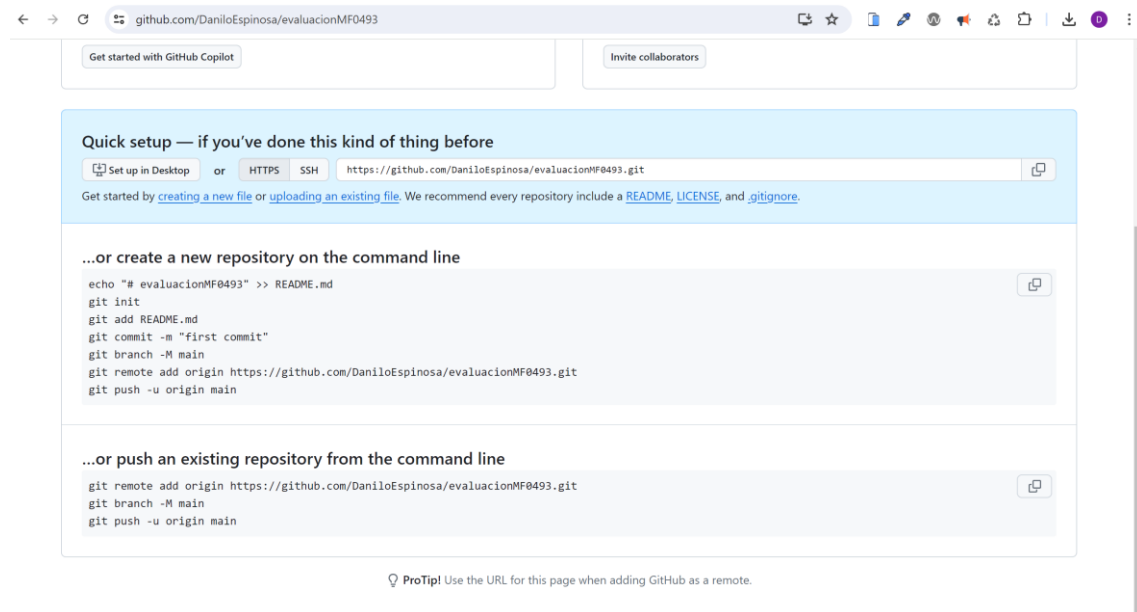


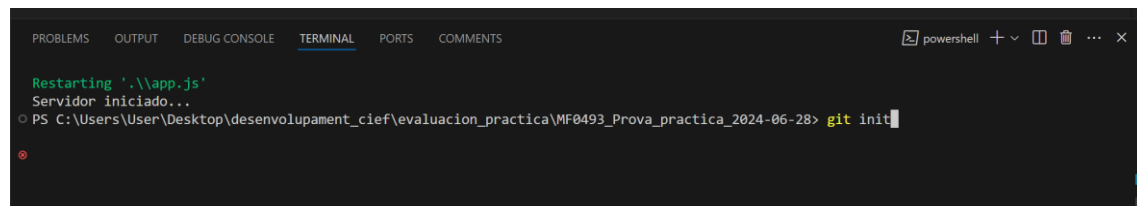
Parte 1: GITHUB

Creo un nuevo repositorio en github

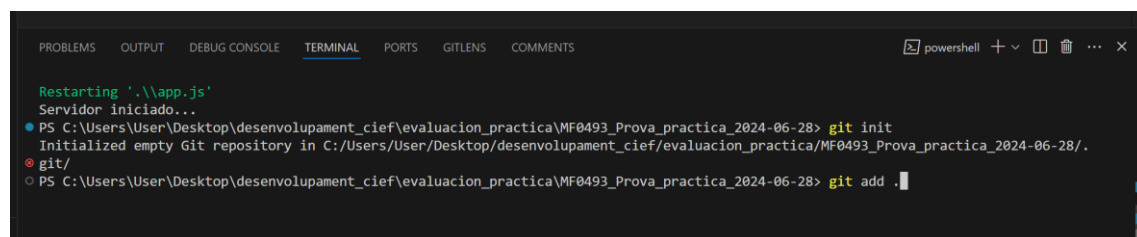
url: <https://github.com/DaniloEspinosa/evaluacionMF0493>



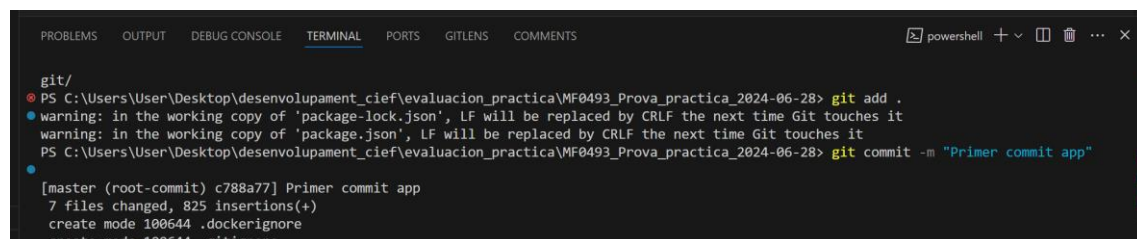
Inicio git con el comando **git init**



Una vez completa la app con el .gitignore y todos los archivos y ficheros necesarios realizo el **git add .** para agregar todo lo que voy a incluir en el commit.



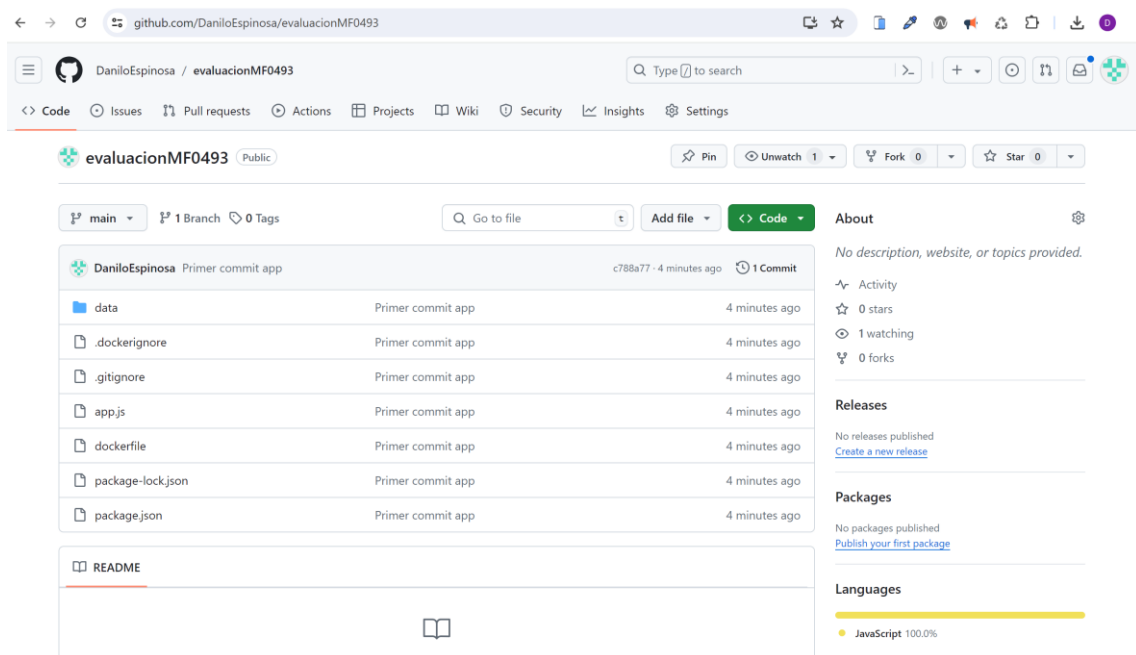
Una vez que los archivos han sido añadidos realizamos el commit con el comando **git commit -m "mensaje a incluir en el commit"**, donde dejamos listo todo para ser cargado en nuestro repositorio



git push -u origin main realizamos el push enviando los archivos y ficheros al repositorio en la rama main.

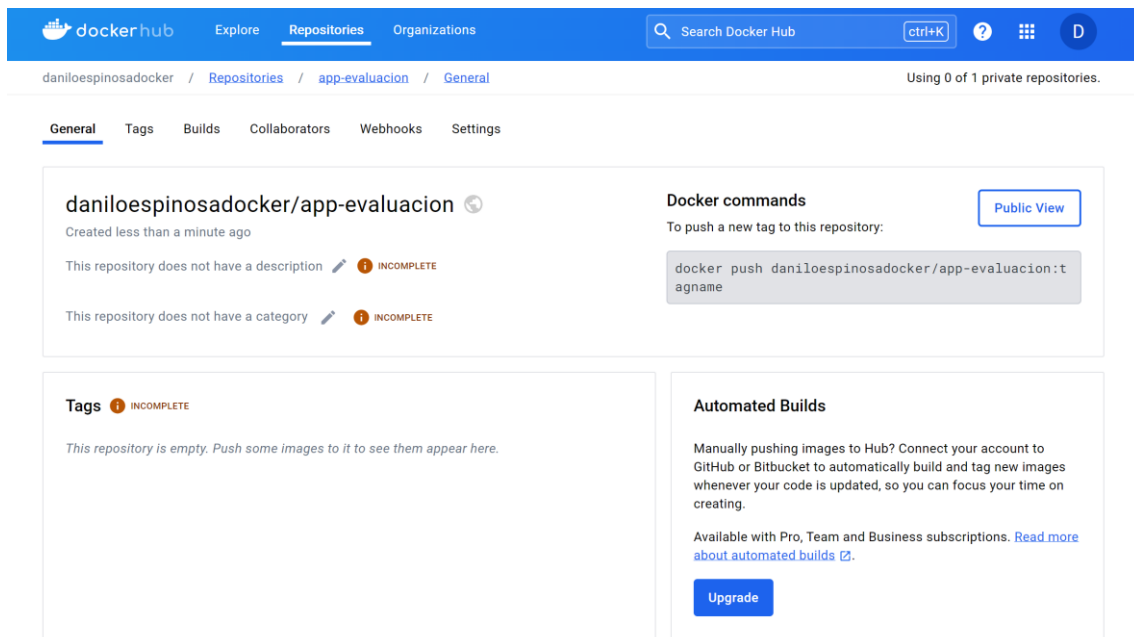
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS
create mode 100644 .gitignore
create mode 100644 app.js
create mode 100644 data/customers.json
create mode 100644 dockerfile
create mode 100644 package-lock.json
create mode 100644 package.json
PS C:\Users\User\Desktop\desenvolvament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28> git remote add origin https://git
hub.com/DaniloEspinosa/evaluacionMF0493.git
>> git branch -M main
>> git push -u origin main
```

url: <https://github.com/DaniloEspinosa/evaluacionMF0493>

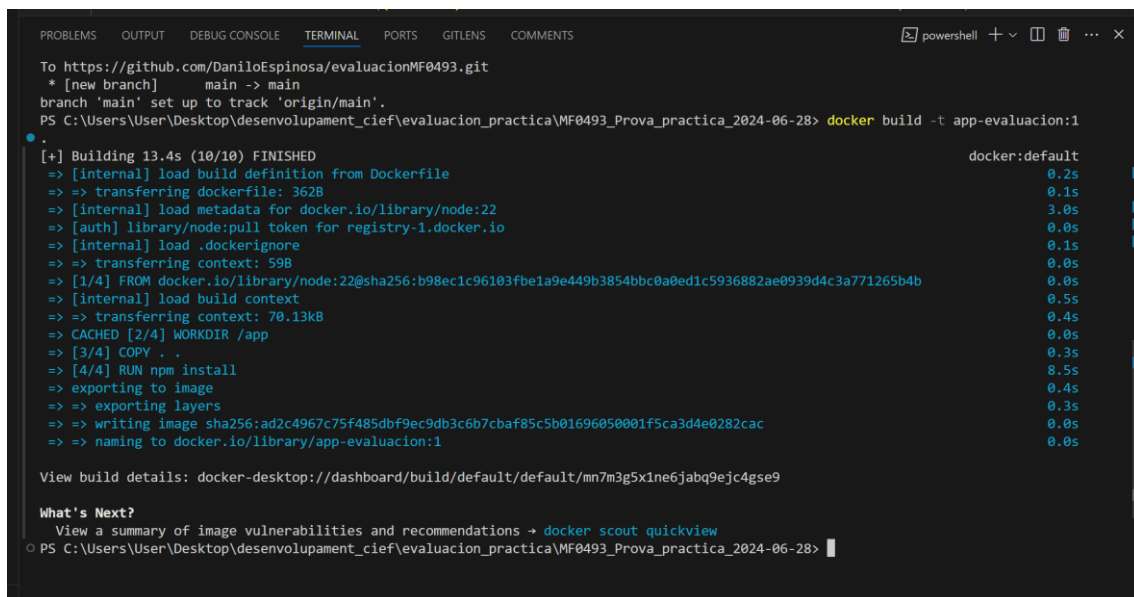


Una vez preparado el Dockerfile con todas las configuraciones necesarias y con su respectivo .dockerignore para no enviar al repositorio las cosas innecesarias iniciamos:

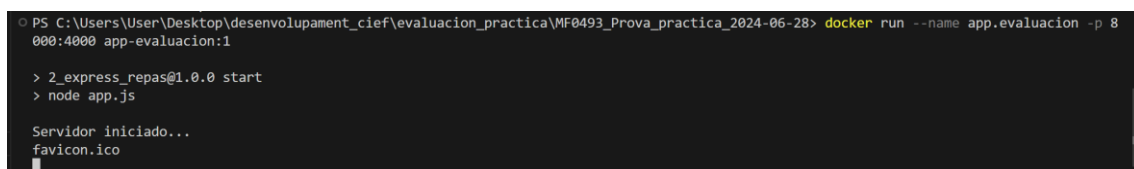
Como primer paso creare un repositorio en Docker hub



Luego abro el Docker desktop y en la consola con el comando **docker build -t nombre-imagen:tag** creo la imagen



Una vez creada la imagen creare el contenedor con la instrucción **docker run -- name nombre-contenedor -p puerto-docker:puerto-app nombre-de-la-app:tag**, esta instrucción también arranca la aplicación en el puerto seleccionado de Docker (el 8000 en este caso)



Ahora detenemos la aplicación para indicar el login a nuestra cuenta de Docker con el comando **docker login -u "usuario-docker" -p "password-de-la-cuenta" docker.io**

```
npm error command sh -c node app.js
npm error A complete log of this run can be found in: /root/.npm/_logs/2024-06-28T07_46_40_613Z-debug-0.log
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28> docker login -u "daniloespinosadocker" -p "Yt.HSub48%F&wVL" docker.io
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28>
```

Ahora indico que nuestra aplicación se va a destinar al repositorio de Docker con la instrucción **docker tag nombre-app:tag usuario-docker/nombre-app:tag**

```
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28> docker tag app-evaluation:1 daniloespinosadocker/app-evaluation:1
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28>
```

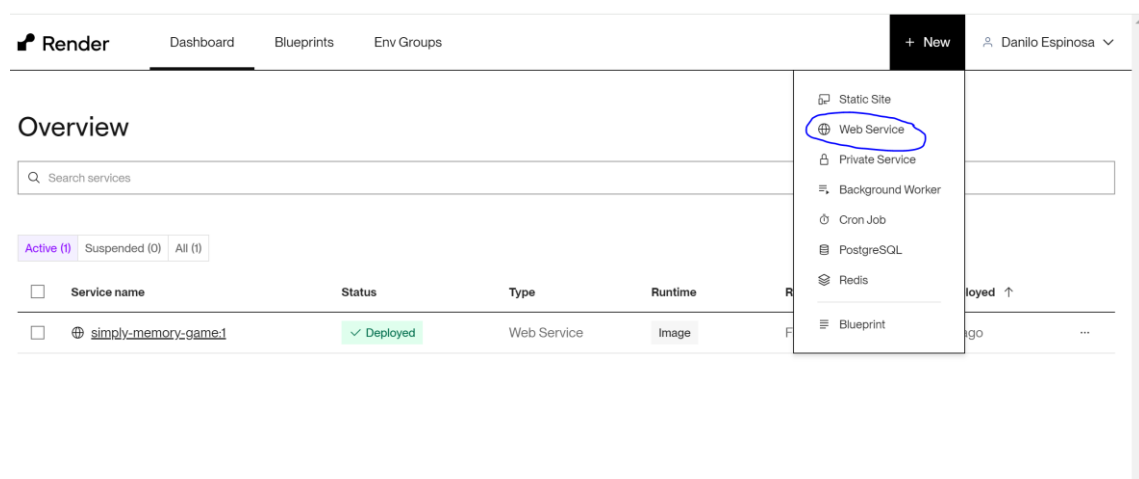
Ahora si esta todo listo para realizar el push hacia el repositorio de Docker hub

docker push usuario-docker/nombre-app:tag

```
daniloespinosadocker/app-evaluation:1
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28> docker push daniloespinosadocker/app-evaluation:1
The push refers to repository [docker.io/daniloespinosadocker/app-evaluation]
d12abb4025bc: Pushed
78f5cc6a2d5c: Pushed
f3c979c78bb3: Mounted from daniloespinosadocker/simply-memory-game
9694718de140: Mounted from daniloespinosadocker/simply-memory-game
7dad415bef86: Mounted from daniloespinosadocker/simply-memory-game
fe24499e17e4: Mounted from daniloespinosadocker/simply-memory-game
2543dc6136c8: Mounted from daniloespinosadocker/simply-memory-game
3999ea91fb6e: Mounted from daniloespinosadocker/simply-memory-game
43df359389fd: Mounted from daniloespinosadocker/simply-memory-game
d3e8d42f967c: Mounted from daniloespinosadocker/simply-memory-game
5d64de483bf5: Mounted from daniloespinosadocker/simply-memory-game
1: digest: sha256:e545969ee9d0478f23e099ea65df931989d25028395e3b07f90a040b54334a5f size: 2630
PS C:\Users\User\Desktop\desenvolupament_cief\evaluacion_practica\MF0493_Prova_practica_2024-06-28>
```

Parte 3: DESPLIEGUE EN RENDER.COM

Una vez que tenemos nuestro repositorio de Docker hub ahora a subirlo a la nube con render



Cuando indicamos nuevo web service se abre esta pantalla, donde escogemos la opción desde una imagen existente, y escribiremos la url de la imagen con su tag correspondiente

The screenshot shows the 'Configure' page in the Render dashboard. At the top, there are navigation links: 'Render', 'Dashboard', 'Blueprints', and 'Env Groups'. On the right, there are buttons for '+ New' and a user profile 'Danilo Espinosa'. The main section is titled 'Configure' and has a 'Source Code' tab selected. Under 'Source Code', there are three options: 'Git Provider', 'Public Git Repository', and 'Existing Image' (which is highlighted). Below these, there is a text input field for 'Image URL' with the placeholder 'Deploy an image from a docker registry'. The URL 'docker.io/daniloespinosadocker/app-evaluacion:1' is entered. There is also a 'Credential (Optional)' dropdown menu set to 'No credential'. A 'Connect' button is at the bottom right.

Al hacer click en connect nos darán estas opciones, el nombre podemos indicar el que sea, en este caso lo dejo como esta, la región elegimos Frankfurt ya que es la mas conveniente según nuestra ubicación, y escogemos el tipo FREE para no pagar por una suscripcion

This screenshot shows the configuration page with the 'Connect' button clicked. The 'Source Code' field now shows the Docker image 'docker.io/daniloespinosadocker/app-evaluacion:1'. The 'Name' field is 'app-evaluacion:1'. The 'Region' is set to 'Frankfurt (EU Central)'. The 'Instance Type' section shows two options: 'For hobby projects' (Free, \$0/month, 512 MB RAM, 0.1 CPU) and 'For professional use'. The 'Free' option is selected and highlighted with a purple border. An 'Upgrade to enable more features' warning is visible next to the free option.

Una vez esta todo configurado y listo desplegamos el servicio web dando click en el botón deploy web service, esto cambiara la pantalla y trabajara por unos minutos hasta que la app esta completamente desplegada en la nube

This screenshot shows the 'Advanced' section of the configuration page. It has a 'Deploy Web Service' button highlighted with a blue circle. Above it, there is an 'Environment Variables' section with a table for adding variables. The table has columns for 'NAME_OF_VARIABLE' and 'value'. There are buttons for '+ Add Environment Variable' and 'Add from .env'. A 'Generate' button is also present.

Una vez que esta todo listo podemos acceder al sitio siguiendo el enlace que nos deja el servicio, también tenemos el indicador de que esta todo funcionando en verde donde dice **live**

The screenshot shows the OnRender dashboard for a web service named "app-evaluacion:1". At the top, there's a header with "WEB SERVICE" and "app-evaluacion:1". Below this, there's a link to "https://app-evaluacion-1.onrender.com" which is circled in blue and labeled "URL" in blue handwriting. The dashboard includes a sidebar with navigation options: Events, Logs, Disks, Environment, Shell, Previews, Jobs, Metrics, Scaling, and Settings. The main content area shows a status message: "Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more." Below this, there's a deployment entry for "June 28, 2024 at 11:03 AM" with a green "Live" status indicator circled in blue and labeled "OK" in blue handwriting. At the bottom, there's a "Live tail" section showing logs, with the first line being "Jun 28 11:03:25 AM ==> Starting service..."

Con todo esto ya tenemos funcionando nuestra app

url: <https://app-evaluacion-1.onrender.com>