

Pasos para Docker

MF0493_3 Manami Naito

Abrir la carpeta con VSC

Instalar dependencias

Terminal > npm i

Ver si aplicación funciona bien

Terminal > node app.js

Hacer el fichero de .gitignore

```
.*
.* /
node_modules
```

Iniciar Git

Terminal > git init

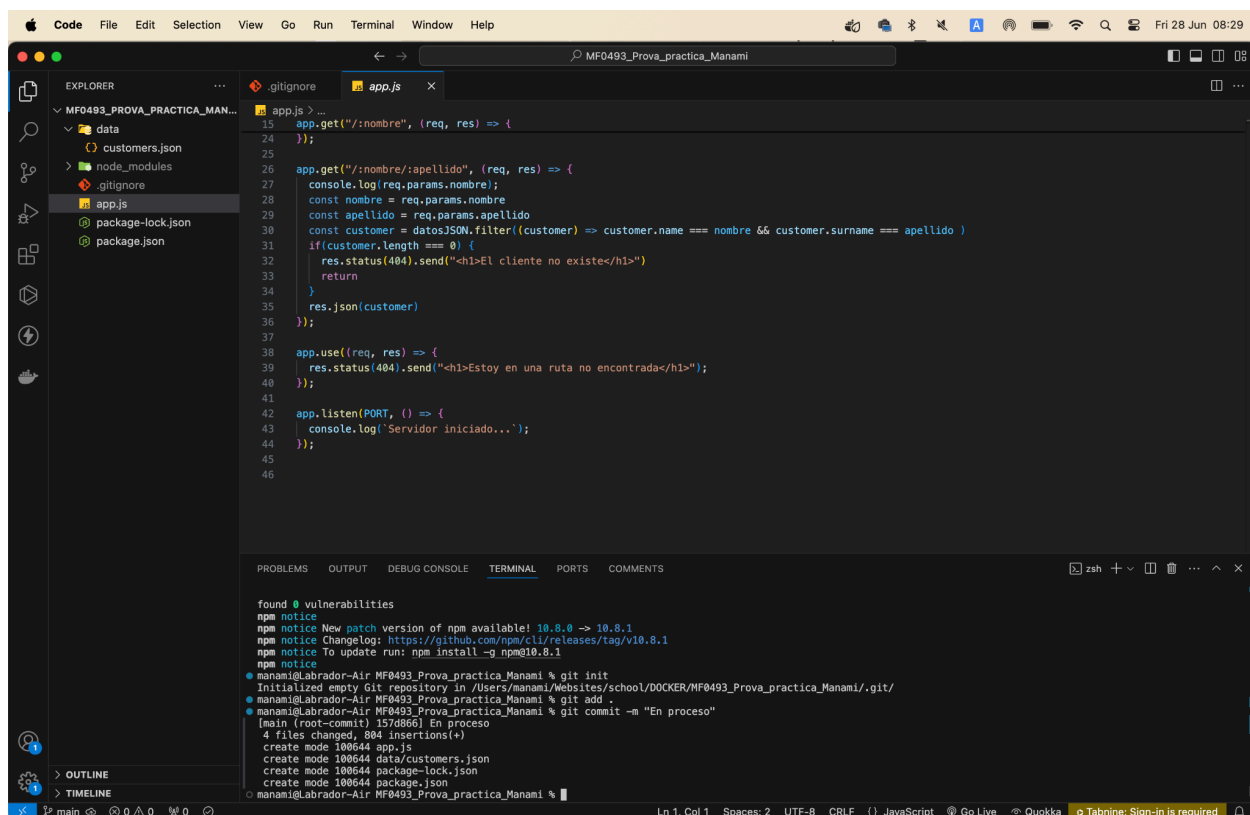
(en mi caso todo configurado en ordenador)

Confirmar ficheros/carpetas indicados en .gitignore están fuera de Git, y agrega todo

Terminal > git add .

Primero Commit

Terminal > git commit -m "En proceso"



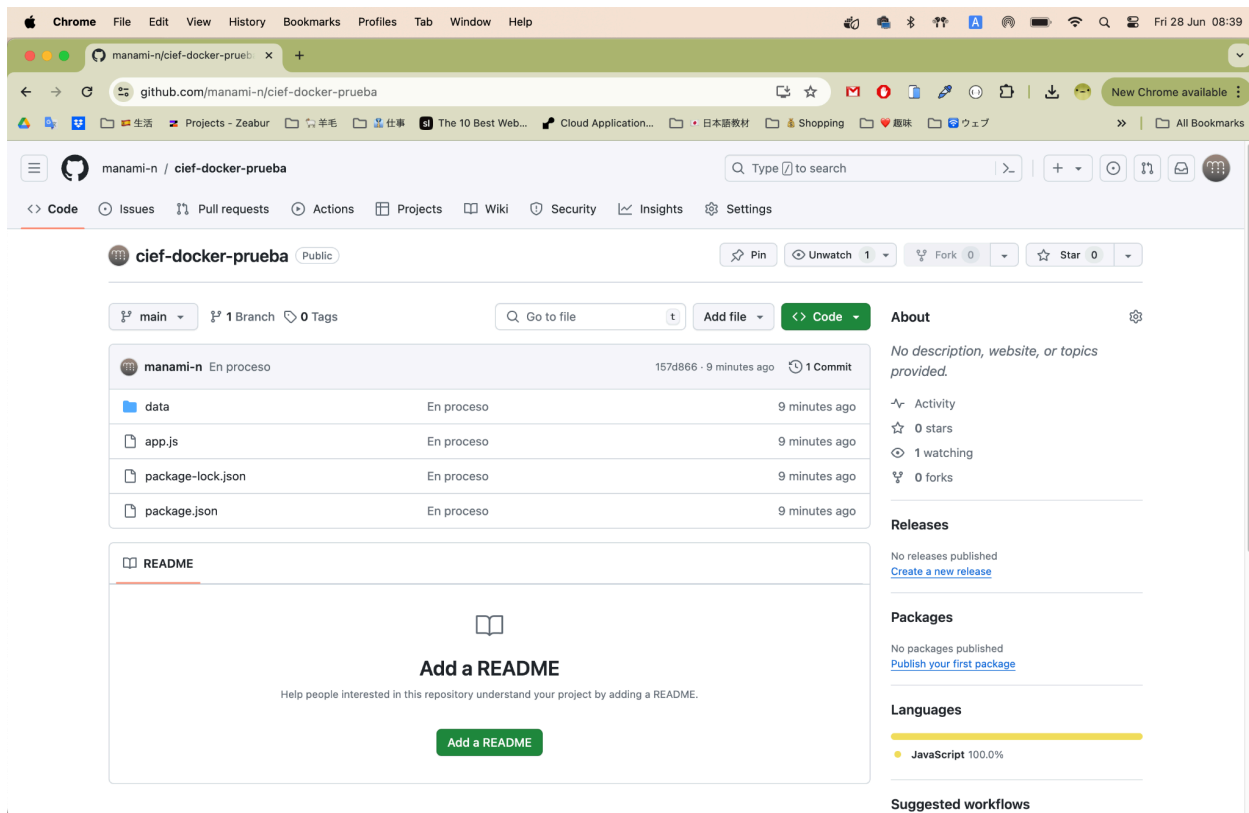
```
Code File Edit Selection View Go Run Terminal Window Help
MF0493_Prova_practica_Manami
EXPLORER
MF0493_PROVA_PRACTICA_MANAMI
data
customers.json
node_modules
.gitignore
app.js
package-lock.json
package.json
app.js
15 app.get("/:nombre", (req, res) => {
16   });
17
18 app.get("/:nombre:apellido", (req, res) => {
19   console.log(req.params.nombre);
20   const nombre = req.params.nombre;
21   const apellido = req.params.apellido;
22   const customer = datosJSON.filter((customer) => customer.name === nombre && customer.surname === apellido);
23   if(customer.length === 0) {
24     res.status(404).send("<h1>El cliente no existe</h1>");
25     return;
26   }
27   res.json(customer);
28 });
29
30 app.use((req, res) => {
31   res.status(404).send("<h1>Estoy en una ruta no encontrada</h1>");
32 });
33
34 app.listen(PORT, () => {
35   console.log('Servidor iniciado...');
36 });
37
38
39
40
41
42
43
44
45
46
TERMINAL
found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 10.8.0 -> 10.8.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.1
npm notice To update run: npm install -g npm@10.8.1
npm notice
manami@labrador-Air MF0493_Prova_practica_Manami % git init
Initialized empty Git repository in /Users/manami/Websites/school/DOCKER/MF0493_Prova_practica_Manami/.git/
manami@labrador-Air MF0493_Prova_practica_Manami % git add .
manami@labrador-Air MF0493_Prova_practica_Manami % git commit -m "En proceso"
[main (root-commit) 1570866] En proceso
4 files changed, 804 insertions(+)
create mode 100644 app.js
create mode 100644 data/customers.json
create mode 100644 package-lock.json
create mode 100644 package.json
manami@labrador-Air MF0493_Prova_practica_Manami %
```

Push a Github

Terminal > git remote add origin https://github.com/manami-n/cief-docker-prueba.git

Terminal > git branch -M main

Terminal > git push -u origin main



(<https://github.com/manami-n/cief-docker-prueba>)

*Esta vez no agregué este enlace a package.json

Hacer el fichero Dockerfile

```
# Que imagen utilizamos
FROM --platform=linux/amd64 node:22

# Definir la carpeta
WORKDIR /app

# Copy package.json and package-lock.json first to leverage Docker cache
# COPY package*.json ./

# Copiar todos al directorio
COPY . .

# Obtener las dependencias
RUN npm i

# Arrancar la aplicacion
CMD ["npm", "start"]
```

(en mi caso, indicar plataforma de linux/amd64)

Instalar base image

Terminal > docker pull node:22

Mirar si script está correcto en package.json

```
"scripts": {  
  "start": "node app.js",  
},
```

Crear el fichero de .dockerignore

```
node_modules/
```

```
*.log
```

```
*.*
```

```
*.* /
```

Build image

Terminal > docker build --no-cache -t manami0/docker-prueba:1.0.3 .

(en mi caso de mac, --no-cache para eliminar todos los caches antiguos para asegurar linux/amd64)

Mirar si funciona bien

Terminal > docker run manami0/docker-prueba:1.0.3

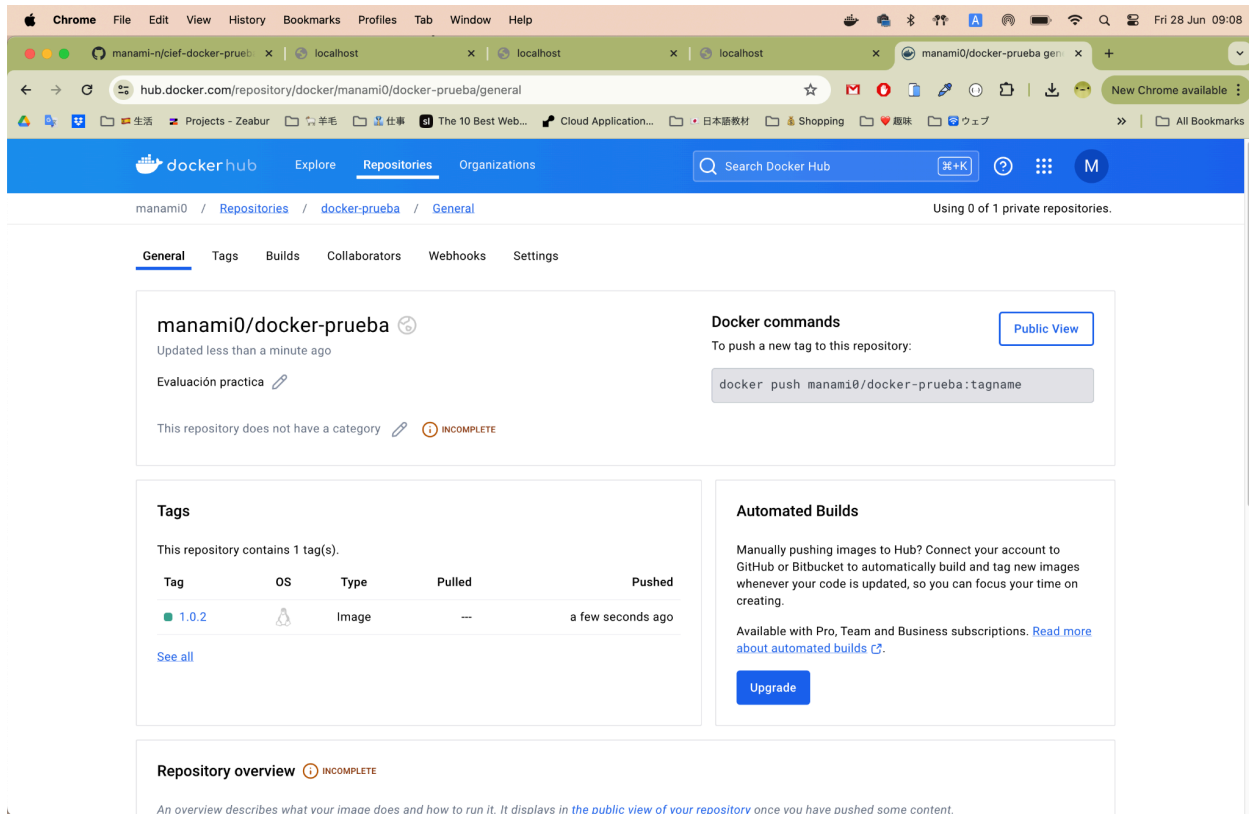
(el mio aparece AMD64 mensaje porque estoy forzando la plataforma diferente que la mía)

The screenshot shows the Docker Desktop application window. The top bar includes the title 'Docker Desktop' and a search bar. The main area is titled 'Containers' and displays system metrics: 'Container CPU usage' at 0.00% / 800% (8 CPUs available) and 'Container memory usage' at 79.73MB / 3.74GB. A table lists the running containers. One container is shown with the name 'hardcore_mcnulty', ID '0f6e47817798', image 'manami0/docker-prueba:1.0.3', status 'Running', ports '4000:3000', CPU usage '0%', and 'Last started' '18 seconds ago'. The bottom status bar shows 'RAM 2.01 GB', 'CPU 0.00%', 'Disk 55.75 GB avail. of 62.67 GB', and version 'v4.31.0'.

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
hardcore_mcnulty 0f6e47817798	manami0/docker-prueba:1.0.3	Running	4000:3000	0%	18 seconds ago	[Stop] [Refresh] [Delete]

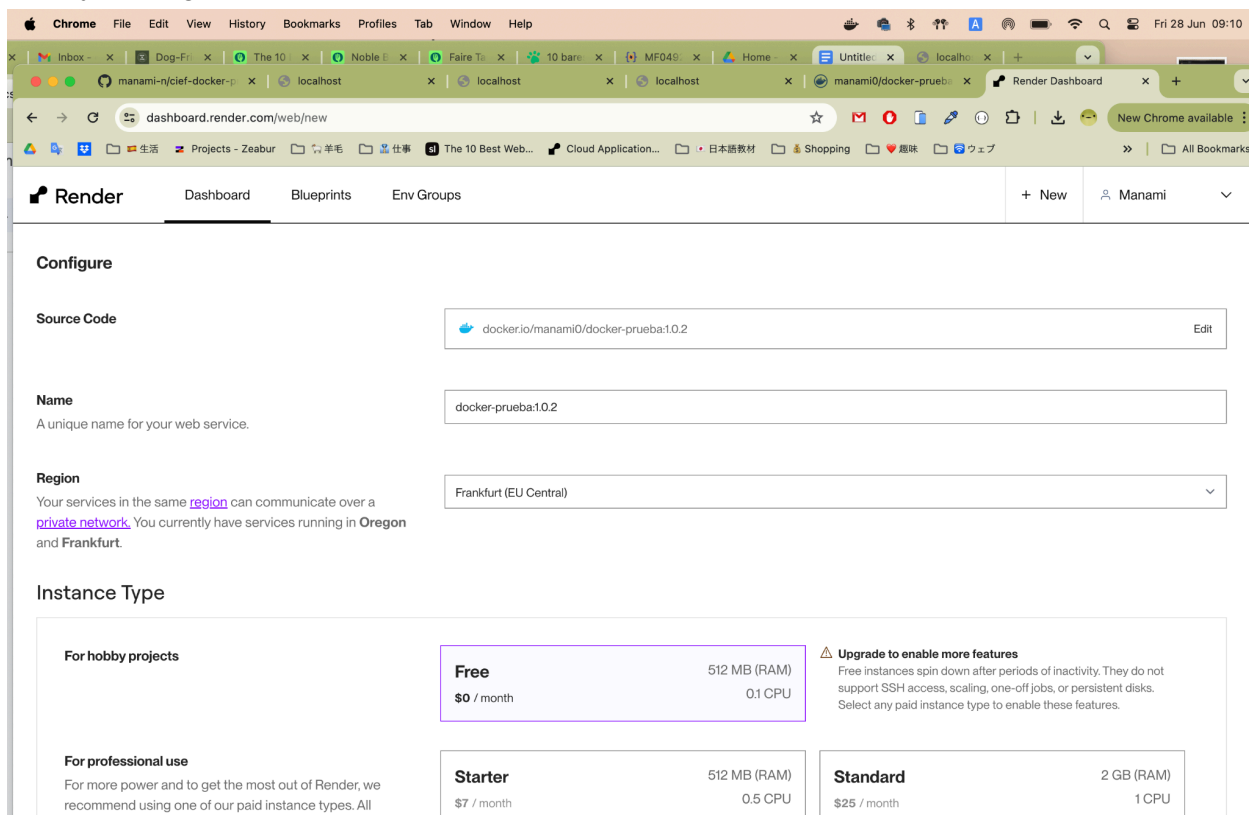
Subir el contenedor a Dockerhub publico

Terminal > docker push manami0/docker-prueba:1.0.2



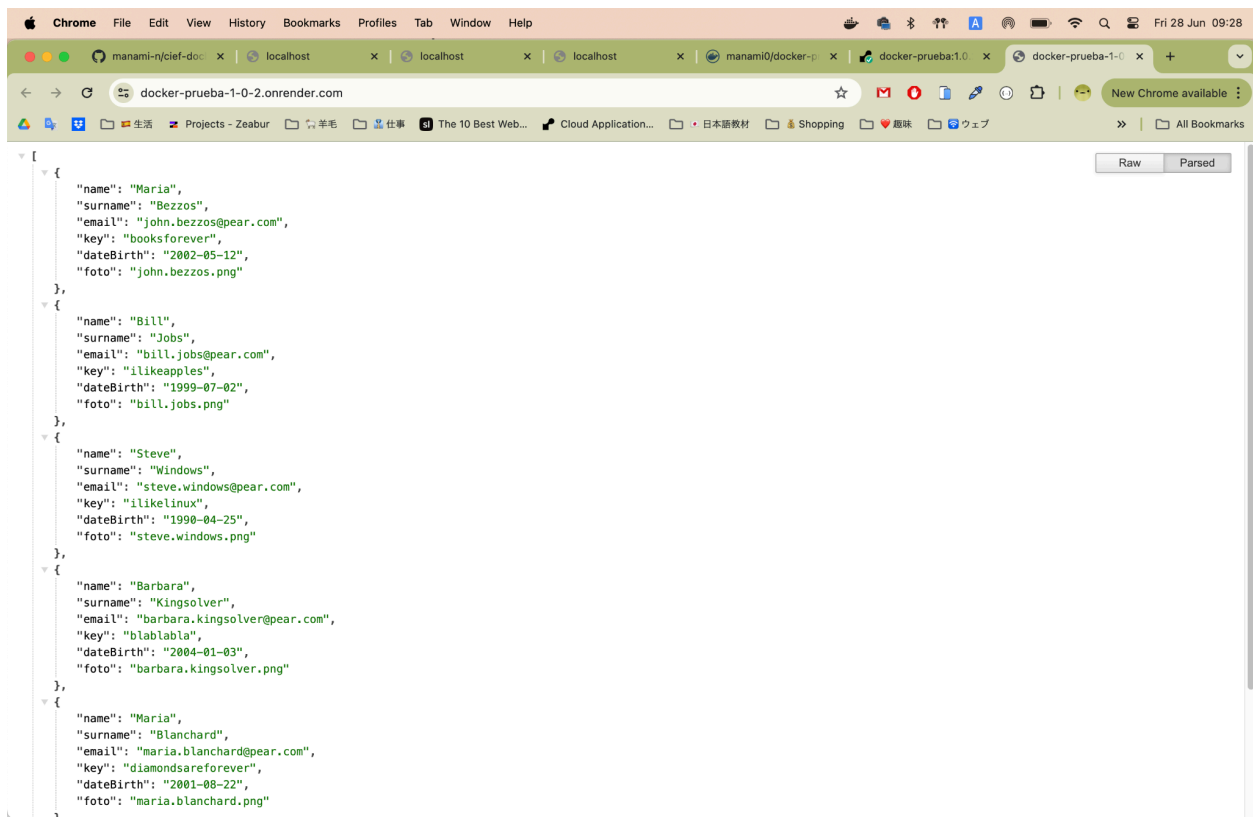
(<https://hub.docker.com/repository/docker/manami0/docker-prueba/general>)

Deploy la imagen en render.com



(New Web service, source from image (dockerhub), elegir servidor más cerca/Frankfurt en este caso)

Run el servicio



The screenshot shows a Google Chrome browser window with the address bar displaying `docker-prueba-1-0-2.onrender.com`. The page content is a JSON array of five user objects, each with fields for name, surname, email, key, date of birth, and a photo file name. The browser's developer tools are open, showing the 'Raw' view of the JSON data.

```
[
  {
    "name": "Maria",
    "surname": "Bezzos",
    "email": "john.bezzos@pear.com",
    "key": "booksforever",
    "dateBirth": "2002-05-12",
    "foto": "john.bezzos.png"
  },
  {
    "name": "Bill",
    "surname": "Jobs",
    "email": "bill.jobs@pear.com",
    "key": "ilikeapples",
    "dateBirth": "1999-07-02",
    "foto": "bill.jobs.png"
  },
  {
    "name": "Steve",
    "surname": "Windows",
    "email": "steve.windows@pear.com",
    "key": "ilikelinux",
    "dateBirth": "1990-04-25",
    "foto": "steve.windows.png"
  },
  {
    "name": "Barbara",
    "surname": "Kingsolver",
    "email": "barbara.kingsolver@pear.com",
    "key": "blablaba",
    "dateBirth": "2004-01-03",
    "foto": "barbara.kingsolver.png"
  },
  {
    "name": "Maria",
    "surname": "Blanchard",
    "email": "maria.blanchard@pear.com",
    "key": "diamondsareforever",
    "dateBirth": "2001-08-22",
    "foto": "maria.blanchard.png"
  }
]
```

(<https://docker-prueba-1-0-2.onrender.com/>)