

MF0493 EF Manuel Mejía

Memory Game

1. Creamos la carpeta “public” (aunque en este caso ya está creada)

2. Creación de los “package.json” e instalación de las librerías con las siguientes instrucciones:

- npm init -y (Crea el package.json)
- npm install express (para instalar la librería)

```
package name: (memory-game) npm warn init canceled
● mejia@MacBook-Pro-de-Manuel memory-game % npm init -y
Wrote to /Users/mejia/Desktop/memory-game/package.json:

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

● mejia@MacBook-Pro-de-Manuel memory-game % npm install express

added 67 packages, and audited 68 packages in 3s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
mejia@MacBook-Pro-de-Manuel memory-game %
```

3. Creación del fichero “.gitignore”

4. Creación del fichero “.env” con el PORT=3000

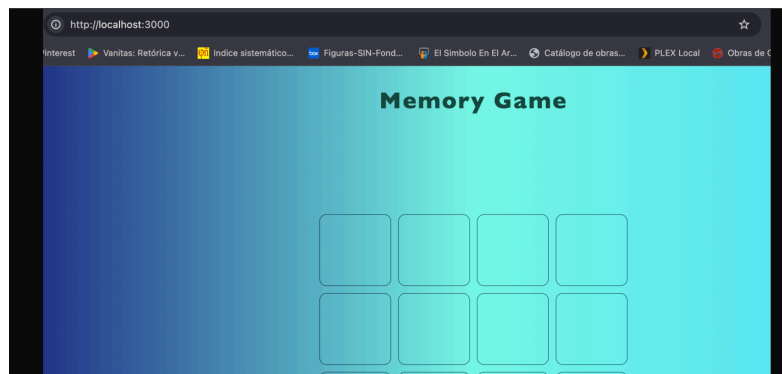
5. En el “package.json” creamos en el comando para iniciar la aplicación

```
"scripts": {
  "start" : "node app.js",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

6. Creamos el fichero “[app.js](#)” que sirve para inicializar el servidor y configurar las rutas y middleware necesarios para la aplicación web, también indicar las rutas para los ficheros estáticos y el puerto de escucha de la app (ver código)

7. Comprobamos que el puerto es escuchado, en este caso en el 3000 con la instrucción

- npm start :
- - > memory-game@1.0.0 start
 - > node app.js
- Servidor escuchando en el puerto <http://localhost:3000>



GitHub

Para subir la “app” a GitHub necesitamos seguir los siguientes pasos

8. Crear un repositorio en GitHub

8.1 Subir el proyecto a github con las siguientes instrucciones:

- git init
- git add README.md
- git commit -m "Memory game"
- git branch -M main
- git remote add origin https://github.com/manu07030/memory_game.git
- git push -u origin main

https://github.com/manu07030/memory_game

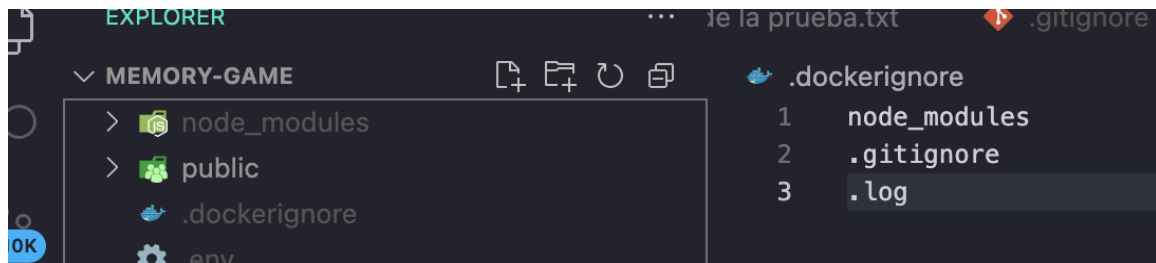
Docker

Haremos la creación de una imagen en Docker con los siguientes pasos:

9. Creación del fichero “dockerfile” donde indicaremos

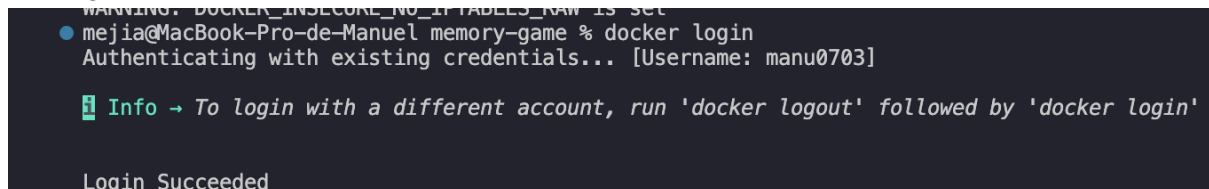
- La imagen base
- Directorio de trabajo dentro del contenedor
- Que copie los archivos de las dependencias
- Que instale las dependencias
- Que copie el resto del código
- Que exponga el puerto
- Los comandos para ejecutar la app
(Ver código)

9.1. Creación del fichero “.dockerignore”



9.2 Iniciar docker.desktop

9.3 Login de docker



9.2. Creación de la imagen con la instrucción:

- `docker build -t memory-game .`

```
Login Succeeded
mejia@MacBook-Pro-de-Manuel memory-game % docker build -t memory-game .
[*] Building 7.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring Dockerfile: 635B
=> [internal] load metadata for docker.io/library/node:24-slim
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 68B
=> [1/5] FROM docker.io/library/node:24-slim@sha256:dd773e49e639dafc0b425bf9ac1d74705086b9aebac960706ab146c5dfa0b32d
=> => resolve docker.io/library/node:24-slim@sha256:dd773e49e639dafc0b425bf9ac1d74705086b9aebac960706ab146c5dfa0b32d
=> [internal] load build context
=> => transferring context: 158.82kB
=> CACHED [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 sha256:3beb61ac4995cae61ad82f81cc61a7408ac2671f6f43bf5ed2c3172c1942c830
=> => exporting config sha256:361b8c652a634f2dd7e9115be4e8fd59a089021936381a34e60b78c29d1f07a9
=> => exporting attestation manifest sha256:8b06360ced41576e6130288dd58a1b104bc53eaf50419308f74f9c8a0697cf6
=> => exporting manifest list sha256:8c00a30fba72833735d18a5c36a308505f04f5c45bd3513aba80cd6fd42a537c
=> => naming to docker-desktop://dashboard/build/desktop-linux/desktop-linux/mhct27azwpoc3015ih0dulqdc
=> => unpacking to docker-desktop://dashboard/build/desktop-linux/desktop-linux/mhct27azwpoc3015ih0dulqdc (cmd + click)

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mhct27azwpoc3015ih0dulqdc
mejia@MacBook-Pro-de-Manuel memory-game %
```

9.3 Comprobar si se creó la imagen con la instrucción

- `docker images`

```
=> => unpacking to docker-desktop://dashboard/build/desktop-linux/desktop-linux/mhct27azwpoc3015ih0dulqdc (cmd + click)
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mhct27azwpoc3015ih0dulqdc
mejia@MacBook-Pro-de-Manuel memory-game % docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
memory-game         latest      8c00e30fbe28  3 minutes ago 337MB
piedra-papel-tijera latest      fb82167737fe 24 hours ago 338MB
```

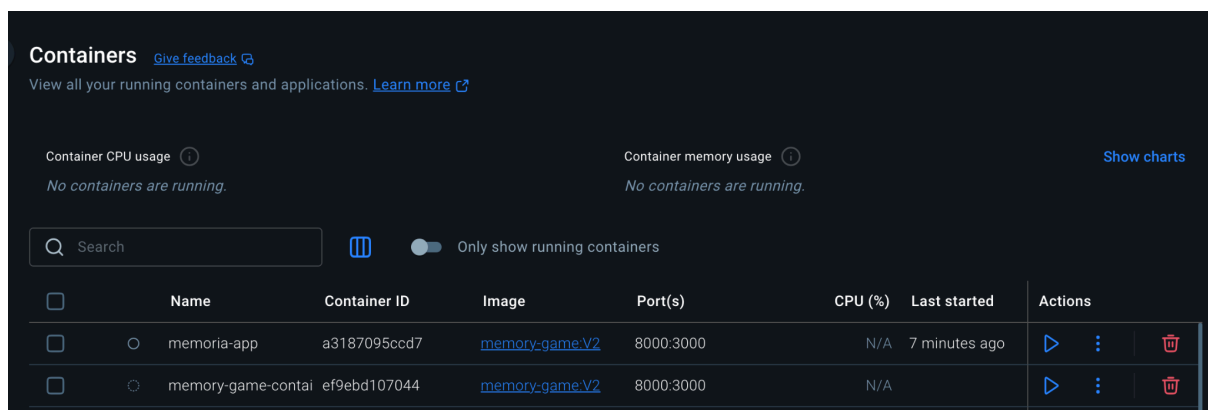
9.3.1 Renombrar el tag

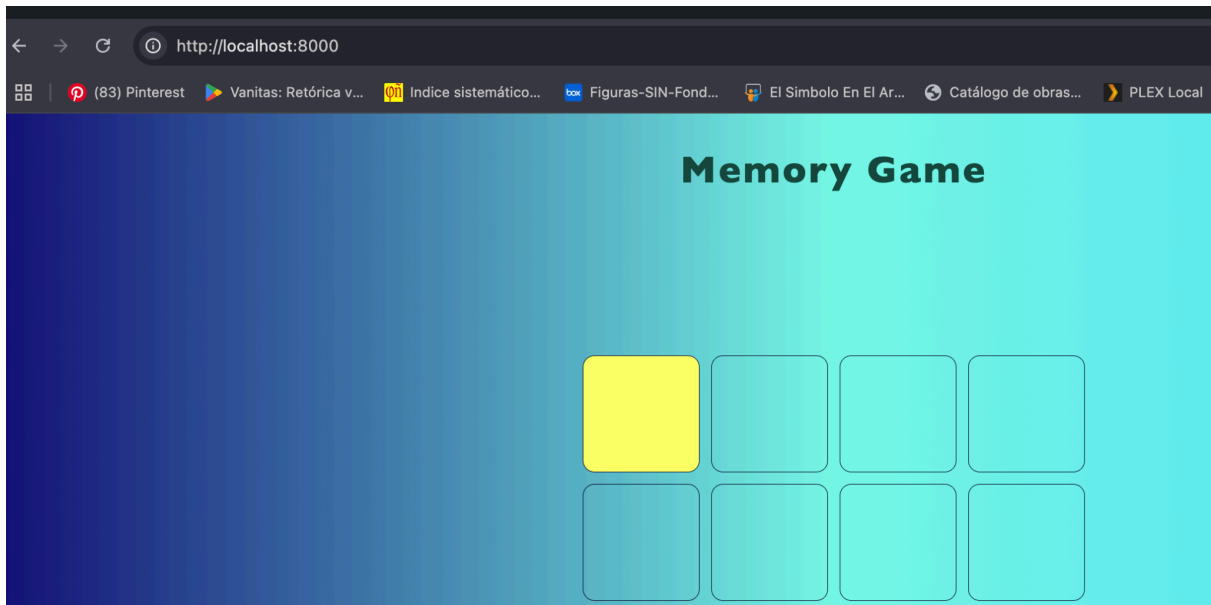
- `docker tag memory-game:latest memory-game:V2`

```
mejia@MacBook-Pro-de-Manuel memory-game % docker tag memory-game:latest memory-game:V2
mejia@MacBook-Pro-de-Manuel memory-game % docker run -d -p 8000:3000 memory-game:v2
ff1c1e0dbbadda3746c83ea5acbbf08080701c375ae8e0071c6e8e90e9ce5c11
mejia@MacBook-Pro-de-Manuel memory-game %
```

9.4. Levantar el puerto con el nombre del container

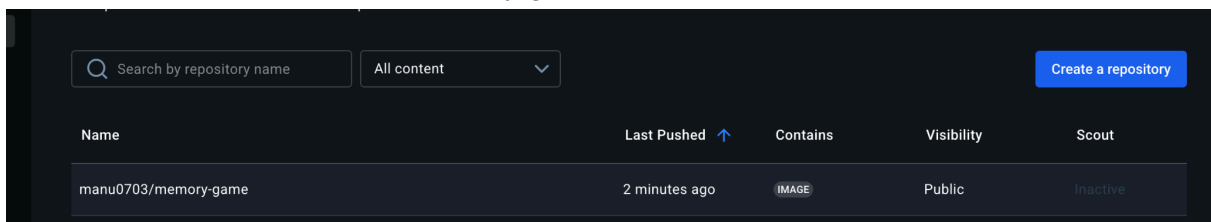
- `docker run -d -p 8000:3000 --name memoria-app memory-game:V2`





9.5 Subir la app al repositorio de docker hub

- `docker push manu0703/memory-game:v2`



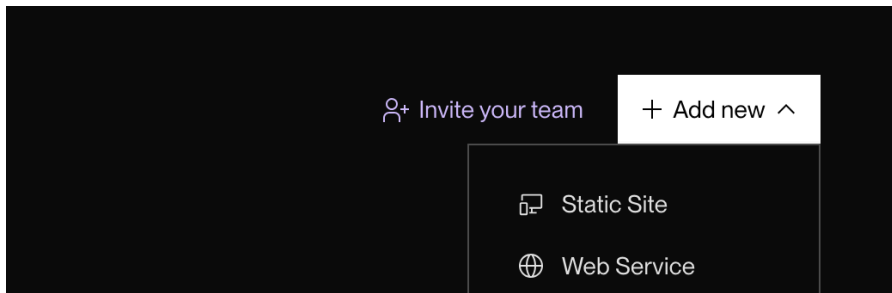
<https://hub.docker.com/repository/docker/manu0703/memory-game/general>

Render

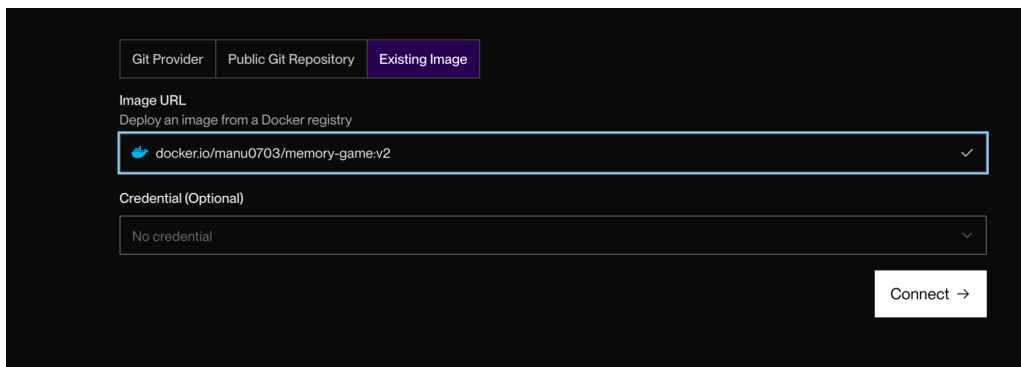
Para subir el proyecto en la nube necesitamos

10. Ir a la siguiente dirección e iniciar sesión

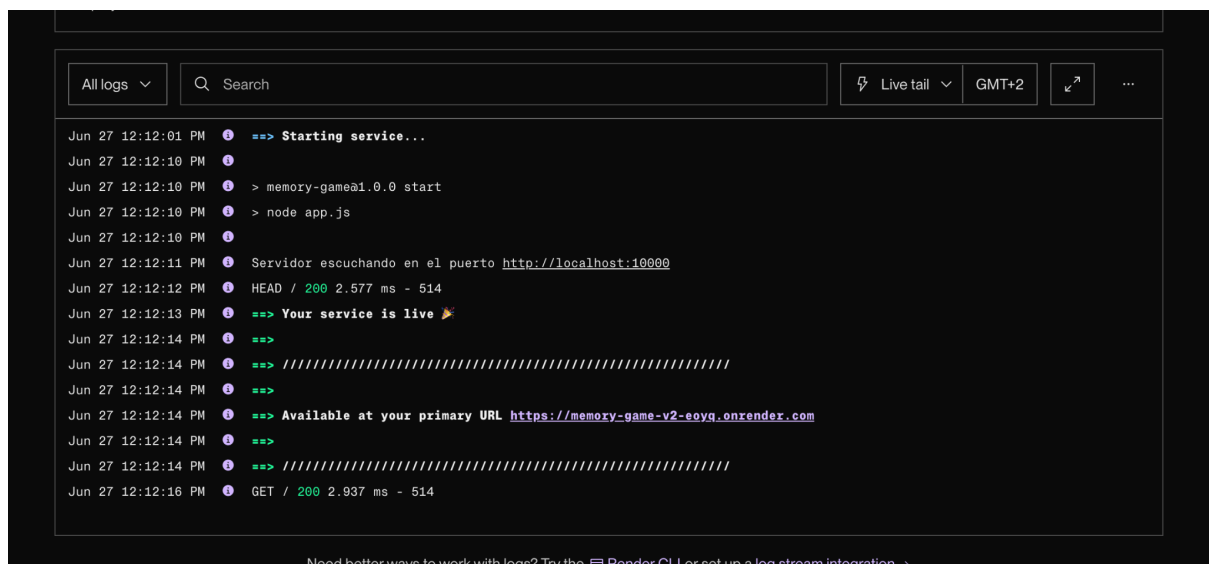
- <https://dashboard.render.com/>
- Agregar nuevo proyecto
- Web service



10.1 Indicar que el código fuente viene de una imagen existente de docker e indicar la URL del repositorio de dockerHub



10.2 Elegir la región a desplegar la app en la nube y desplegar la Web service



<https://memory-game-v2-eoyq.onrender.com>