

Práctica Microservicios y Contenedores

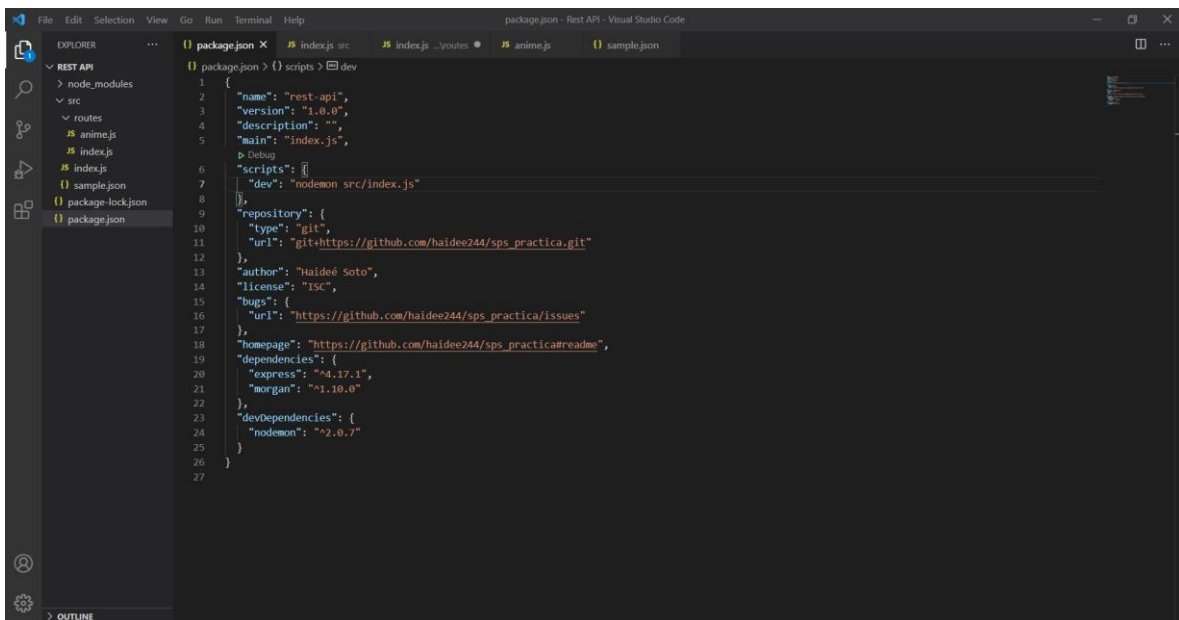
Se utilizó para el desarrollo de esta práctica el programa NodeJS, junto con Visual Studio Code, para facilitar la programación en JavaScript, siendo Java uno de los lenguajes más utilizados en la actualidad y aprovechando la incorporación de librerías tales como Morgan, Express y Nodemon.

Las pruebas dentro del proceso de desarrollo se realizaron en el programa Postman por su facilidad de uso y su acoplamiento con Visual Studio Code.

Inspirada en la práctica anterior, decidí realizar esta REST API con el mismo tema, incluyendo algunas de las películas y series de anime como un mensaje para mostrar al usuario y ser modificadas de así requerirlo.

Todos los archivos se guardaron dentro de una misma carpeta llamada Rest_API, la cual contiene tanto los archivos utilizados para la programación en JSON como los utilizados para Docker.

Como primer paso, se utilizó el comando `npm init` dentro de la terminal de Visual Studio Code, para poder inicializar las características de nuestra aplicación. Posteriormente, se importaron 3 librerías mediante la terminal de Visual Studio Code que servirían para el funcionamiento total de la api, las cuales son Morgan, Express y Nodemon, más adelante se observará su uso dentro del archivo `index.js`.

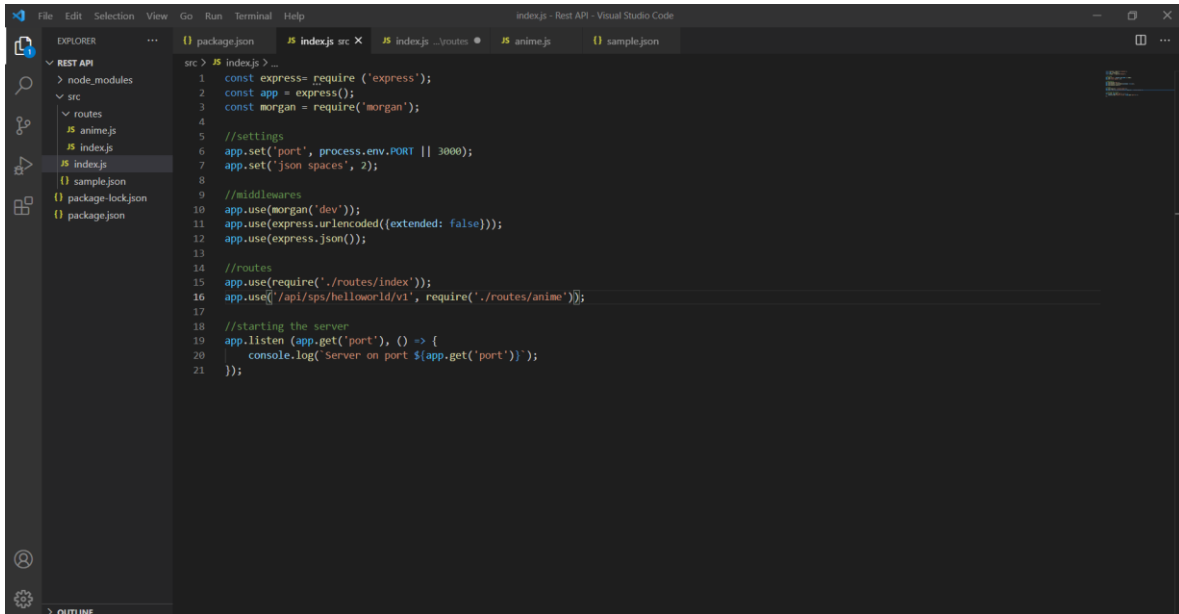
A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure with folders like 'node_modules', 'src', and 'routes', and files like 'anime.js', 'index.js', 'sample.json', 'package-lock.json', and 'package.json'. The main editor area displays the 'package.json' file with the following content:

```
1 {
2   "name": "rest-api",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "dev": "nodemon src/index.js"
8   },
9   "repository": {
10    "type": "git",
11    "url": "git+https://github.com/haidee244/sps_practica.git"
12  },
13  "author": "Haidee Soto",
14  "license": "ISC",
15  "bugs": {
16    "url": "https://github.com/haidee244/sps_practica/issues"
17  },
18  "homepage": "https://github.com/haidee244/sps_practica#readme",
19  "dependencies": {
20    "express": "^4.17.1",
21    "morgan": "^1.10.0"
22  },
23  "devDependencies": {
24    "nodemon": "^2.0.7"
25  }
26 }
```

Archivo package.json.

Posteriormente, ya definidas las dependencias a utilizar dentro del desarrollo del programa, se creó la carpeta `src` y el archivo `index.js`, dentro de este archivo se

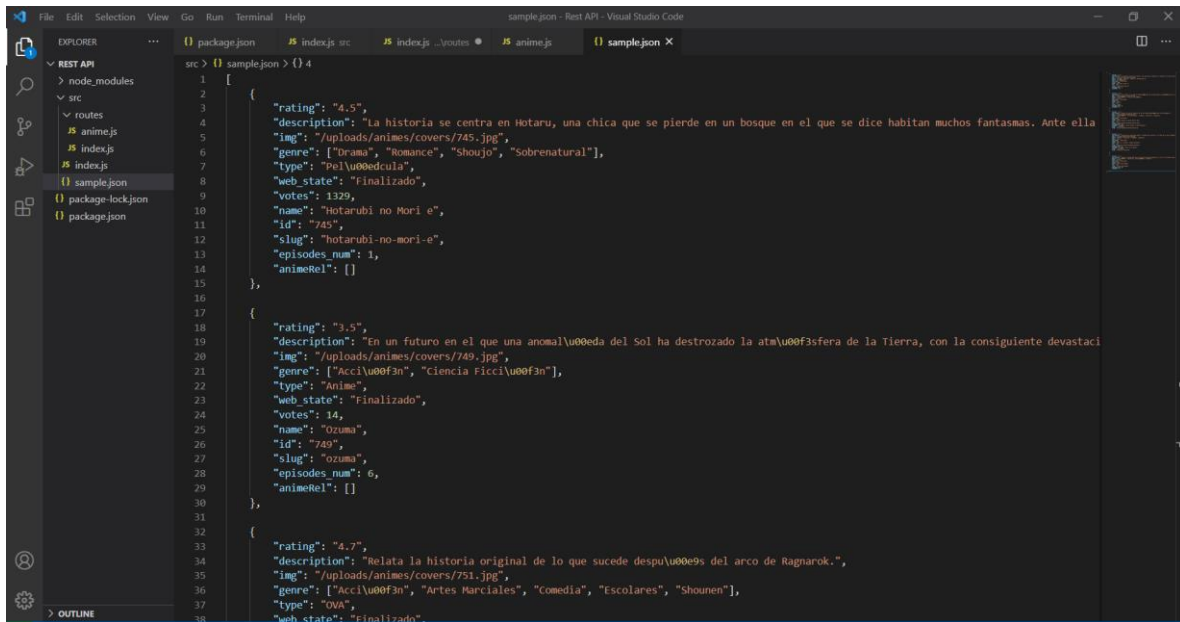
utilizan las librerías mencionadas al inicio de la práctica. Se establece que se utilizará el puerto 3000 para las pruebas o el puerto determinado por el usuario más tarde. Se programaron algunos middlewares, para mejorar la comunicación con el usuario, para lo cual se utilizaron las librerías Express y Morgan. De igual manera, se estableció como dirección principal la dirección solicitada en el tercer punto de la práctica, con distintos endpoints para realizar las pruebas sin modificar dicho endpoint al principio.

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure for a REST API with folders like node_modules, src, routes, anime.js, index.js, sample.json, package-lock.json, and package.json. The main editor area displays the content of index.js, which is a Node.js script using Express and Morgan. The code includes imports, settings for port and JSON spaces, middleware usage, route definitions for index and anime, and a server listening function on port 3000.

```
1 const express= require ('express');
2 const app = express();
3 const morgan = require('morgan');
4
5 //settings
6 app.set('port', process.env.PORT || 3000);
7 app.set('json spaces', 2);
8
9 //middlewares
10 app.use(morgan('dev'));
11 app.use(express.urlencoded({extended: false}));
12 app.use(express.json());
13
14 //routes
15 app.use(require('./routes/index'));
16 app.use('/api/sps/helloworld/v1', require('./routes/anime'));
17
18 //starting the server
19 app.listen (app.get('port'), () => {
20   console.log('Server on port ${app.get('port')}');
21 });
```

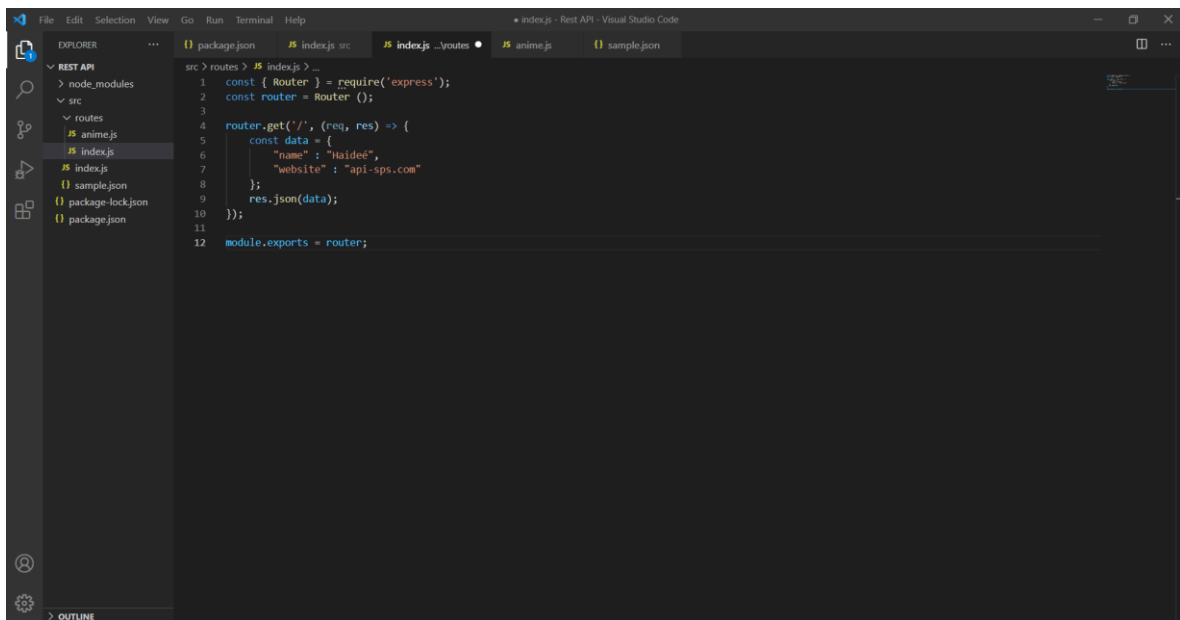
Archivo index.js.

Se creó el archivo sample.json en el cual se copiaron los 5 primeros elementos del archivo original de anime para utilizar la información dentro de la API posteriormente.



Archivo sample.json.

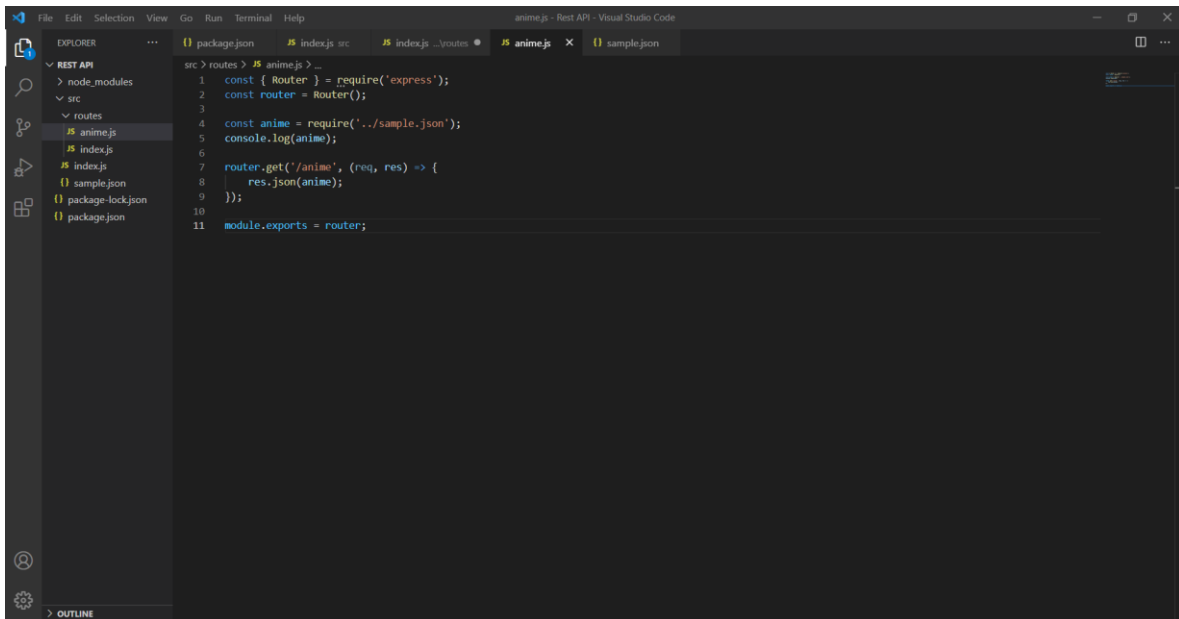
Se creó un segundo archivo `index.js`, dentro de la carpeta `routes`, para poder realizar las primeras pruebas imprimiendo mi nombre y una dirección web falsa. Después dicho archivo se utilizó para desplegar la información dentro de la API. Se decidió crear una carpeta con el propósito de poder organizar las rutas que seguiría la API de mejor manera y mantener mayor control sobre cualquier posible error.



Archivo index.js utilizado para las rutas que seguiría la página para obtener datos.

A continuación, se creó el archivo `anime.js`, utilizado para desplegar la información de las películas incluidas en el archivo `sample.json`. Dicho archivo se renombró

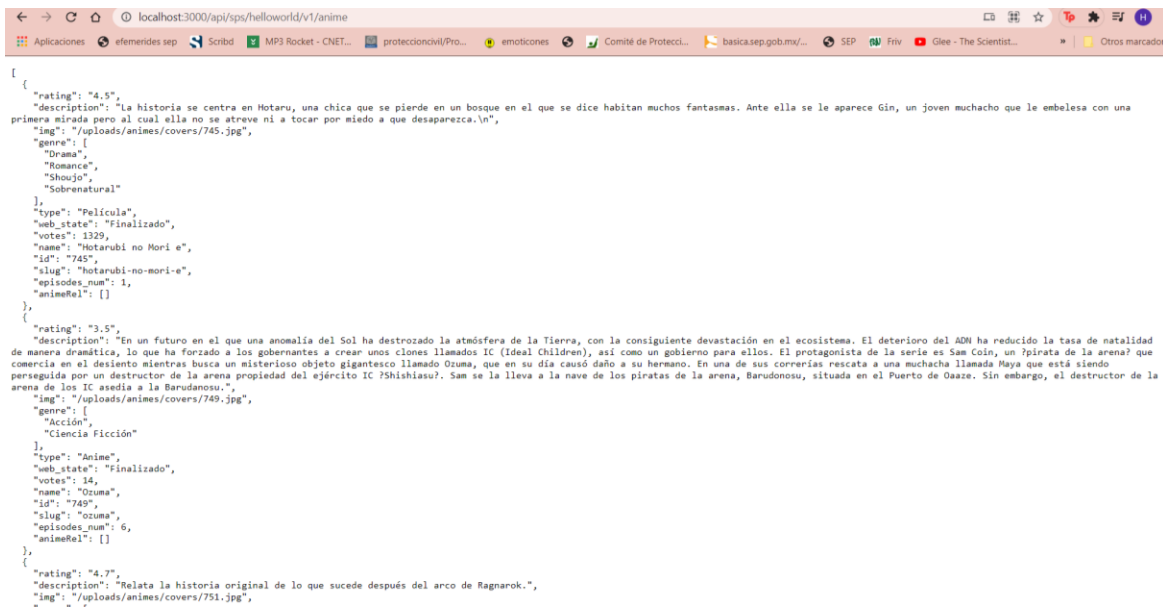
posteriormente como v1, para poder seguir de manera exitosa la ruta solicitada como endpoint para mantener la comunicación con el usuario.



```

src > routes > anime.js > ...
1  const { Router } = require('express');
2  const router = Router();
3
4  const anime = require('../sample.json');
5  console.log(anime);
6
7  router.get('/anime', (req, res) => {
8    res.json(anime);
9  });
10
11 module.exports = router;
  
```

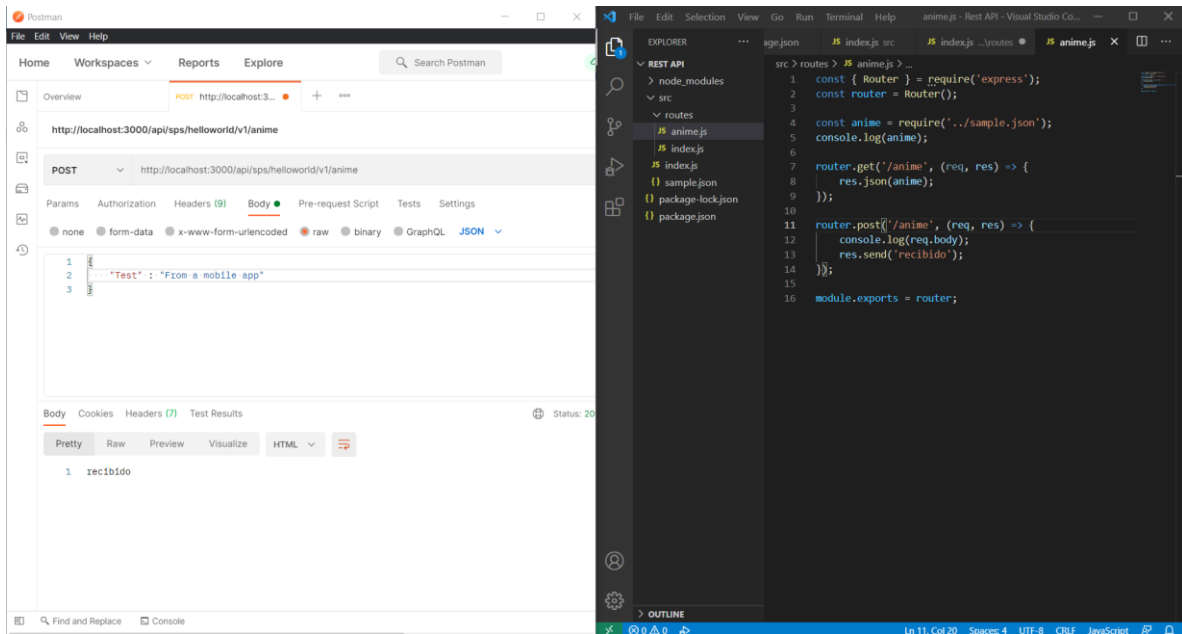
Archivo anime.js.



```

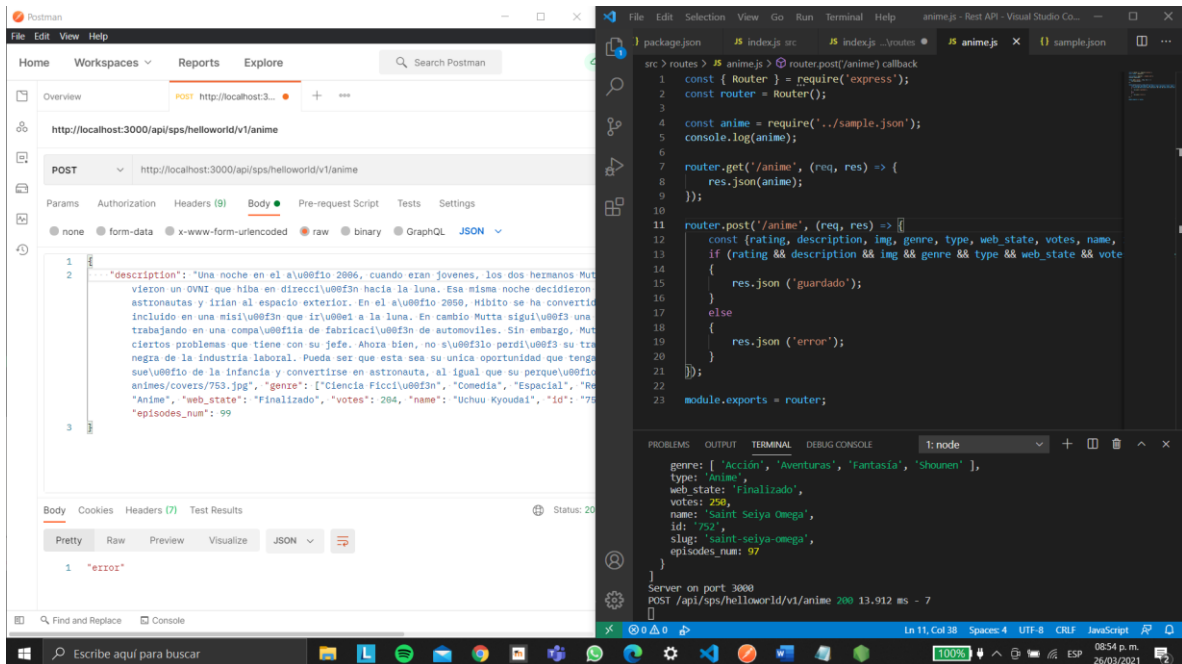
[
  {
    "rating": "4.5",
    "description": "La historia se centra en Motaru, una chica que se pierde en un bosque en el que se dice habitan muchos fantasmas. Ante ella se le aparece Gin, un joven muchacho que le enbelesa con una primera mirada pero al cual ella no se atreve ni a tocar por miedo a que desaparezca.\n",
    "img": "/uploads/animes/covers/745.jpg",
    "genre": [
      "Drama",
      "Romance",
      "Shounjo",
      "Sobrenatural"
    ],
    "type": "Película",
    "web_state": "Finalizado",
    "votes": 1329,
    "name": "Hotarubi no Mori e",
    "id": "745",
    "slug": "hotarubi-no-mori-e",
    "episodes_num": 1,
    "animeRel": []
  },
  {
    "rating": "3.5",
    "description": "En un futuro en el que una anomalía del Sol ha destruido la atmósfera de la Tierra, con la consiguiente devastación en el ecosistema. El deterioro del ADN ha reducido la tasa de natalidad de manera dramática, lo que ha forzado a los gobernantes a crear unos clones llamados IC (Ideal Children), así como un gobierno para ellos. El protagonista de la serie es Sam Coin, un ?pirata de la arena? que comercia en el desierto mientras busca un misterioso objeto gigantesco llamado Ozuma, que en su día causó daño a su hermano. En una de sus correrías rescata a una muchacha llamada Maya que está siendo perseguida por un destructor de la arena propiedad del ejército IC ?Shishiasu?. Sam se la lleva a la nave de los piratas de la arena, Barudonosu, situada en el Puerto de Oaze. Sin embargo, el destructor de la arena de los IC asedia a la Barudanosu.",
    "img": "/uploads/animes/covers/749.jpg",
    "genre": [
      "Acción",
      "Ciencia Ficción"
    ],
    "type": "Anime",
    "web_state": "Finalizado",
    "votes": 14,
    "name": "Ozuma",
    "id": "749",
    "slug": "ozuma",
    "episodes_num": 6,
    "animeRel": []
  },
  {
    "rating": "4.7",
    "description": "Relata la historia original de lo que sucede después del arco de Ragnarok.",
    "img": "/uploads/animes/covers/751.jpg",
    " "
  }
]
  
```

Al comprobar mediante el localhost que se podía recibir información, se modificó el archivo anime.json para poder enviar información y recibir un mensaje de confirmación de recepción. Esta funcionalidad se probó enviando un mensaje de tipo json por medio de la aplicación Postman.

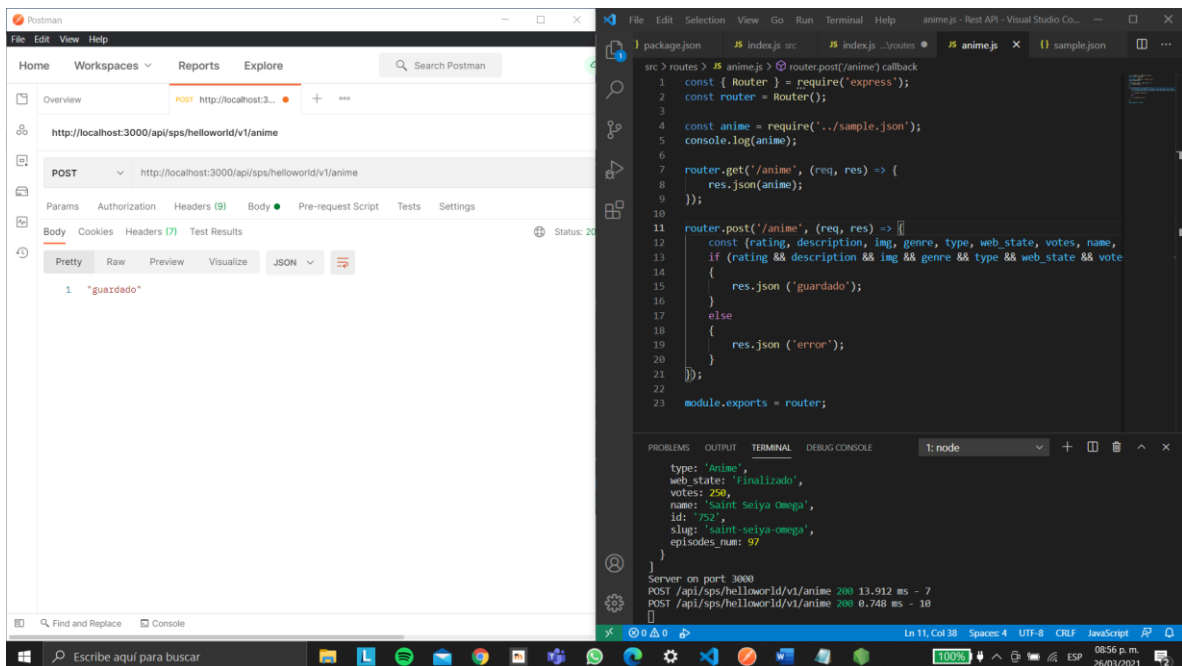


Al comprobar una recepción exitosa del mensaje enviado, la siguiente prueba fue intentar mandar una película con todos sus datos, obteniendo como resultado una recepción de igual manera exitosa, como se puede observar en la terminal de Visual Code Studio.

Posteriormente, se modificó el código del archivo anime.js para poder comprobar si se enviaba la información completa o no antes de guardarla en la base de datos. Como se puede observar en la imagen, en este caso faltaba el rating por lo que se recibe un error, al contrario, si se envía completa la información, se recibe un mensaje que indica que fue guardada.

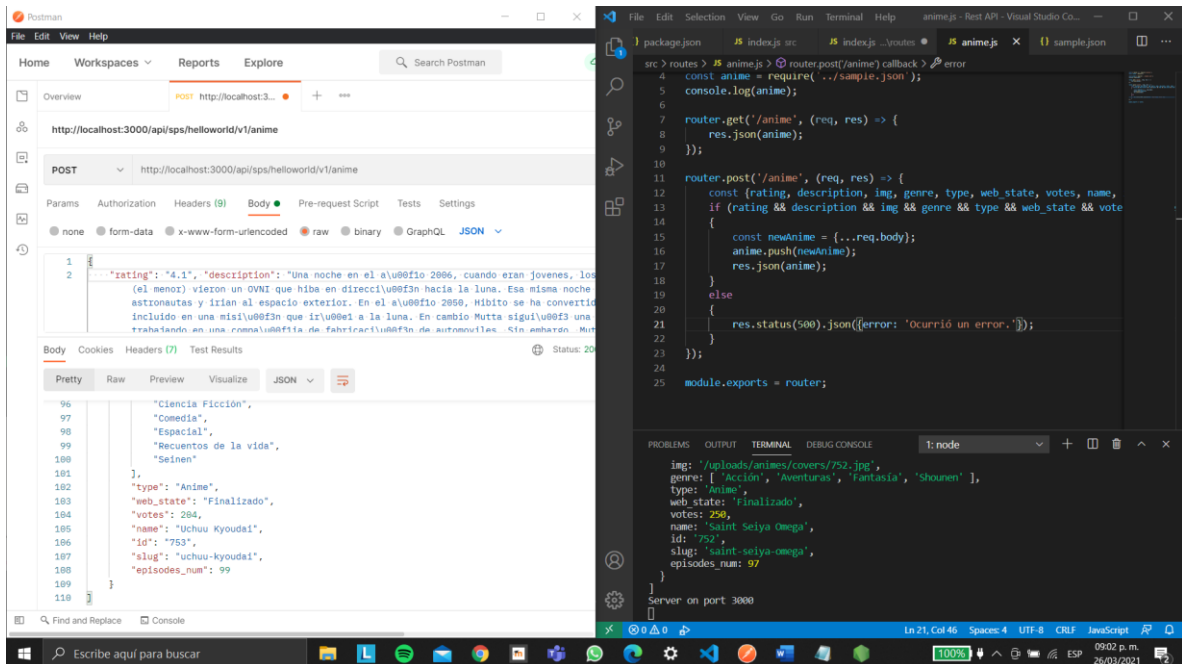


Prueba 3: Obtención de datos no exitosa.

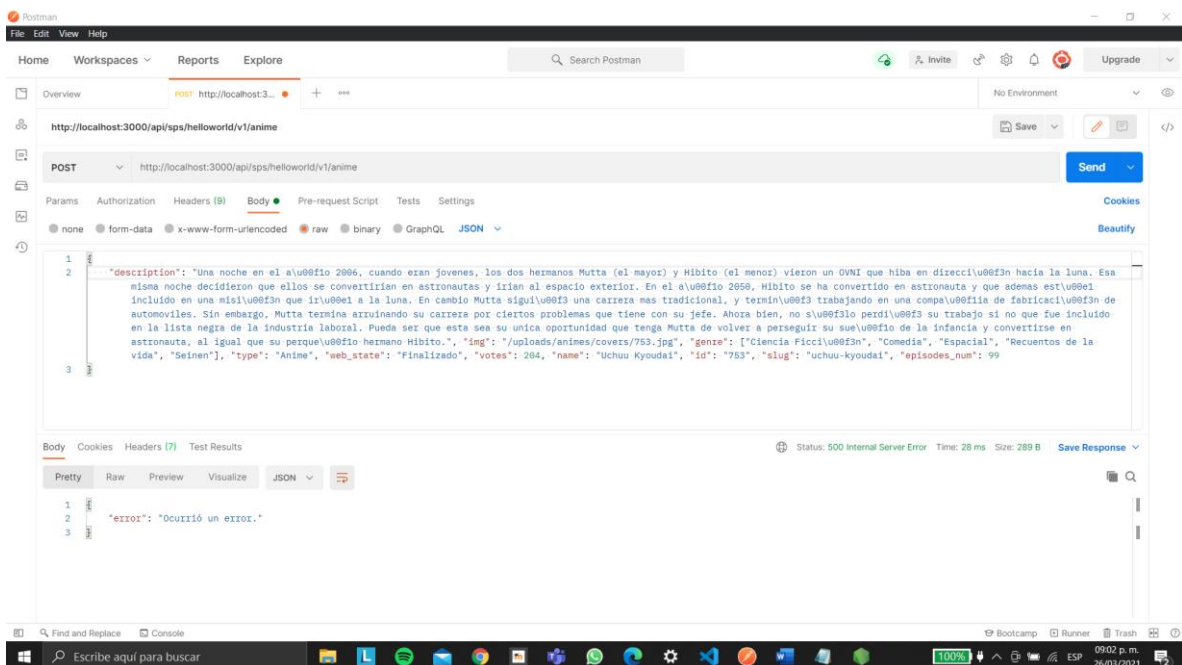


Prueba 4: Obtención de datos exitosa.

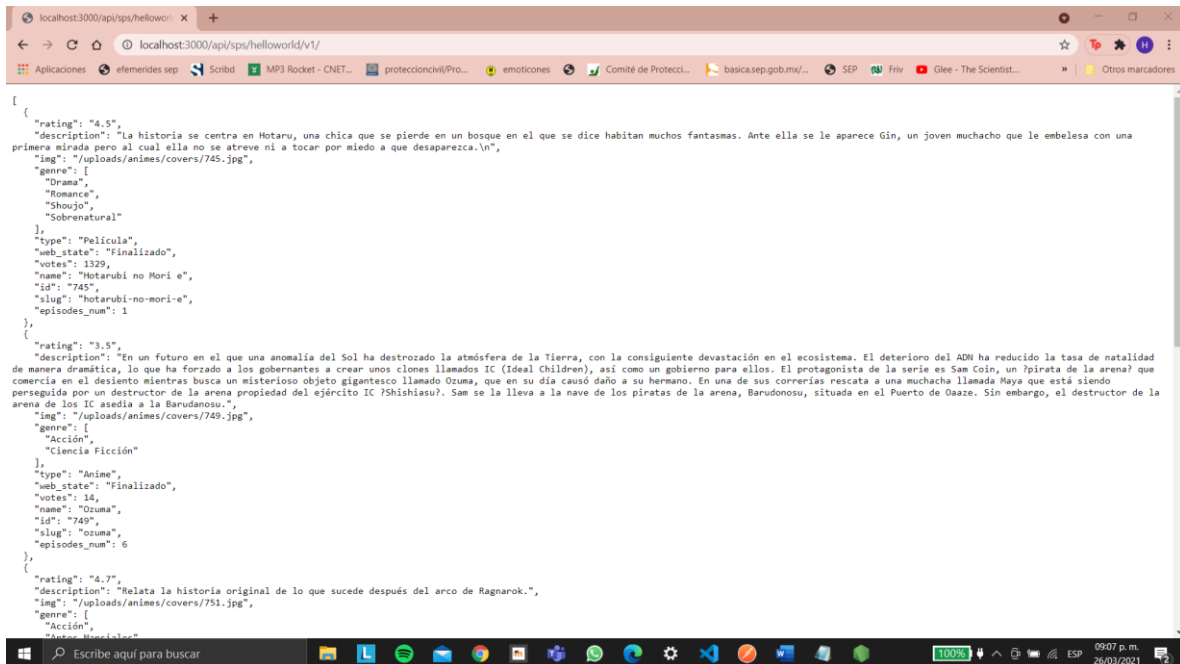
Posteriormente a dicha comprobación, se modificó el programa para que no solo enviara una respuesta satisfactoria, sino que también guardara el nuevo anime dentro de la base de datos ya existente.



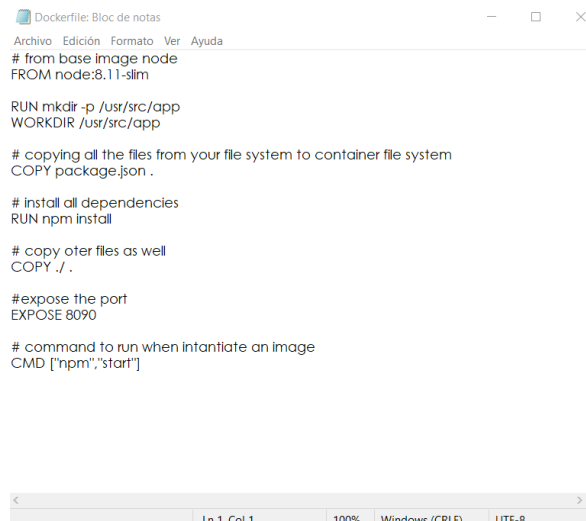
De igual manera, como se observa en la imagen anterior, se modific\u00f3 el c\u00f3digo de error para que el cliente pueda obtener m\u00e1s informaci\u00f3n, como el status de Internal Server Error.



A continuaci\u00f3n, se cambi\u00f3 el endpoint al solicitado dentro de la pr\u00e1ctica, realizando la modificaci\u00f3n al nombre del archivo `anime.js`, como se puede observar en la imagen.



Para contenerizar la API creada, se creó un archivo sin nombre titulado Dockerfile donde se incluyeron las instrucciones para crear la imagen en Docker. Dichas instrucciones debieron ser modificadas, pues al utilizar nodemon, no era posible el envío del comando start.



Posteriormente, se corrieron los comandos necesarios para obtener la imagen utilizando la terminal de Windows, obteniéndola satisfactoriamente. Dichos comandos, como se puede observar, son : Docker build -t <etiqueta> <path>.


```

C:\Windows\PowerShell>
latest: Pulling from library/busybox
ba3af22564a: Pull complete
Digest: sha256:c2360d5189a033012fbad1635e07be86f23b65cf6d76b436d0931af390a2ac
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
PS C:\Users\valde> docker run http://localhost:8090/api/sps/helloworld/v1
docker: invalid reference format.

See 'docker run --help'.

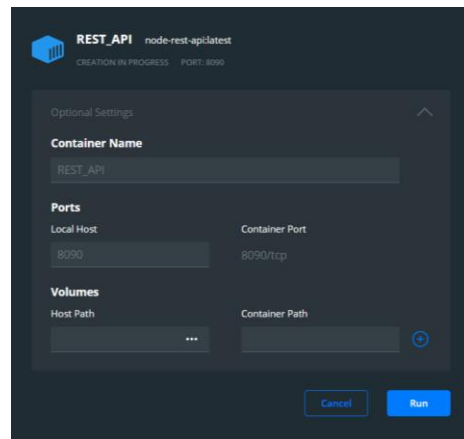
PS C:\Users\valde> docker build -t node-rest-api
'docker build' requires exactly 1 argument.

See 'docker build --help'.

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
PS C:\Users\valde> docker build -t node-rest-api D:/Halded/Documentos/Rest_API
[*] Building 21.6s (12/12) FINISHED
[+] Building 21.6s (12/12) FINISHED
  #0  FROM library/busybox:latest
  #0  WORKDIR /app
  #0  COPY . .
  #0  RUN npm install
  #0  RUN npm run build
  #0  RUN npm run start
  #0  CMD ["node", "dist/index.js"]
  #0  DONE 0.0s
  #1  FROM library/busybox:latest
  #1  WORKDIR /app
  #1  COPY . .
  #1  RUN npm install
  #1  RUN npm run build
  #1  RUN npm run start
  #1  CMD ["node", "dist/index.js"]
  #1  DONE 0.0s
  #2  FROM library/busybox:latest
  #2  WORKDIR /app
  #2  COPY . .
  #2  RUN npm install
  #2  RUN npm run build
  #2  RUN npm run start
  #2  CMD ["node", "dist/index.js"]
  #2  DONE 0.0s
  #3  FROM library/busybox:latest
  #3  WORKDIR /app
  #3  COPY . .
  #3  RUN npm install
  #3  RUN npm run build
  #3  RUN npm run start
  #3  CMD ["node", "dist/index.js"]
  #3  DONE 0.0s
  #4  FROM library/busybox:latest
  #4  WORKDIR /app
  #4  COPY . .
  #4  RUN npm install
  #4  RUN npm run build
  #4  RUN npm run start
  #4  CMD ["node", "dist/index.js"]
  #4  DONE 0.0s
  #5  FROM library/busybox:latest
  #5  WORKDIR /app
  #5  COPY . .
  #5  RUN npm install
  #5  RUN npm run build
  #5  RUN npm run start
  #5  CMD ["node", "dist/index.js"]
  #5  DONE 0.0s
  #6  FROM library/busybox:latest
  #6  WORKDIR /app
  #6  COPY . .
  #6  RUN npm install
  #6  RUN npm run build
  #6  RUN npm run start
  #6  CMD ["node", "dist/index.js"]
  #6  DONE 0.0s
  #7  FROM library/busybox:latest
  #7  WORKDIR /app
  #7  COPY . .
  #7  RUN npm install
  #7  RUN npm run build
  #7  RUN npm run start
  #7  CMD ["node", "dist/index.js"]
  #7  DONE 0.0s
  #8  FROM library/busybox:latest
  #8  WORKDIR /app
  #8  COPY . .
  #8  RUN npm install
  #8  RUN npm run build
  #8  RUN npm run start
  #8  CMD ["node", "dist/index.js"]
  #8  DONE 0.0s
  #9  FROM library/busybox:latest
  #9  WORKDIR /app
  #9  COPY . .
  #9  RUN npm install
  #9  RUN npm run build
  #9  RUN npm run start
  #9  CMD ["node", "dist/index.js"]
  #9  DONE 0.0s
  #10  FROM library/busybox:latest
  #10  WORKDIR /app
  #10  COPY . .
  #10  RUN npm install
  #10  RUN npm run build
  #10  RUN npm run start
  #10  CMD ["node", "dist/index.js"]
  #10  DONE 0.0s
  #11  FROM library/busybox:latest
  #11  WORKDIR /app
  #11  COPY . .
  #11  RUN npm install
  #11  RUN npm run build
  #11  RUN npm run start
  #11  CMD ["node", "dist/index.js"]
  #11  DONE 0.0s
  #12  FROM library/busybox:latest
  #12  WORKDIR /app
  #12  COPY . .
  #12  RUN npm install
  #12  RUN npm run build
  #12  RUN npm run start
  #12  CMD ["node", "dist/index.js"]
  #12  DONE 0.0s
  #13  FROM library/busybox:latest
  #13  WORKDIR /app
  #13  COPY . .
  #13  RUN npm install
  #13  RUN npm run build
  #13  RUN npm run start
  #13  CMD ["node", "dist/index.js"]
  #13  DONE 0.0s
  #14  FROM library/busybox:latest
  #14  WORKDIR /app
  #14  COPY . .
  #14  RUN npm install
  #14  RUN npm run build
  #14  RUN npm run start
  #14  CMD ["node", "dist/index.js"]
  #14  DONE 0.0s
  #15  FROM library/busybox:latest
  #15  WORKDIR /app
  #15  COPY . .
  #15  RUN npm install
  #15  RUN npm run build
  #15  RUN npm run start
  #15  CMD ["node", "dist/index.js"]
  #15  DONE 0.0s
  #16  FROM library/busybox:latest
  #16  WORKDIR /app
  #16  COPY . .
  #16  RUN npm install
  #16  RUN npm run build
  #16  RUN npm run start
  #16  CMD ["node", "dist/index.js"]
  #16  DONE 0.0s
  #17  FROM library/busybox:latest
  #17  WORKDIR /app
  #17  COPY . .
  #17  RUN npm install
  #17  RUN npm run build
  #17  RUN npm run start
  #17  CMD ["node", "dist/index.js"]
  #17  DONE 0.0s
  #18  FROM library/busybox:latest
  #18  WORKDIR /app
  #18  COPY . .
  #18  RUN npm install
  #18  RUN npm run build
  #18  RUN npm run start
  #18  CMD ["node", "dist/index.js"]
  #18  DONE 0.0s
  #19  FROM library/busybox:latest
  #19  WORKDIR /app
  #19  COPY . .
  #19  RUN npm install
  #19  RUN npm run build
  #19  RUN npm run start
  #19  CMD ["node", "dist/index.js"]
  #19  DONE 0.0s
  #20  FROM library/busybox:latest
  #20  WORKDIR /app
  #20  COPY . .
  #20  RUN npm install
  #20  RUN npm run build
  #20  RUN npm run start
  #20  CMD ["node", "dist/index.js"]
  #20  DONE 0.0s
  #21  FROM library/busybox:latest
  #21  WORKDIR /app
  #21  COPY . .
  #21  RUN npm install
  #21  RUN npm run build
  #21  RUN npm run start
  #21  CMD ["node", "dist/index.js"]
  #21  DONE 0.0s
  #22  FROM library/busybox:latest
  #22  WORKDIR /app
  #22  COPY . .
  #22  RUN npm install
  #22  RUN npm run build
  #22  RUN npm run start
  #22  CMD ["node", "dist/index.js"]
  #22  DONE 0.0s
  #23  FROM library/busybox:latest
  #23  WORKDIR /app
  #23  COPY . .
  #23  RUN npm install
  #23  RUN npm run build
  #23  RUN npm run start
  #23  CMD ["node", "dist/index.js"]
  #23  DONE 0.0s
  #24  FROM library/busybox:latest
  #24  WORKDIR /app
  #24  COPY . .
  #24  RUN npm install
  #24  RUN npm run build
  #24  RUN npm run start
  #24  CMD ["node", "dist/index.js"]
  #24  DONE 0.0s
  #25  FROM library/busybox:latest
  #25  WORKDIR /app
  #25  COPY . .
  #25  RUN npm install
  #25  RUN npm run build
  #25  RUN npm run start
  #25  CMD ["node", "dist/index.js"]
  #25  DONE 0.0s
  #26  FROM library/busybox:latest
  #26  WORKDIR /app
  #26  COPY . .
  #26  RUN npm install
  #26  RUN npm run build
  #26  RUN npm run start
  #26  CMD ["node", "dist/index.js"]
  #26  DONE 0.0s
  #27  FROM library/busybox:latest
  #27  WORKDIR /app
  #27  COPY . .
  #27  RUN npm install
  #27  RUN npm run build
  #27  RUN npm run start
  #27  CMD ["node", "dist/index.js"]
  #27  DONE 0.0s
  #28  FROM library/busybox:latest
  #28  WORKDIR /app
  #28  COPY . .
  #28  RUN npm install
  #28  RUN npm run build
  #28  RUN npm run start
  #28  CMD ["node", "dist/index.js"]
  #28  DONE 0.0s
  #29  FROM library/busybox:latest
  #29  WORKDIR /app
  #29  COPY . .
  #29  RUN npm install
  #29  RUN npm run build
  #29  RUN npm run start
  #29  CMD ["node", "dist/index.js"]
  #29  DONE 0.0s
  #30  FROM library/busybox:latest
  #30  WORKDIR /app
  #30  COPY . .
  #30  RUN npm install
  #30  RUN npm run build
  #30  RUN npm run start
  #30  CMD ["node", "dist/index.js"]
  #30  DONE 0.0s
  #31  FROM library/busybox:latest
  #31  WORKDIR /app
  #31  COPY . .
  #31  RUN npm install
  #31  RUN npm run build
  #31  RUN npm run start
  #31  CMD ["node", "dist/index.js"]
  #31  DONE 0.0s
  #32  FROM library/busybox:latest
  #32  WORKDIR /app
  #32  COPY . .
  #32  RUN npm install
  #32  RUN npm run build
  #32  RUN npm run start
  #32  CMD ["node", "dist/index.js"]
  #32  DONE 0.0s
  #33  FROM library/busybox:latest
  #33  WORKDIR /app
  #33  COPY . .
  #33  RUN npm install
  #33  RUN npm run build
  #33  RUN npm run start
  #33  CMD ["node", "dist/index.js"]
  #33  DONE 0.0s
  #34  FROM library/busybox:latest
  #34  WORKDIR /app
  #34  COPY . .
  #34  RUN npm install
  #34  RUN npm run build
  #34  RUN npm run start
  #34  CMD ["node", "dist/index.js"]
  #34  DONE 0.0s
  #35  FROM library/busybox:latest
  #35  WORKDIR /app
  #35  COPY . .
  #35  RUN npm install
  #35  RUN npm run build
  #35  RUN npm run start
  #35  CMD ["node", "dist/index.js"]
  #35  DONE 0.0s
  #36  FROM library/busybox:latest
  #36  WORKDIR /app
  #36  COPY . .
  #36  RUN npm install
  #36  RUN npm run build
  #36  RUN npm run start
  #36  CMD ["node", "dist/index.js"]
  #36  DONE 0.0s
  #37  FROM library/busybox:latest
  #37  WORKDIR /app
  #37  COPY . .
  #37  RUN npm install
  #37  RUN npm run build
  #37  RUN npm run start
  #37  CMD ["node", "dist/index.js"]
  #37  DONE 0.0s
  #38  FROM library/busybox:latest
  #38  WORKDIR /app
  #38  COPY . .
  #38  RUN npm install
  #38  RUN npm run build
  #38  RUN npm run start
  #38  CMD ["node", "dist/index.js"]
  #38  DONE 0.0s
  #39  FROM library/busybox:latest
  #39  WORKDIR /app
  #39  COPY . .
  #39  RUN npm install
  #39  RUN npm run build
  #39  RUN npm run start
  #39  CMD ["node", "dist/index.js"]
  #39  DONE 0.0s
  #40  FROM library/busybox:latest
  #40  WORKDIR /app
  #40  COPY . .
  #40  RUN npm install
  #40  RUN npm run build
  #40  RUN npm run start
  #40  CMD ["node", "dist/index.js"]
  #40  DONE 0.0s
  #41  FROM library/busybox:latest
  #41  WORKDIR /app
  #41  COPY . .
  #41  RUN npm install
  #41  RUN npm run build
  #41  RUN npm run start
  #41  CMD ["node", "dist/index.js"]
  #41  DONE 0.0s
  #42  FROM library/busybox:latest
  #42  WORKDIR /app
  #42  COPY . .
  #42  RUN npm install
  #42  RUN npm run build
  #42  RUN npm run start
  #42  CMD ["node", "dist/index.js"]
  #42  DONE 0.0s
  #43  FROM library/busybox:latest
  #43  WORKDIR /app
  #43  COPY . .
  #43  RUN npm install
  #43  RUN npm run build
  #43  RUN npm run start
  #43  CMD ["node", "dist/index.js"]
  #43  DONE 0.0s
  #44  FROM library/busybox:latest
  #44  WORKDIR
```

Por último, se creó un contenedor dentro del programa Docker, en el que se especificó que el puerto en el que se mostraría la API sería el 8090. Obteniendo una corrida exitosa del mismo al terminar de crearse.



Todos los códigos utilizados dentro de la creación de dicha práctica se pueden encontrar dentro del repositorio: https://github.com/haidee244/sps_practica.