José De León
1170419

Virtualización
Parcial 3

Inciso 8

Generar Back

Generamos un archivo .js con su archivo package.json

En este caso api.js y package.json

Api.js

Con este generamos un endPoint en el cual obtenedremos
La hora actual
El nombre del creador (yo)

Y lo solicitado el Hello World!
Así mismo para poder hacer uso del endPoint habilitamos los Cors

```
const express = require('express');
const app = express();


app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
  next();
});

// Ruta para obtener la hora y el nombre
app.get('/info', (req, res) => {
  const nombre = "José De León";
  const hora = new Date().toLocaleTimeString();
  const data = {
    nombre: nombre,
    hora: hora,
    hello: "Hello-World!"
  };
  res.json(data);
});

// Puerto en el que escucha el servidor
const PORT = 3000;

// Iniciar el servidor
app.listen(PORT, () => {
  console.log(`Servidor API escuchando en el puerto ${PORT}`);
});
```

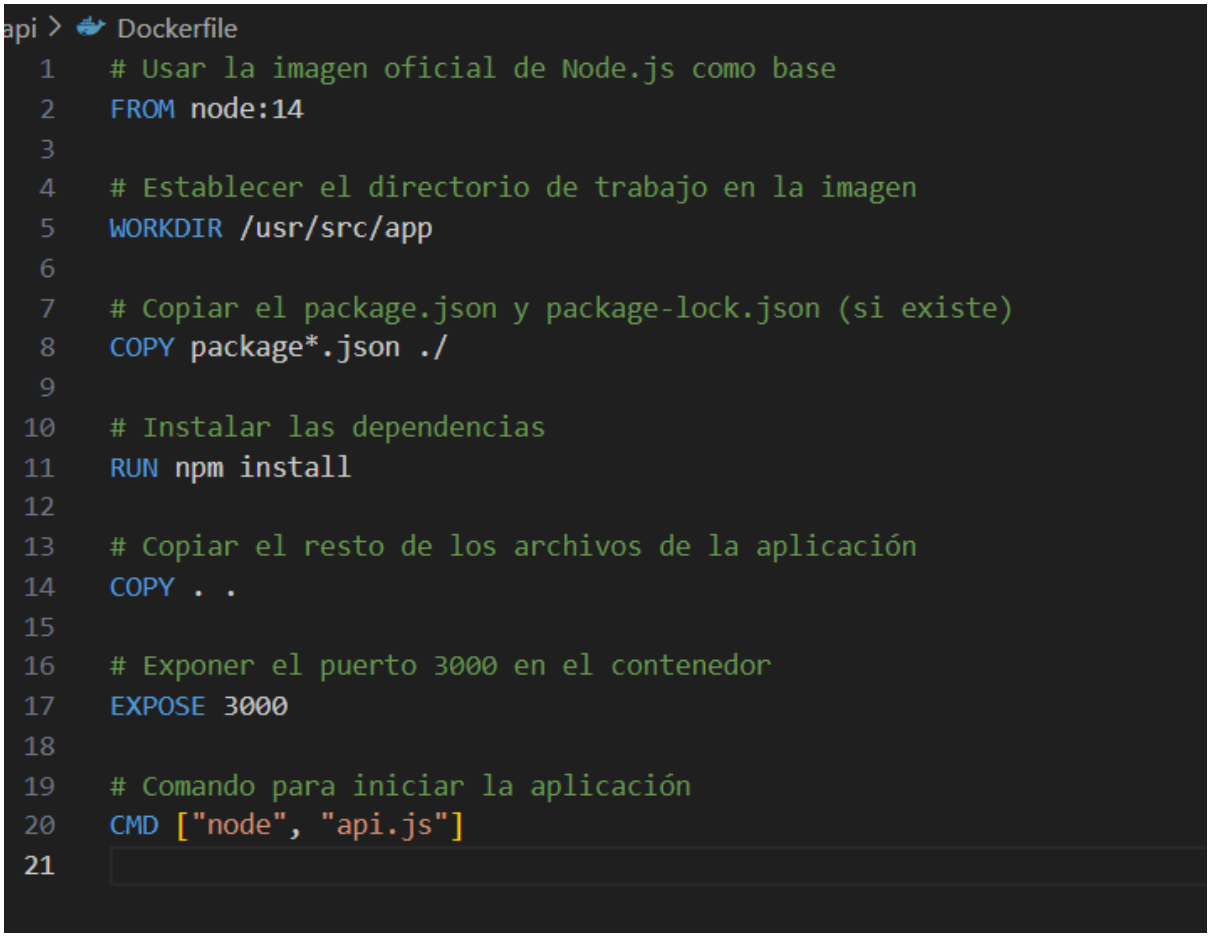
José De León
1170419
Package.json

Virtualización
Parcial 3

Este para que docker pueda hacer uso de sus dependencias

Ahora generamos un archivo dockerfile para dockerizar el archivo

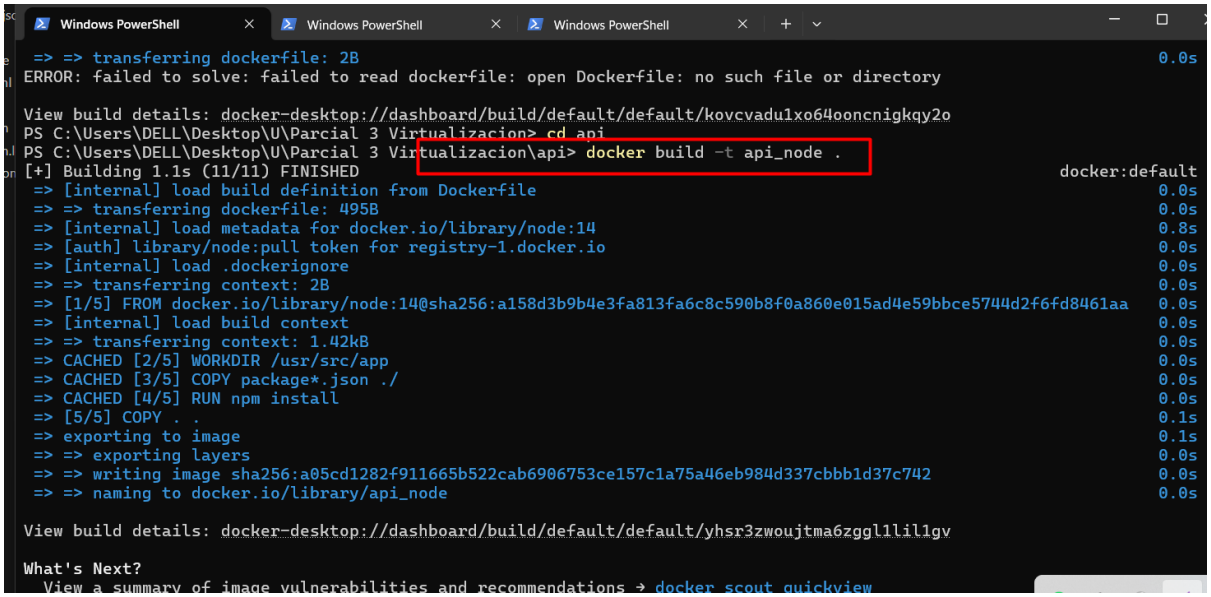
La imagen node:14 y el puerto 3000



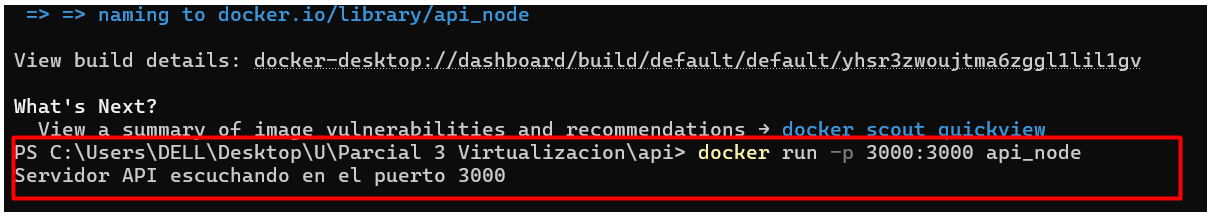
Y corremos con los siguientes comandos, con powershell

José De León
1170419

Virtualización
Parcial 3



Y el siguiente para indicar el puerto



José De León
1170419

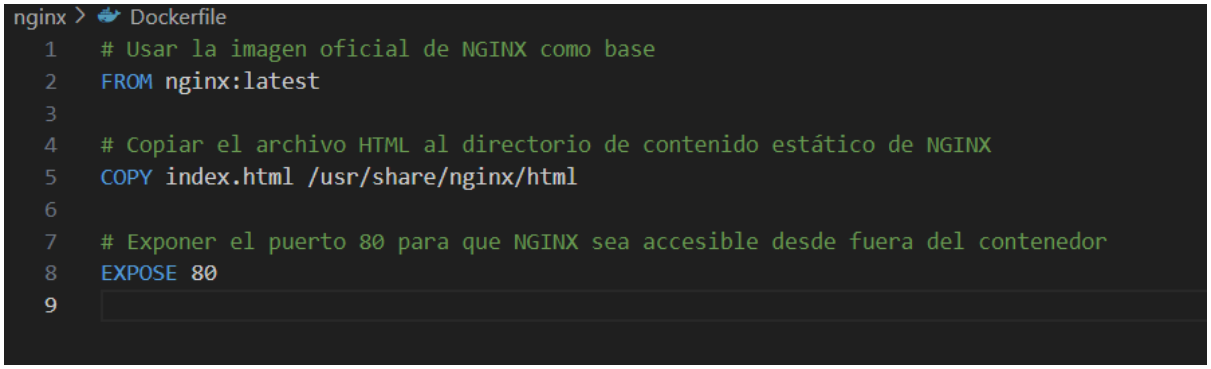
Virtualización
Parcial 3

Generar Front

Generamos un archivo HTML, con su apartado de JavaScript, para poder hacer uso del backend (en este caso ya vemos su llamado a la dirección, puerto y ruta).

```
nginx > <> index.html > ...
<!-- index.html -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Consumo de Endpoint</title>
</head>
<body>
    <h1>Consumo de Endpoint</h1>
    <button id="fetchData">Obtener Datos</button>
    <p id="data"></p>

    <script>
        document.getElementById('fetchData').addEventListener('click', async () => {
            const response = await fetch('http://localhost:3000/info');
            const data = await response.json();
            document.getElementById('data').innerText = `Nombre: ${data.nombre}, Hora: ${data.hora}, HolaMUND!!!: ${data.hello} `;
        });
    </script>
</body>
</html>

```

Para poder dockerizarlo hacemos uso de este archivo “dockerfile”

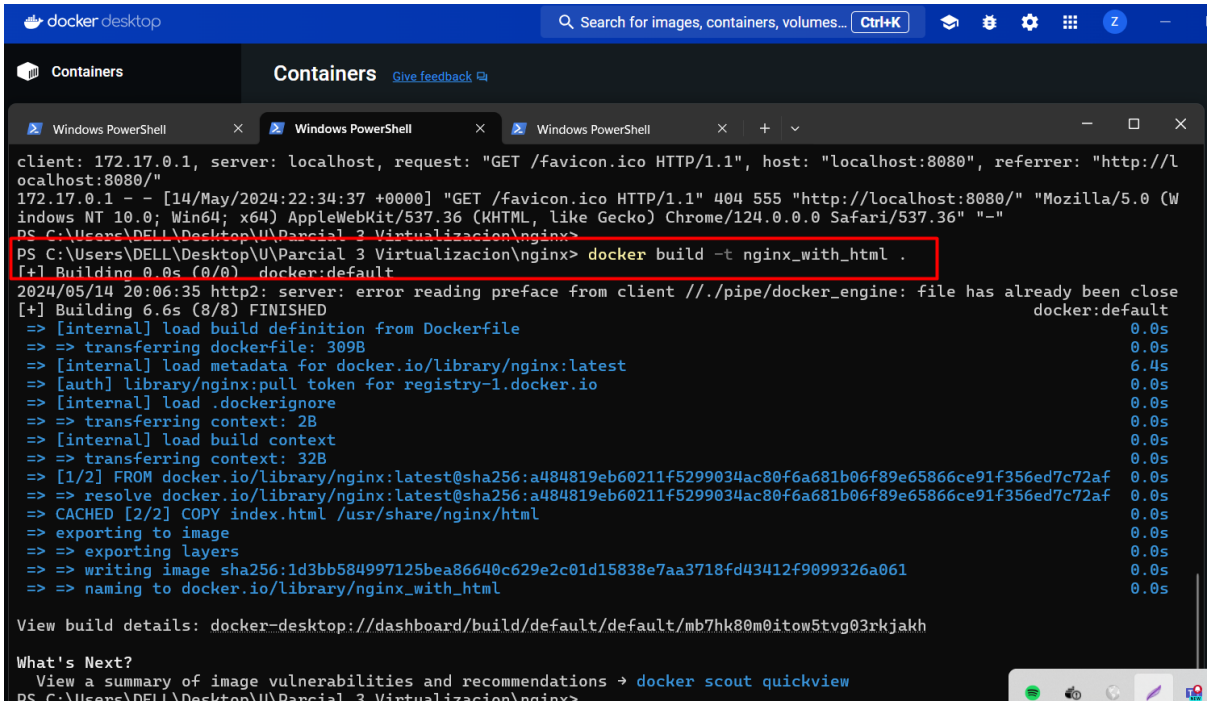


```
nginx > Dockerfile
# Usar la imagen oficial de NGINX como base
FROM nginx:latest

# Copiar el archivo HTML al directorio de contenido estático de NGINX
COPY index.html /usr/share/nginx/html

# Exponer el puerto 80 para que NGINX sea accesible desde fuera del contenedor
EXPOSE 80

```

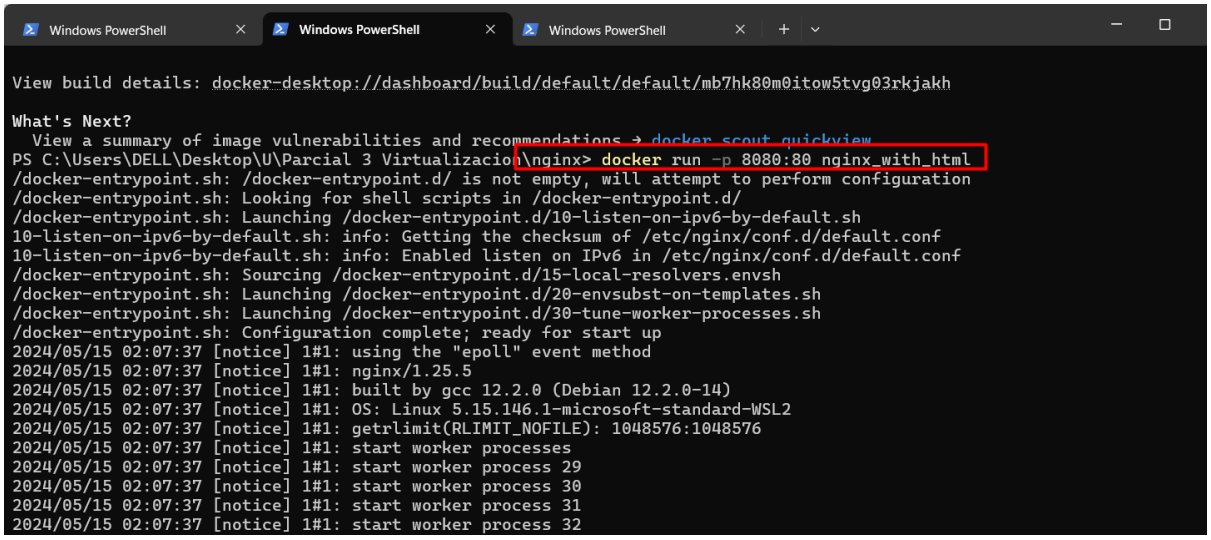
José De León
1170419

Virtualización
Parcial 3

Para poder correrlo, uso powershell para facilidad, con estos comandos,



Y luego este

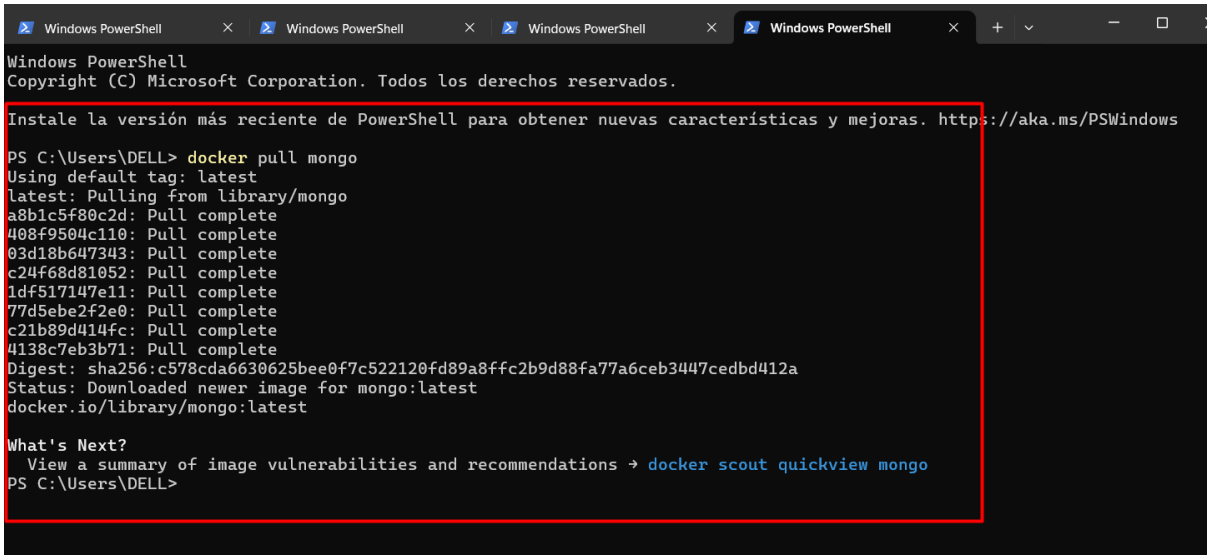


Para indicarle puerto.

José De León
1170419
Base de datos

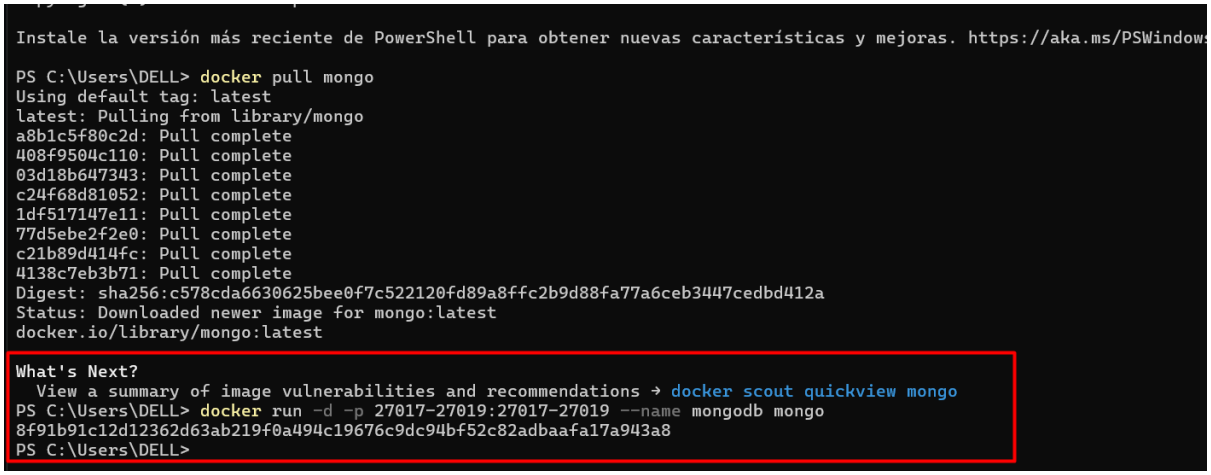
Virtualización
Parcial 3

Para esta al ser no relacional usaremos mongo db con el siguiente comando



Al tener la imagen, hacemos uso de este para poder levantar la instancia de mongoDB

Este se crea de esta forma ya para levantar el servicio y uno poder hacerle las modificaciones

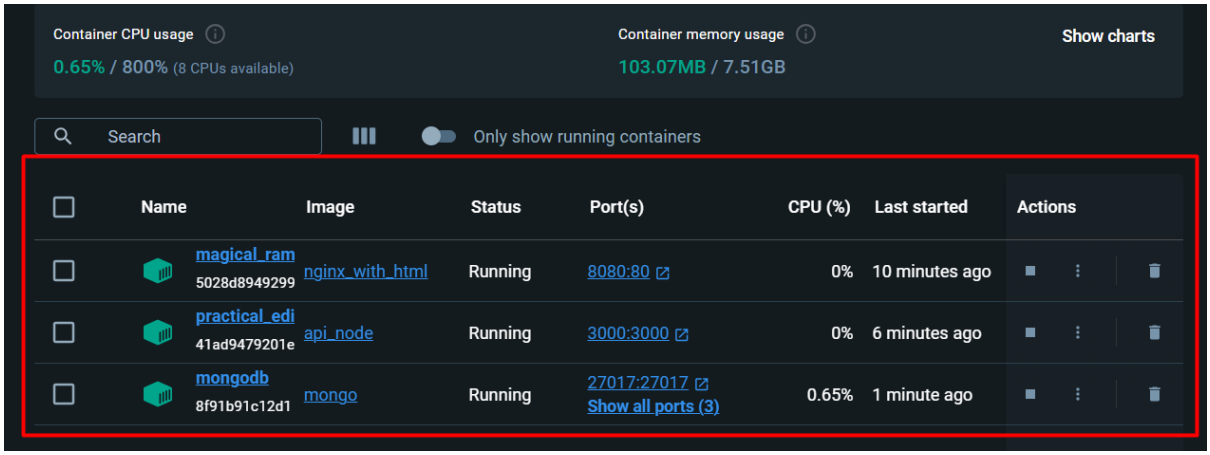


José De León
1170419

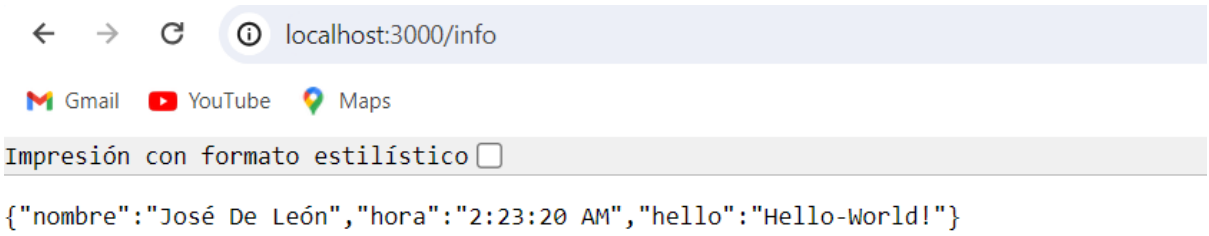
Virtualización
Parcial 3

Anexos

Ejecución de contenedores.



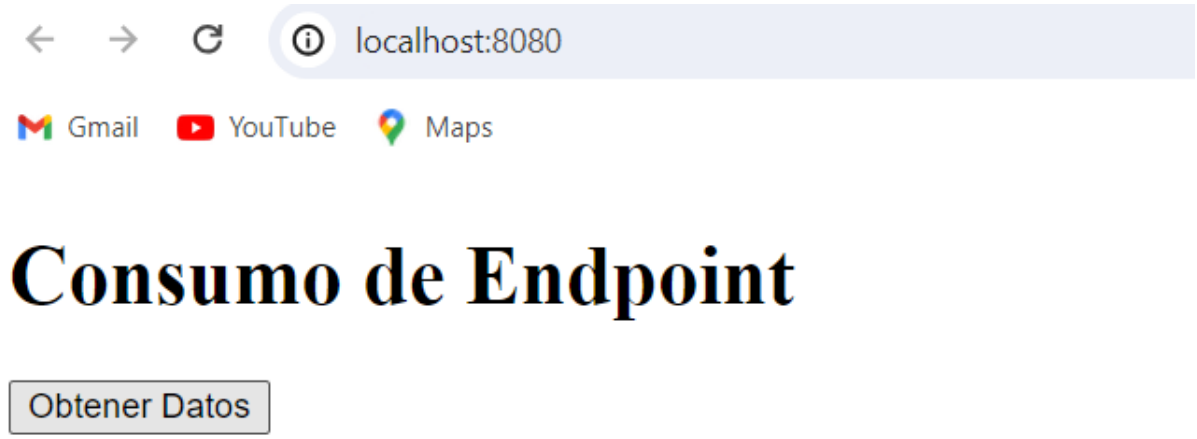
Ejecución del API (por alguna razon la hora no la da bien)



José De León
1170419

Virtualización
Parcial 3

Front antes de consumo



Front despues de consumo