

CEIoT - FIUBA

Desarrollo de Aplicaciones Web

8va cohorte 2023

Trabajo Práctico Final

Autor

Ing. Fabián Alejandro Banderas Benítez

1 Introducción

1.1 Propósito

El objetivo de este documento es presentar el despliegue de una aplicación IoT de Smart Home. Dicha presentación será utilizada como Trabajo Práctico Final de la materia Desarrollo de Aplicaciones web.

1.2 Alcance

En el presente definirá:

- Documentación y estructuración del proyecto.
- Claridad de programación, documentación de código.
- Resolución de las consignas aplicando los conocimientos vistos en clase.

2 Objetivos

2.1 Tener en un repositorio con todo el código fuente de la materia versionado.

Repositorio versionado [app-fullstack-base-2023-i08 Fabián Banderas](#)

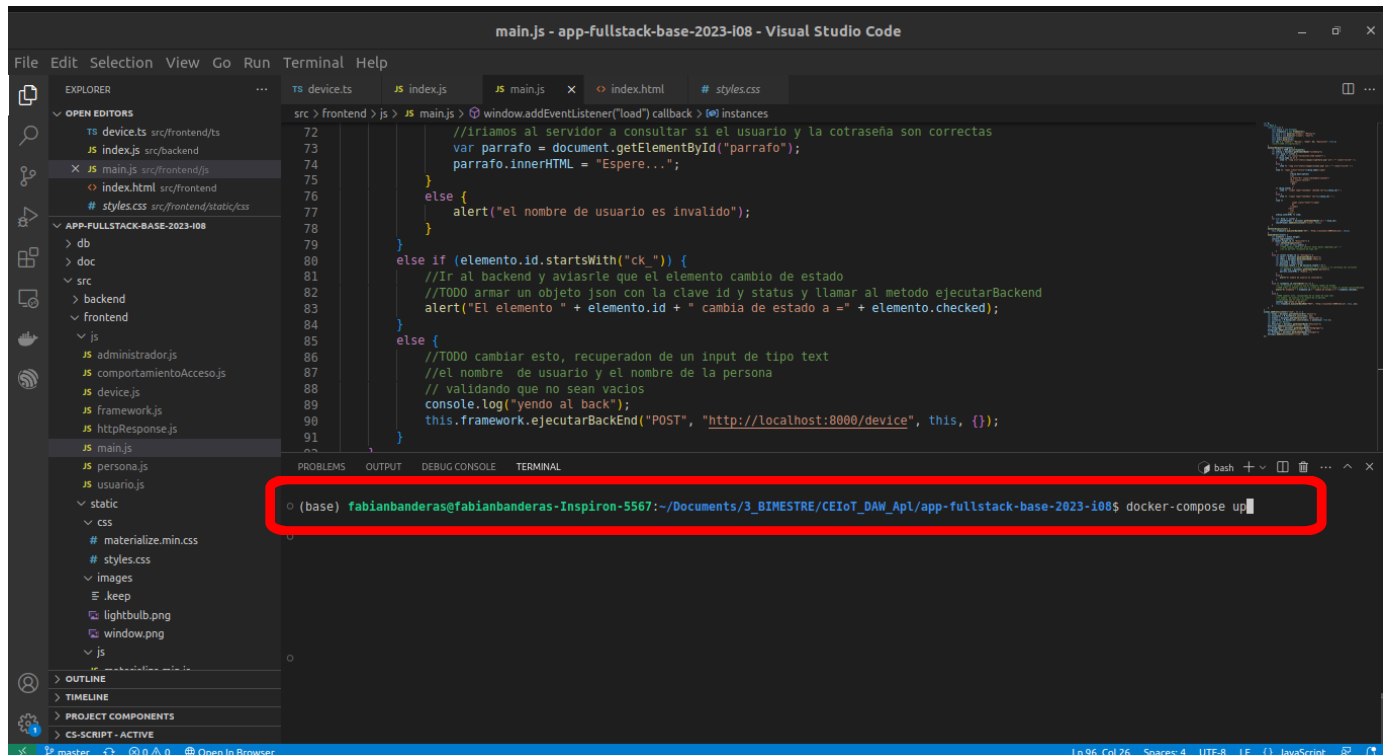
2.2 Tener un archivo README dentro del repositorio del proyecto con toda la información necesaria de contexto, así como también para poder correrlo.

[Link README.md](#)

2.3 Correr el proyecto desarrollado con la herramienta Docker Compose.

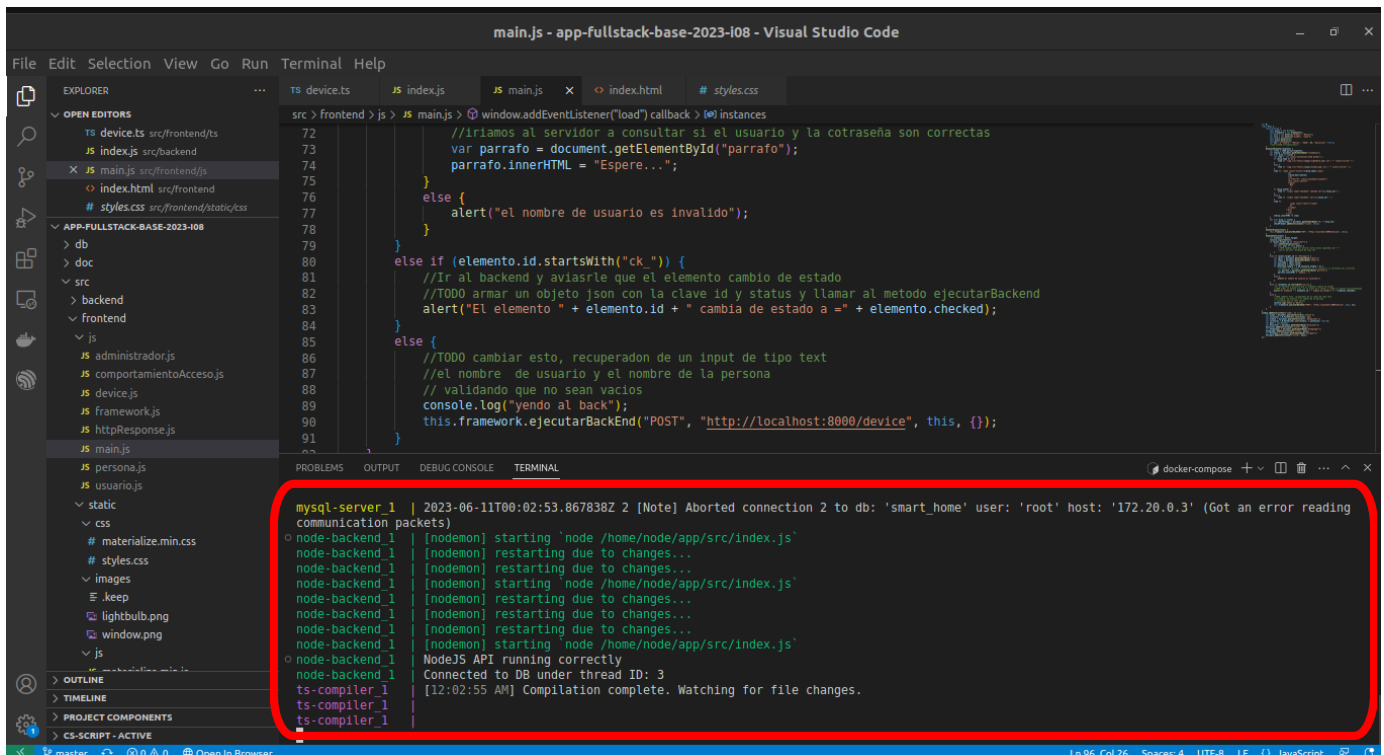
Para correr es necesario utilizar el comando:

docker-compose up



Si todo inicia se puede mostrar de la siguiente forma:

Trabajo Práctico Final



```
main.js - app-fullstack-base-2023-i08 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    TS device.ts src/frontend/ts
    JS index.js src/backend
    JS main.js src/frontend/js
    index.html src/frontend
    # styles.css src/frontend/static/css
  APP-FULLSTACK-BASE-2023-I08
    > db
    > doc
    > src
    > backend
    > frontend
    > js
    JS administrador.js
    JS comportamientoAcceso.js
    JS device.js
    JS framework.js
    JS httpResponse.js
    JS main.js
    JS persona.js
    JS usuario.js
    > static
    > css
    # materialize.min.css
    # styles.css
    > images
    E keep
    lightbulb.png
    window.png
    > js
    > OUTLINE
    > TIMELINE
    > PROJECT COMPONENTS
    > CS-SCRIPT - ACTIVE

main.js
72 //Iriamos al servidor a consultar si el usuario y la contraseña son correctas
73 var parrafo = document.getElementById("parrafo");
74 parrafo.innerHTML = "Espere...";
75 }
76 else {
77     alert("el nombre de usuario es invalido");
78 }
79 }
80 //Ir al backend y avisarle que el elemento cambio de estado
81 //TODO armar un objeto json con la clave id y status y llamar al metodo ejecutarBackend
82 alert("El elemento " + elemento.id + " cambia de estado a = " + elemento.checked);
83 }
84 }
85 else {
86     //TODO cambiar esto, recuperadon de un input de tipo text
87     //el nombre de usuario y el nombre de la persona
88     // validando que no sean vacios
89     console.log("yendo al back");
90     this.framework.ejecutarBackend("POST", "http://localhost:8000/device", this, {});
91 }
92 }

TERMINAL
mysql-server_1 | 2023-06-11T00:02:53.867838Z [Note] Aborted connection 2 to db: 'smart_home' user: 'root' host: '172.20.0.3' (Got an error reading communication packets)
node-backend_1 | [nodemon] starting `node /home/node/app/src/index.js`
node-backend_1 | [nodemon] restarting due to changes...
node-backend_1 | [nodemon] restarting due to changes...
node-backend_1 | [nodemon] starting `node /home/node/app/src/index.js`
node-backend_1 | [nodemon] restarting due to changes...
node-backend_1 | [nodemon] restarting due to changes...
node-backend_1 | [nodemon] restarting due to changes...
node-backend_1 | [nodemon] starting `node /home/node/app/src/index.js`
node-backend_1 | NodeJS API running correctly
node-backend_1 | Connected to DB under thread ID: 3
ts-compiler_1 | [12:02:55 AM] Compilation complete. Watching for file changes.
ts-compiler_1 |
```

2.4 Entregar el repositorio con todo lo necesario para correr la aplicación.

[Repositorio](#)

3 Actividades a realizar

A continuación se describen las actividades realizadas para completar el proyecto.

3.1 Documentación y código.

- README.md modificado del proyecto, contiene la información de introducción al proyecto, cómo instalar las dependencias y cómo ejecutar la aplicación. Además serán valorados si se agregan detalles extras tales como interacción de API entre

frontend-backend, con detalles adicionales

- El código es lo más prolijo posible, cuidando tabulaciones, los nombres de variables y funciones, comentarios en el código.

3.2 Propuestas de frontend

- Edición de dispositivos existentes. Nombre, descripción, tipo.
- Posibilidad de agregar o quitar dispositivos. Aplicación de componentes de la biblioteca Materialize y los principios de diseño de una Single Page Application.
- Se indica qué valores de estado puede asumir el dispositivo. Actualmente los dispositivos asumen un estado binario (0 ó 1) a través de un interruptor (switch) implementado en Materialize. En un sistema de domótica es deseable que se contemple dimerizar las luces o regular la apertura de cortinas. Esta intensidad, si el dispositivo la soporta, se indica utilizando los números reales comprendidos entre 0 y 1, siendo por ejemplo 0.5 una lámpara dimerizada al 50% o cortinas a mitad de su recorrido. Para manipular el valor utiliza un control Materialize que permite elegir entre un rango numérico. No olvidar que en Javascript/Typescript, todo número se representa como un número real.

3.3 Propuestas de backend

- Tiene la posibilidad desde el backend para actualizar el valor del estado y de actualizar cualquier valor.
- Funcionalidad para borrar un dispositivo.
- Funcionalidad para agregar un nuevo dispositivo.