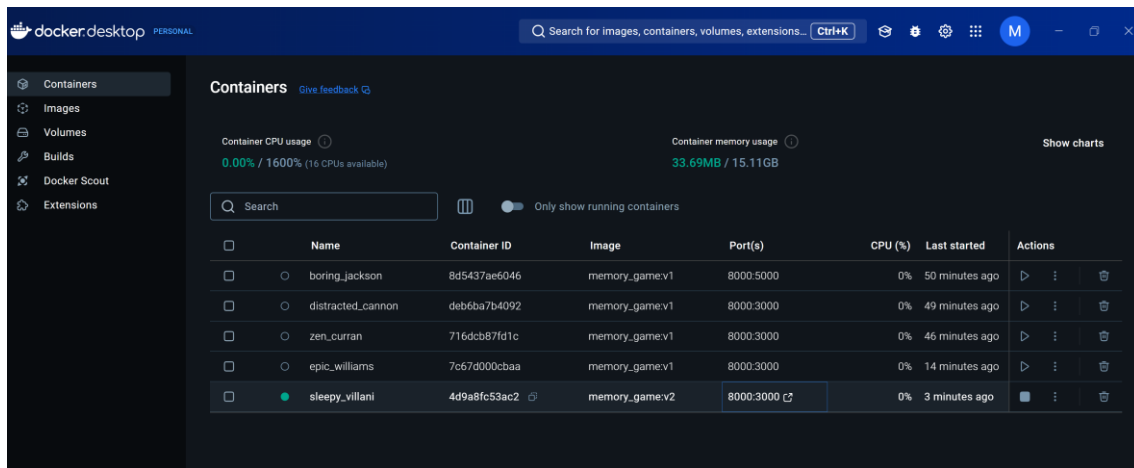


The image shows a Visual Studio Code editor with a Dockerfile open. The Dockerfile contains instructions to build a Node.js application container. The terminal at the bottom shows the command to start the container and the resulting output, which includes the web server starting on http://localhost:3000 and several GET requests being served.

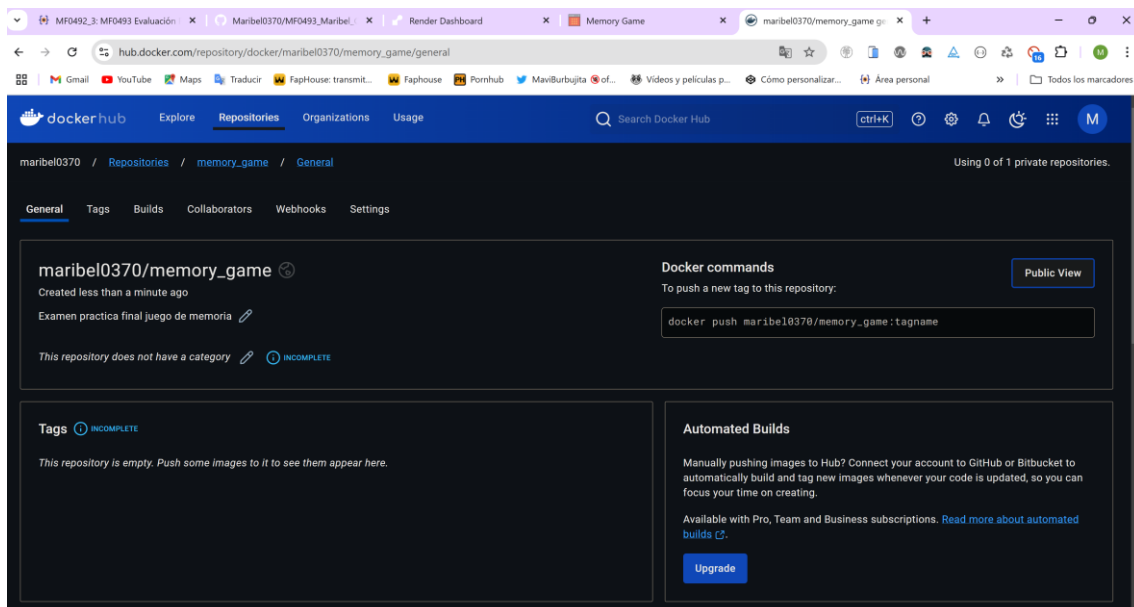
```
1 # Indicar que imagen queremos utilizar
2 FROM node:23-slim
3
4 # Indicar el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar los archivos necesarios
8 COPY package*.json ./
9
10 # Instalar las dependencias
11 RUN npm install \
12     && npm cache clean --force \
13     && rm -rf /tmp/* /root/.npm/_cacache
14
15 # Copiar el resto de los archivos del proyecto
16 COPY . .
17
18 # Exponer el puerto 5000 internamente
19 EXPOSE 5000
20
21 # Comando para iniciar la aplicación
22 CMD ["npm", "start"]
23
24
25
```

```
> mf0493_maribel_garcia@0.0.1 start
> node src/index.js

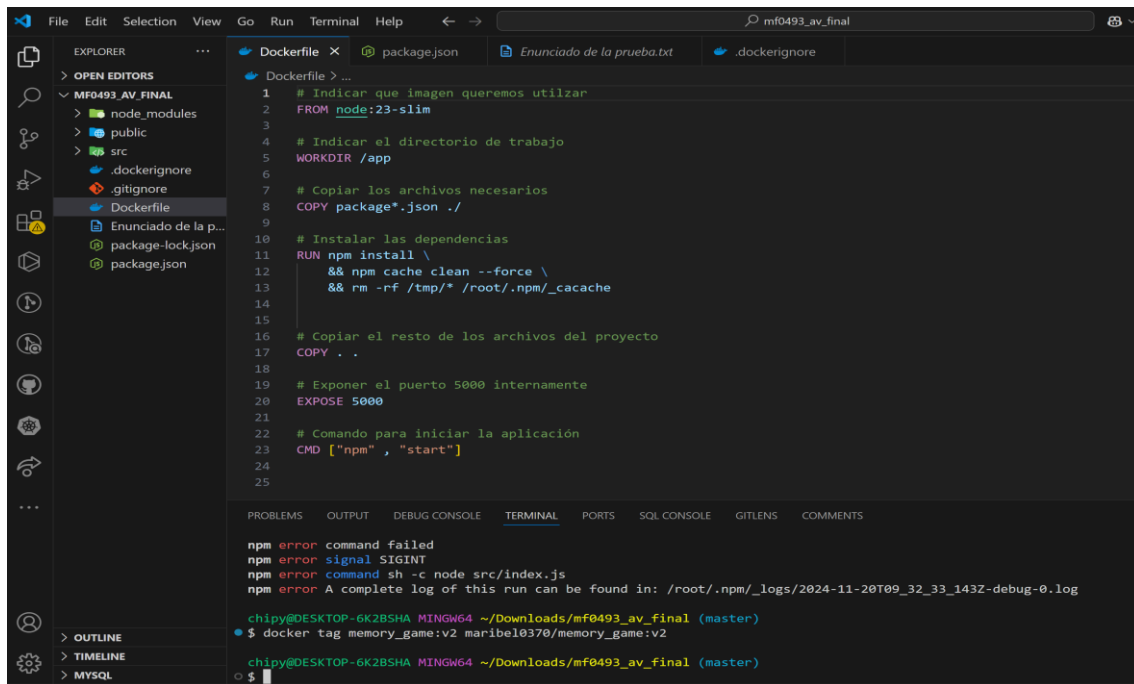
Servidor web arrancat a http://localhost:3000
GET / 200 4.571 ms - 514
GET /styles.css 200 1.243 ms - 1220
GET /script.js 200 0.735 ms - 3314
GET /favicon.svg 200 1.144 ms - 2917
```



Se crea un repositorio en DockerHub



Usuario Docker



The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains instructions to build a Node.js application container. The terminal at the bottom shows the execution of the Docker build command, which fails due to a missing command 'sh' in the CMD instruction.

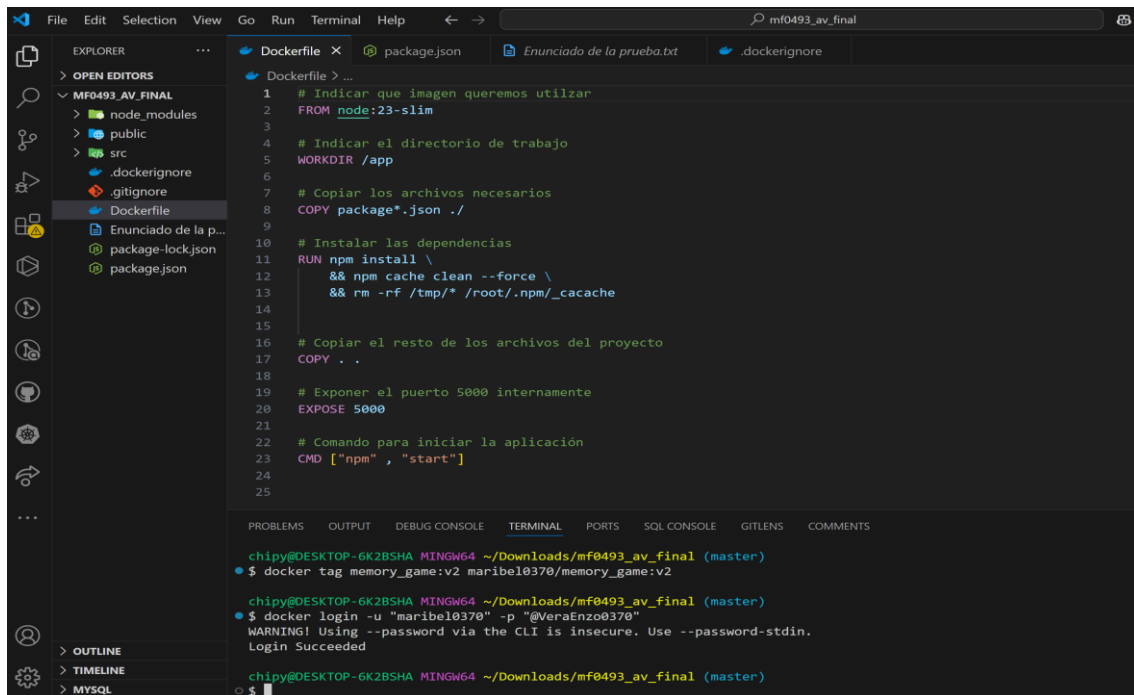
```
1 # Indicar que imagen queremos utilizar
2 FROM node:23-slim
3
4 # Indicar el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar los archivos necesarios
8 COPY package*.json ./
9
10 # Instalar las dependencias
11 RUN npm install \
12     && npm cache clean --force \
13     && rm -rf /tmp/* /root/.npm/_cacache
14
15
16 # Copiar el resto de los archivos del proyecto
17 COPY . .
18
19 # Exponer el puerto 5000 internamente
20 EXPOSE 5000
21
22 # Comando para iniciar la aplicación
23 CMD ["npm", "start"]
24
25
```

```
npm error command failed
npm error signal SIGINT
npm error command sh -c node src/index.js
npm error A complete log of this run can be found in: /root/.npm/_logs/2024-11-20T09_32_33_143Z-debug-0.log

chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$ docker tag memory_game:v2 maribel0370/memory_game:v2

chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$
```

Password Docker



The screenshot shows the Visual Studio Code interface with the same Dockerfile. The terminal now shows the execution of the Docker login command, which is successful. The user is prompted to enter a password, and the login is successful.

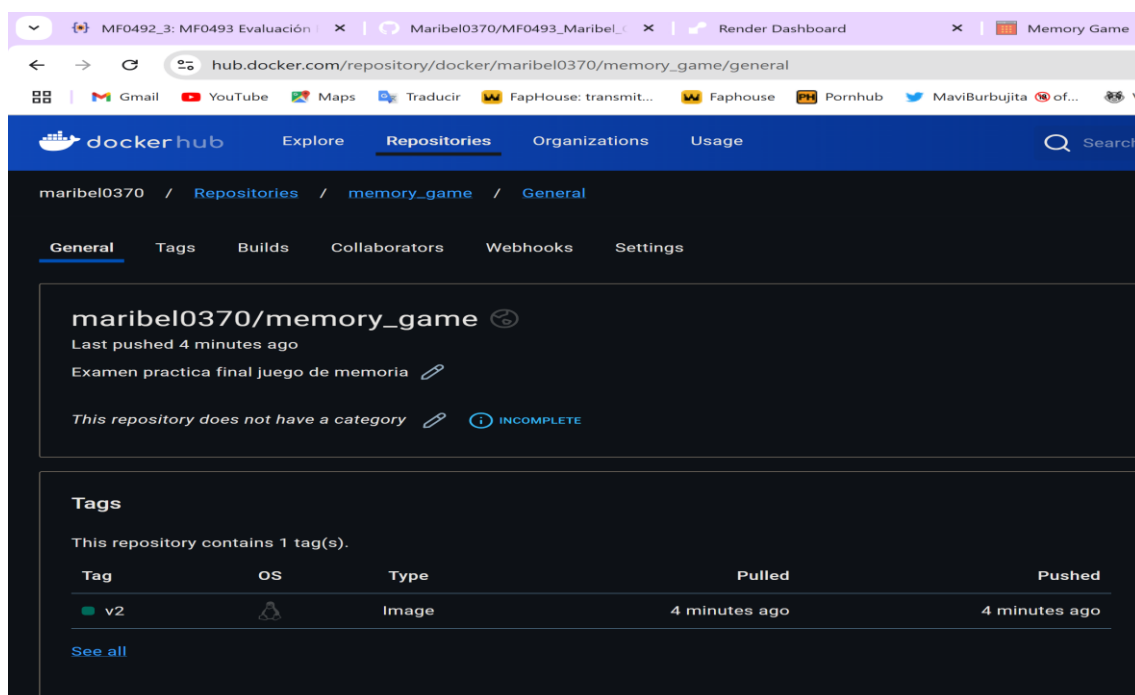
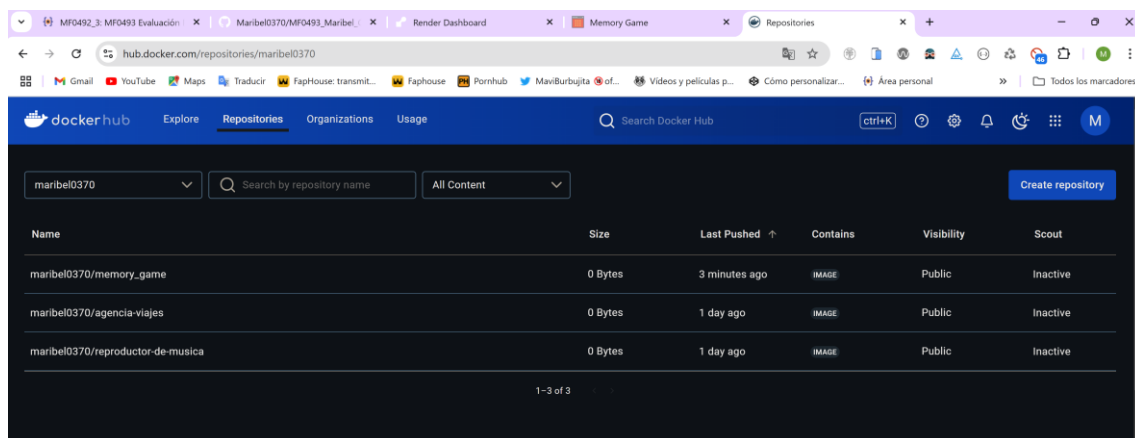
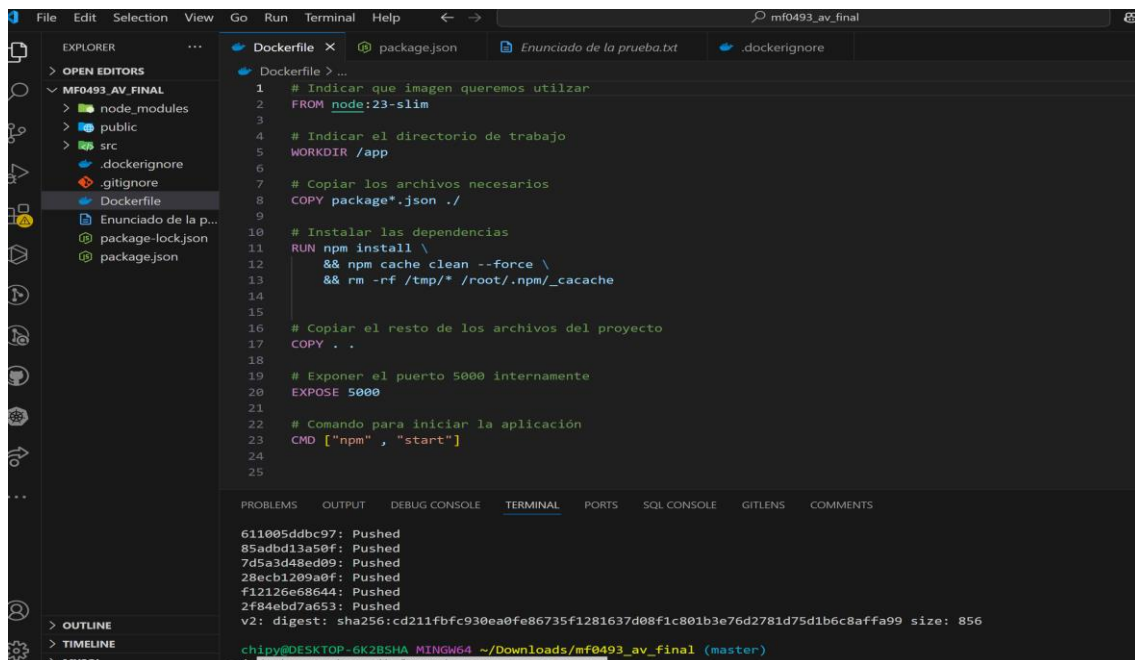
```
1 # Indicar que imagen queremos utilizar
2 FROM node:23-slim
3
4 # Indicar el directorio de trabajo
5 WORKDIR /app
6
7 # Copiar los archivos necesarios
8 COPY package*.json ./
9
10 # Instalar las dependencias
11 RUN npm install \
12     && npm cache clean --force \
13     && rm -rf /tmp/* /root/.npm/_cacache
14
15
16 # Copiar el resto de los archivos del proyecto
17 COPY . .
18
19 # Exponer el puerto 5000 internamente
20 EXPOSE 5000
21
22 # Comando para iniciar la aplicación
23 CMD ["npm", "start"]
24
25
```

```
chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$ docker tag memory_game:v2 maribel0370/memory_game:v2

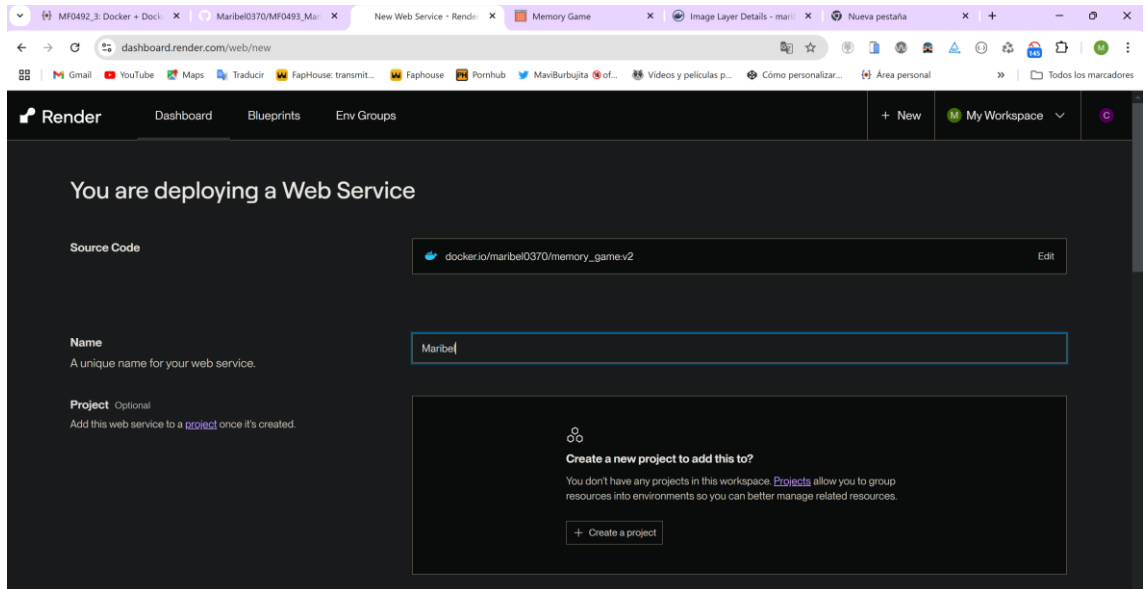
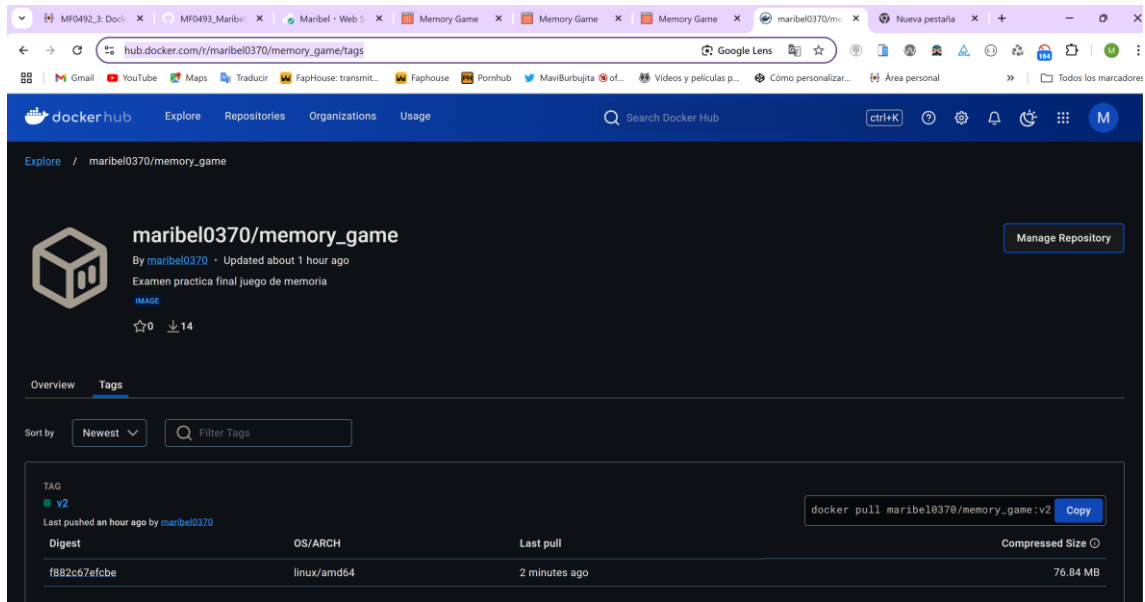
chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$ docker login -u "maribel0370" -p "@VeraEnzo0370"
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded

chipy@DESKTOP-6K2BSHA MINGW64 ~/Downloads/mf0493_av_final (master)
$
```

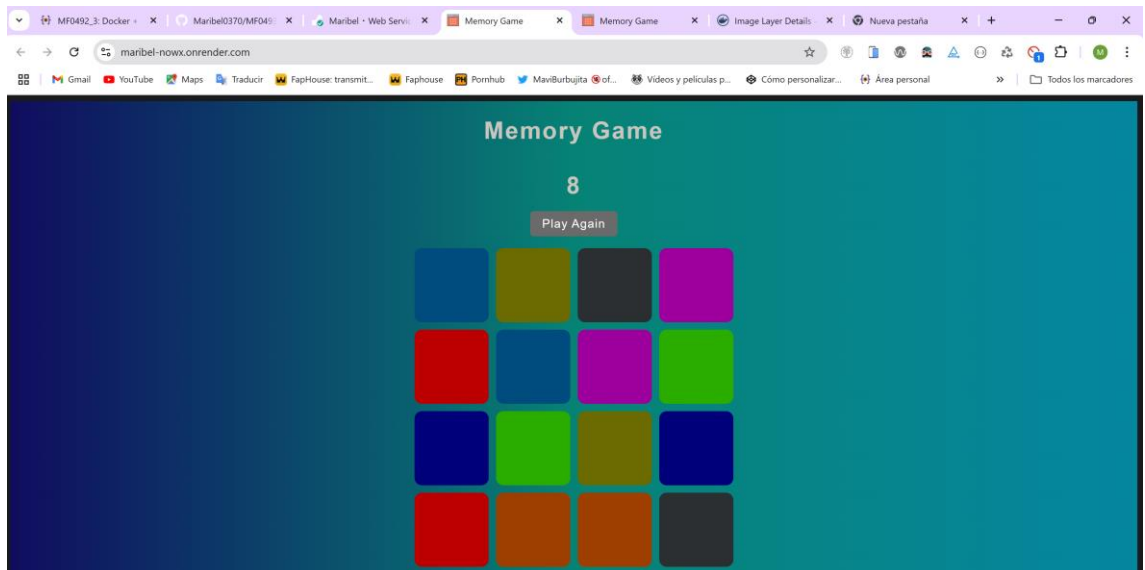
Subir los archivos a Docker

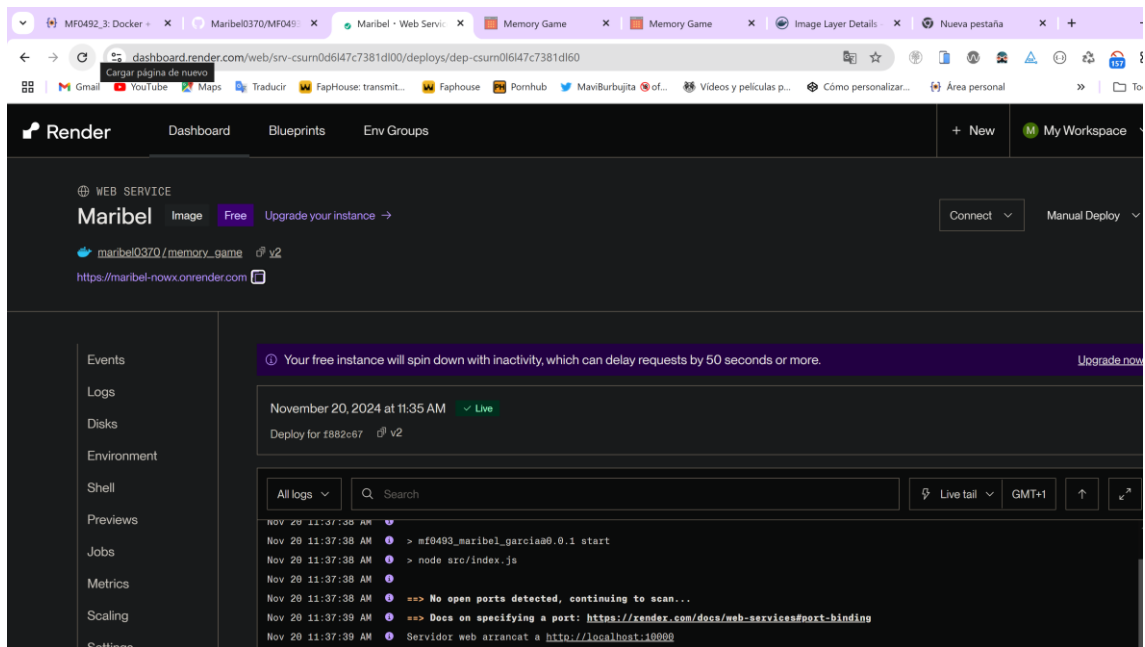


https://hub.docker.com/r/maribel0370/memory_game/tags

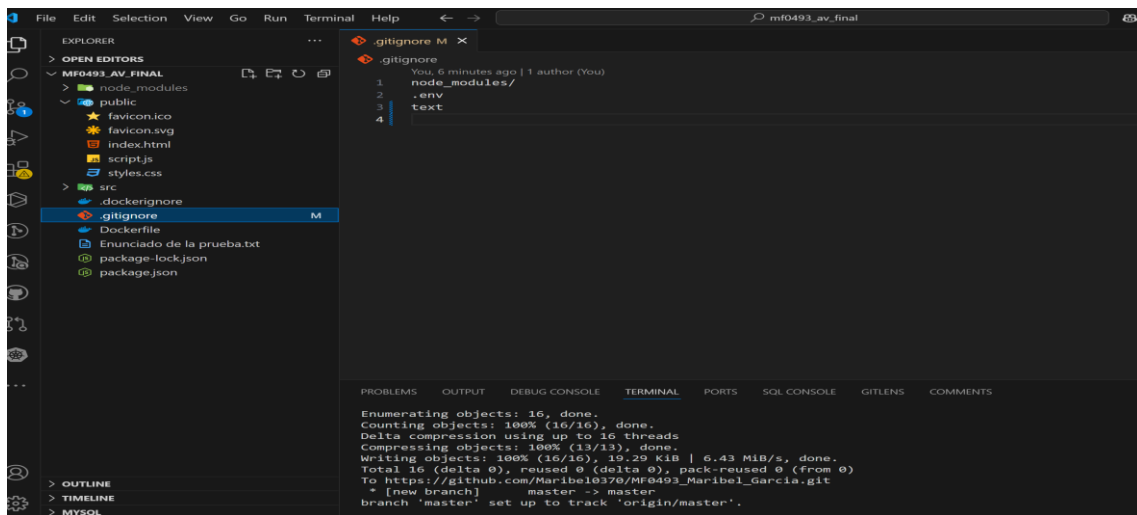


<https://maribel-nowx.onrender.com>





Subimos los archivos a github desde vs code



https://github.com/Maribel0370/MF0493_Maribel_Garcia/tree/master

