Eduardo Leiver Vargas Mencia

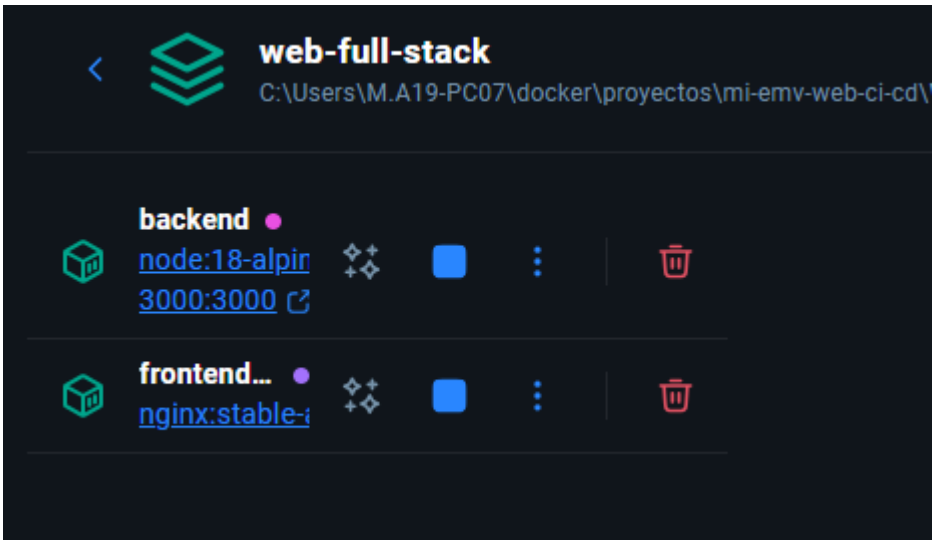
A - UT 2.6 - Taller Docker Frontend-Backend

Clonar el repositorio:
con git clone, dentro de mi carpeta mi-emv-web-ci-cd

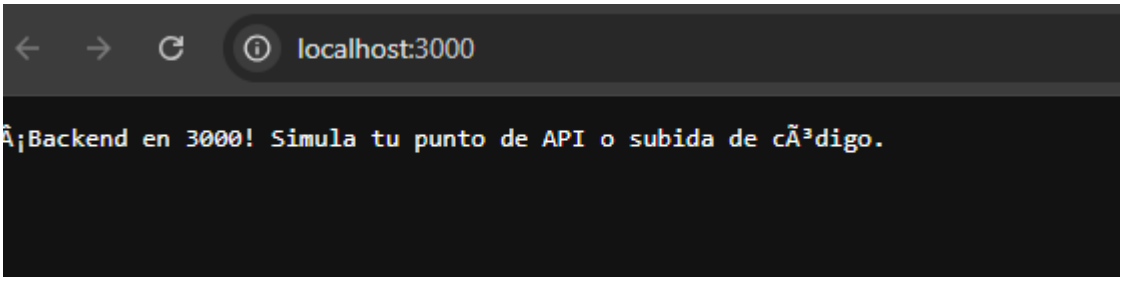
```
C:\Users\M.A19-PC07\docker\proyectos\mi-emv-web-ci-cd>git clone https://github.com/jrmcero07/Web-Full-Stack.git
Cloning into 'Web-Full-Stack'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 18 (delta 3), reused 15 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (18/18), 6.02 KiB | 2.01 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

PASOS PARA REALIZAR LA ACTIVIDAD EN EL AULA
1.Levantar el entorno , donde se inicia los servicios de frontend (Nginx) y backend (Node.js), se levanta el entorno dentro de la carpeta mi-emv-web-ci-cd con el comando : docker compose up -d

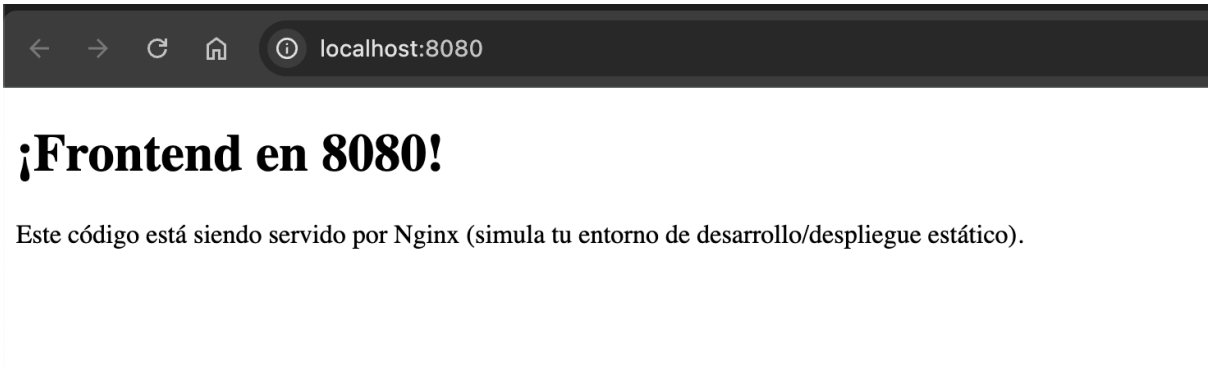
2.Verificar Endpoints
después de aplicar el compose :



resultado del backend



resultado del frontend



3.Modificación del backend/server.js .

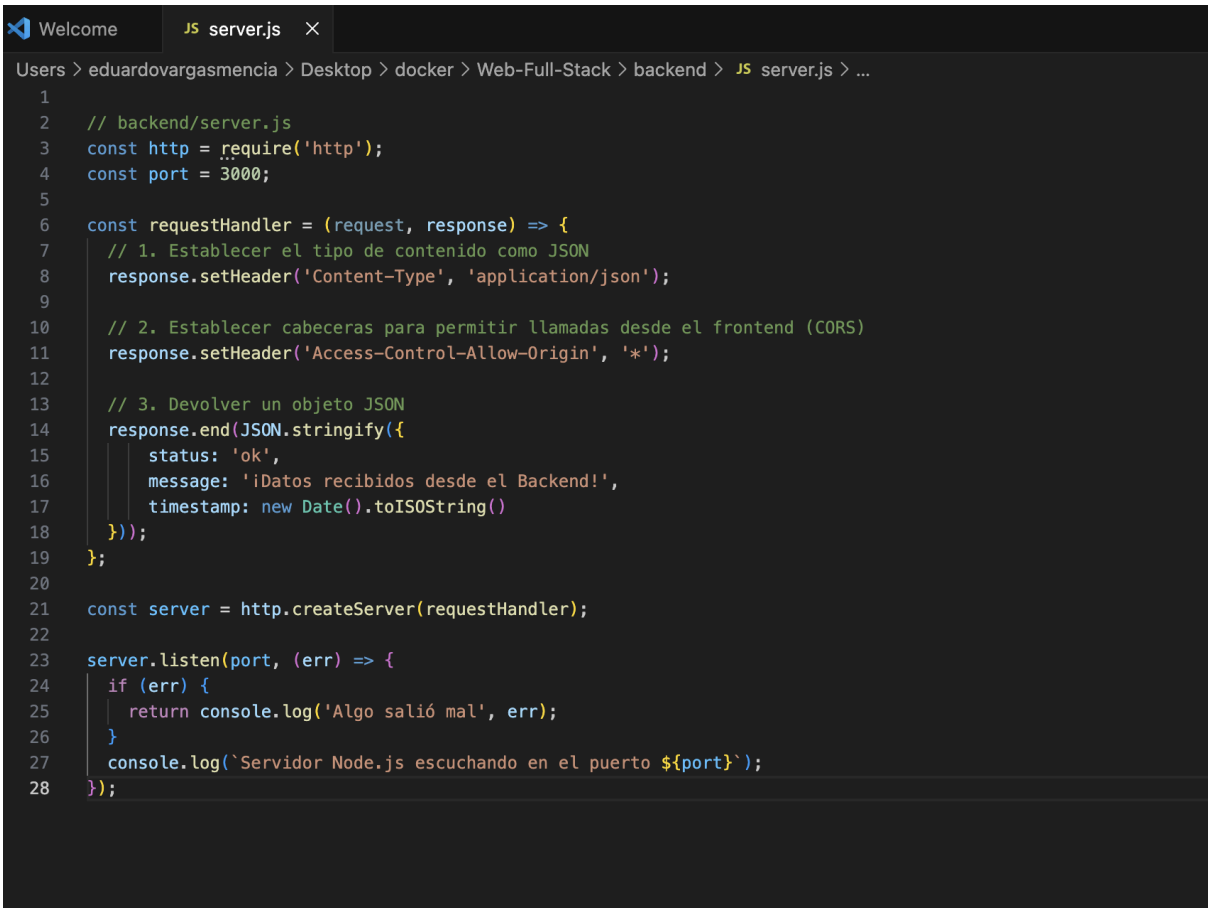
Para ello hay que acceder a la carpeta donde hayamos hecho el git clone, hay una carpeta llamada 'BACKEND', dentro de esa carpeta está el archivo backend/server.js

Hay que modificar su contenido para que sirva datos en formato JSON, y eso lo hacemos mediante el contenido que ha proporcionado el profesor, este ;

Eduardo Leiver Vargas Mencia

```
// backend/server.js
const http = require('http');
const port = 3000;

const requestHandler = (request, response) => {
  // 1. Establecer el tipo de contenido como JSON
  response.setHeader('Content-Type', 'application/json');

  // 2. Establecer cabeceras para permitir llamadas desde el frontend (CORS)
  response.setHeader('Access-Control-Allow-Origin', '*');

  // 3. Devolver un objeto JSON
  response.end(JSON.stringify({
      status: 'ok',
      message: 'iDatos recibidos desde el Backend!',
      timestamp: new Date().toISOString()
  }));
};

const server = http.createServer(requestHandler);

server.listen(port, (err) => {
  if (err) {
    return console.log('Algo salió mal', err);
  }
  console.log(`Servidor Node.js escuchando en el puerto ${port}`);
});
```

Implementas los cambios y los guardas.

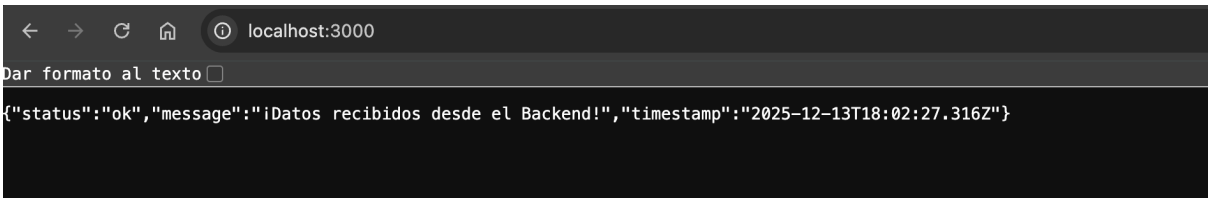


3.2 Reiniciar el contenedor de Backend, ya que hemos cambiado el código de Node.js, para ellos usaremos el siguiente comando

docker compose restart backend

```
iMac-de-EDUARDO:Web-Full-Stack eduardovargasmencia$ docker compose restart backe
nd
WARN[0000] /Users/eduardovargasmencia/desktop/docker/Web-Full-Stack/docker-compo
se.yml: the attribute `version` is obsolete, it will be ignored, please remove i
t to avoid potential confusion
[+] Restarting 1/1
 ✔ Container web_backend_ci  Started                                        2.1s
iMac-de-EDUARDO:Web-Full-Stack eduardovargasmencia$
```

3.3 Verificación del backend



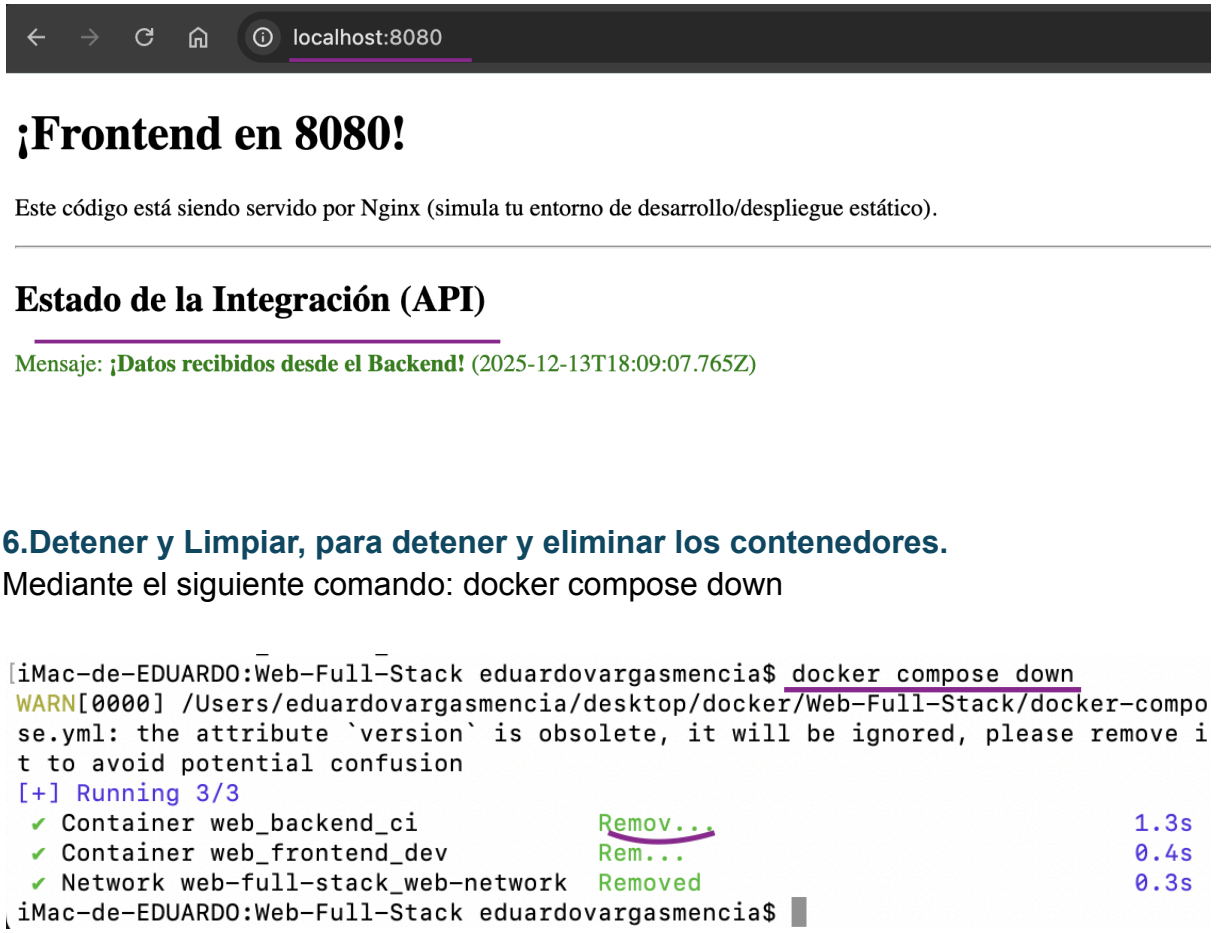
4.Modificación del Frontend, hay que añadir el bloque JavaScript

```
<hr>
<h2>Estado de la Integración (API)</h2>
<p id="api-status">Cargando datos del backend...</p>

<script>
  // Llama al servicio 'backend' a través del puerto expuesto 3000
  fetch('http://localhost:3000')
    .then(response => response.json())
    .then(data => {
      const statusElement = document.getElementById('api-status');
      statusElement.innerHTML = `Mensaje: <strong>${data.message}</strong> (${data.timestamp})`;
      statusElement.style.color = 'green';
    })
    .catch(error => {
      console.error('Error al conectar con el backend:', error);
      document.getElementById('api-status').textContent = 'Error al conectar con la API del Backend.';
      document.getElementById('api-status').style.color = 'red';
    });
</script>
```

La carpeta la cual hemos clonado para hacer la actividad, dentro de ella hay una carpeta que se llama frontend, dentro de ella hay un archivo html, abrimos el archivo por ejemplo en Visual Studio Code, para editarlo y añadimos los cambios facilitados por el profe al final del archivo, dentro de la etiqueta body , y los guardamos.

5. Prueba Final(Integración de Servicios)

A) Recargar el Frontend, vuelve a http://localhost:8080 y recarga la página



6.Detener y Limpiar, para detener y eliminar los contenedores.

Mediante el siguiente comando: docker compose down

```
iMac-de-EDUARDO:Web-Full-Stack eduardovargasmencia$ docker compose down
WARN[0000] /Users/eduardovargasmencia/desktop/docker/Web-Full-Stack/docker-compo
se.yml: the attribute `version` is obsolete, it will be ignored, please remove i
t to avoid potential confusion
[+] Running 3/3
 ✔ Container web_backend_ci            Remov...                          1.3s
 ✔ Container web_frontend_dev          Rem...                            0.4s
 ✔ Network web-full-stack_web-network  Removed                           0.3s
iMac-de-EDUARDO:Web-Full-Stack eduardovargasmencia$
```