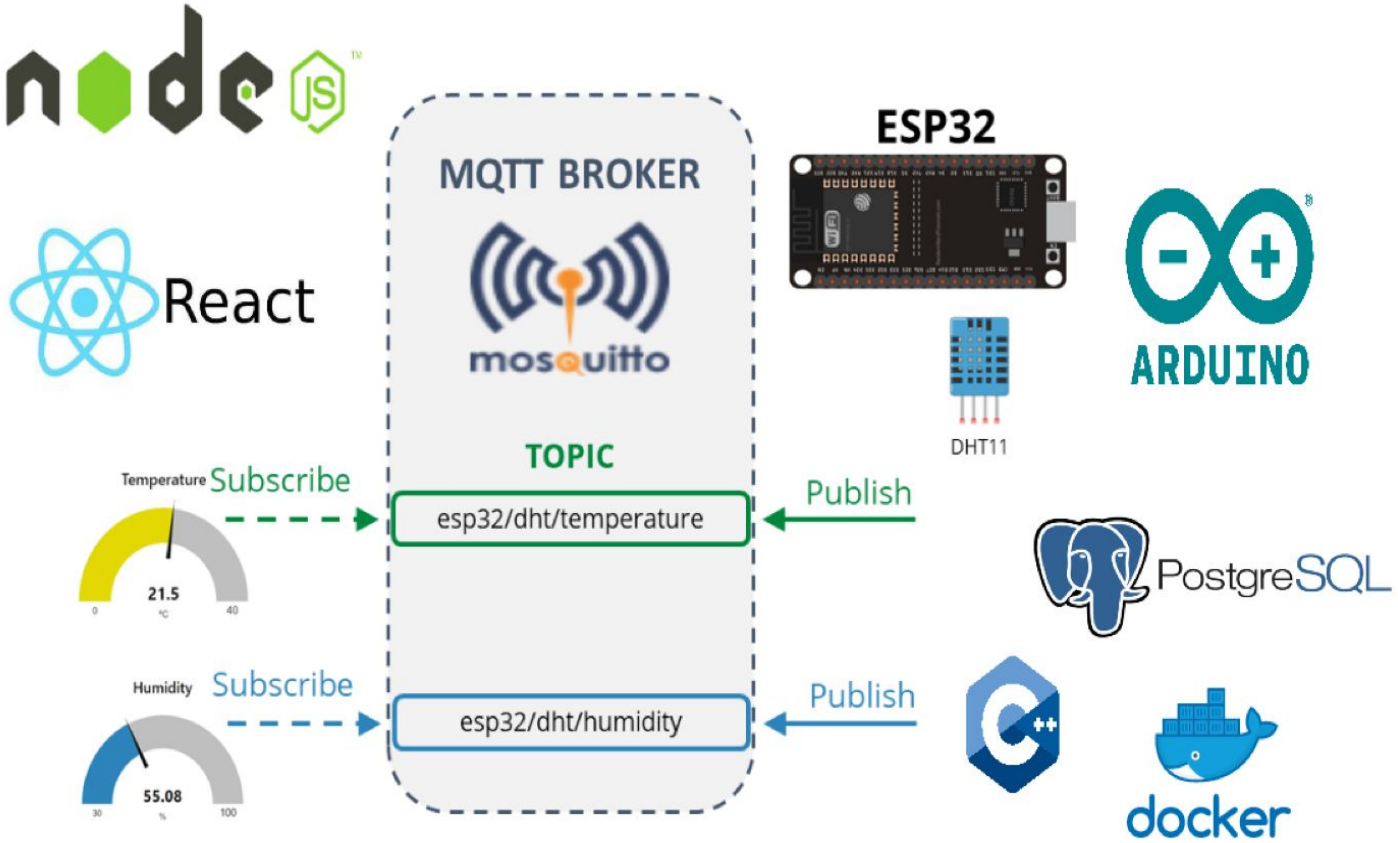
Projeto de IOT

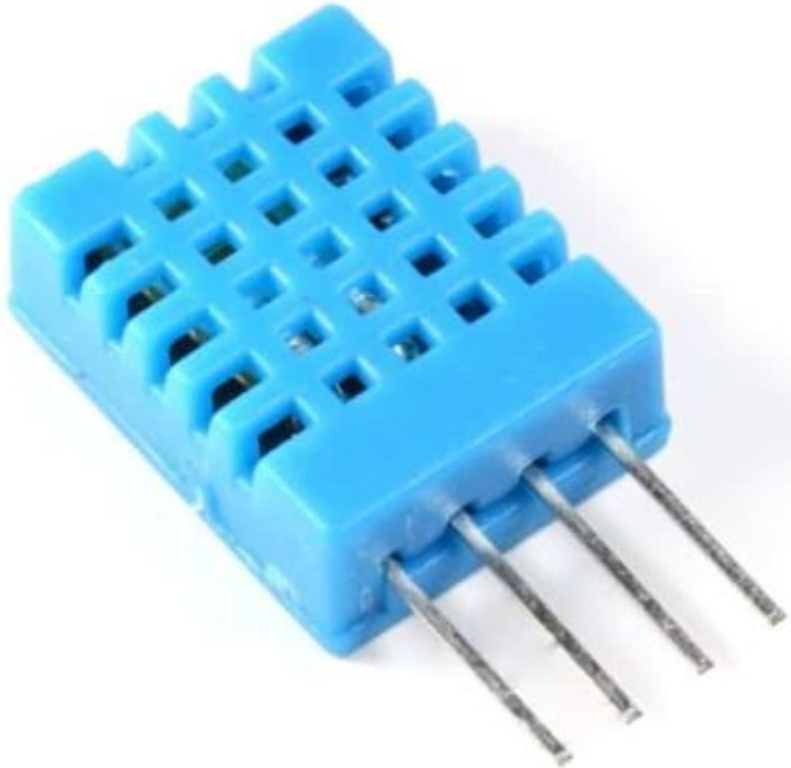
Gustav Shigueo Nicioka Asano 11212355
Marcos Antonio Nobre Coutinho 10716397
Pedro Fernando Christofoletti dos Santos 11218560
Altair Fernando Pereira Junior 9391831

Objetivo: Publicar leituras de temperatura e umidade de um sensor DHT11 via MQTT com o módulo ESP32. As mensagens publicadas no broker serão capturadas por uma API que será consumida por uma interface Web.

Uma imagem vale mais que mil palavras



DHT11



Sensor de Umidade e Temperatura DHT11

REF: 9SS10

O DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, **muito usado** para projetos com Arduino.

```
// Digital pin connected to the DHT sensor
#define DHTPIN 4
#define DHTTYPE DHT11

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE);
```

ESP32 - Código em C++

Bibliotecas usadas:

```
#include "DHT.h"
#include <WiFi.h>
#include "esp_wpa2.h"
extern "C" {
  #include "freertos/FreeRTOS.h"
  #include "freertos/timers.h"
}
#include <AsyncMqttClient.h>
```

Conexão com Wifi:

```
//Eduroam enterprise connection
bool shouldConnectToEduroam = true;
#define WIFI_EDUROAM_SSID "eduroam"
#define EDUROAM_EAP_IDENTITY "11111111@usp.br"
#define EDUROAM_EAP_PASSWORD "senha123Usp"

//Home connection
#define HOME_WIFI_SSID "Meu_Wifi"
#define HOME_WIFI_PASSWORD "senha123"
```

ESP32 - Código em C++

Conexão com o Broker

```
// Mosquitto MQTT Broker
const char *MQTT_HOST = "andromeda.lasdpc.icmc.usp.br";
#define MQTT_PORT 8123
#define MQTT_USER "admin"
#define MQTT_PASSWRD "admin12345"
```

Tópicos

```
// Temperature MQTT Topics
#define MQTT_PUB_TEMP "esp32/dht/temperature"
#define MQTT_PUB_HUM  "esp32/dht/humidity"
```

Loop

```
void loop() {
  unsigned long currentMillis = millis();
  // Every X number of seconds (interval = 10 seconds)
  // it publishes a new MQTT message
  if (currentMillis - previousMillis >= interval) {
    // Save the last time a new reading was published
    previousMillis = currentMillis;
    // New DHT sensor readings
    hum = dht.readHumidity();
    // Read temperature as Celsius (the default)
    temp = dht.readTemperature();

    // Check if any reads failed and exit early (to try again).
    if (isnan(temp) || isnan(hum)) {
      Serial.println(F("Failed to read from DHT sensor!"));
      return;
    }

    // Publish an MQTT message on topic esp32/dht/temperature
    uint16_t packetIdPub1 = mqttClient.publish(MQTT_PUB_TEMP, 1, true, String(temp).c_str());
    Serial.printf("Publishing on topic %s at QoS 1, packetId: %i, ", MQTT_PUB_TEMP, packetIdPub1);
    Serial.printf("Temperature: %.2f \n", temp);

    // Publish an MQTT message on topic esp32/dht/humidity
    uint16_t packetIdPub2 = mqttClient.publish(MQTT_PUB_HUM, 1, true, String(hum).c_str());
    Serial.printf("Publishing on topic %s at QoS 1, packetId %i:, ", MQTT_PUB_HUM, packetIdPub2);
    Serial.printf("Humidity: %.2f \n", hum);
  }
}
```

Mosquitto

Docker compose

docker-compose-mosquitto.yaml

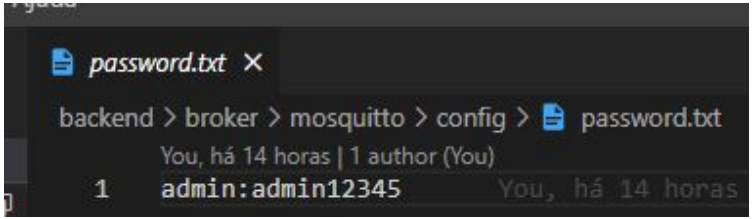
backend > broker > docker-compose-mosquitto.yaml

You, há 12 horas | 1 author (You)

```yaml
version: '3.7'

services:
#Servico MQTT usando o eclipse-mosquitto utilizando as seguintes pastas para guardar as configuracoes, os dados e os logs utilizando a porta 8123.
  mqtt:
    container_name: mqtt
    image: eclipse-mosquitto
    restart: always
    volumes:      You, há 14 horas • Adding mosquitto setup …
      - ./mosquitto:/mosquitto
      - ./mosquitto/config/mosquitto.conf:/mosquitto/config/mosquitto.conf
      - ./mosquitto/data:/mosquitto/data
      - ./mosquitto/log:/mosquitto/log
    ports:
      - 8123:8123

```

Mosquitto

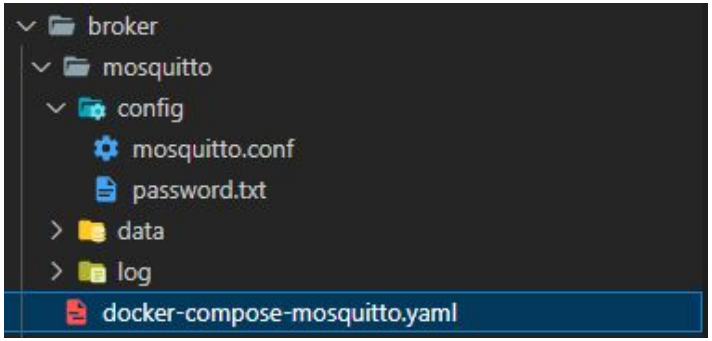
Conf e Senha




```
You, há 13 horas | 2 authors (Gustav Shigueo Nicioka Asano and others)
#Arquivo de configuracao do eclipse mosquitto.

#Guardar os dados de conexao, inscricao e os dados das mensagens na pasta especificada.
persistence true
persistence_location /mosquitto/data/

#Destino dos logs em um arquivo.
log_dest file /mosquitto/log/mosquitto.log

#Especificar a porta para conexao.
listener 8123

#Seguranca: Temporariamente liberado o acesso sem usuario e senha que estao armazenados no seguinte arquivo.
allow_anonymous true
#password_file /mosquitto/config/password.txt      You, há 13 horas • Comment line 15 …
```

Mosquitto

Configuração e Estrutura




```
docker compose -f giotgrad09/backend/broker/docker-compose-mosquitto.yaml up -d --remove-orphans
```

```
docker exec -it mqtt /bin/sh
```

```
mosquitto_passwd -U mosquitto/config/password.txt
```

```
vi mosquitto/config/mosquitto.conf
```

Descomente a linha 15 do arquivo `mosquitto.conf` e salve as alterações. (INSERT -> descomente -> ESC -> :wq)

```
exit
```

Restart o container

```
docker restart mqtt
```

Nodejs

Bibliotecas usadas

```
"dependencies": {
  "bcryptjs": "^2.4.3",
  "consign": "^0.1.6",
  "cors": "^2.8.5",
  "dotenv": "^16.0.3",
  "express": "^4.18.2",
  "jwt-simple": "^0.5.6",
  "knex": "^2.3.0",
  "mqtt": "^4.3.7",
  "passport": "^0.6.0",
  "passport-jwt": "^4.0.0",
  "pg": "^8.8.0",
  "pm2": "^5.2.2"
},
```

Ambiente docker

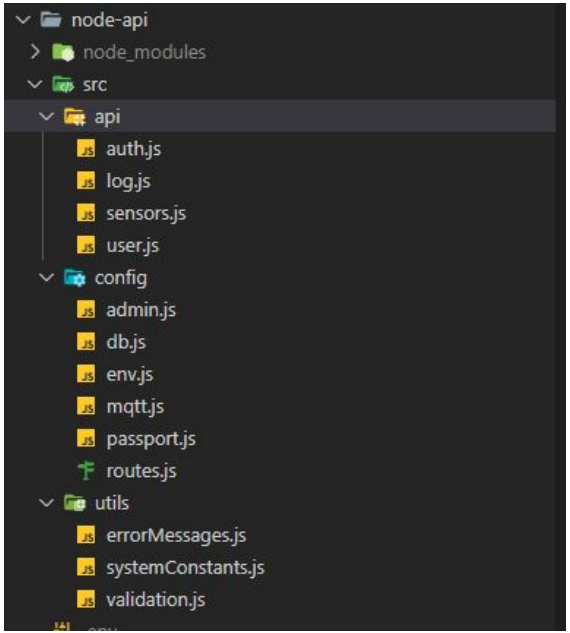
docker-compose-node-postgres.yml

backend > node-api > docker-compose-node-postgres.yml

```
services:
  db:
    volumes:
  api:
    container_name: node-api
    build:
      context: ./
    depends_on:
      - db
    ports:
      - '8323:8323'
    environment:
      DB_HOST: andromeda.lasdpc.icmc.usp.br
      DB_PORT: 8223
      DB_USER: postgres
      DB_PASSWORD: postgres
      DB_DATABASE: postgres
      AUTH_SECRET: 3dsdçfg&FDG(&#%GH%&sdfhhhkfg456erbs%%&!@@!{ÇBVB
      API_PORT: 8323
      MQTT_HOST: andromeda.lasdpc.icmc.usp.br
      MQTT_PORT: 8123
      MQTT_USERNAME: admin
      MQTT_PASSWORD: admin12345
      LOG: Y
      SHOULD_AUTHENTICATE: Y
    links:      You, mês passado • Adding docker stuff …
      - db
    volumes:
      - './:/src'
```

Nodejs

Estrutura

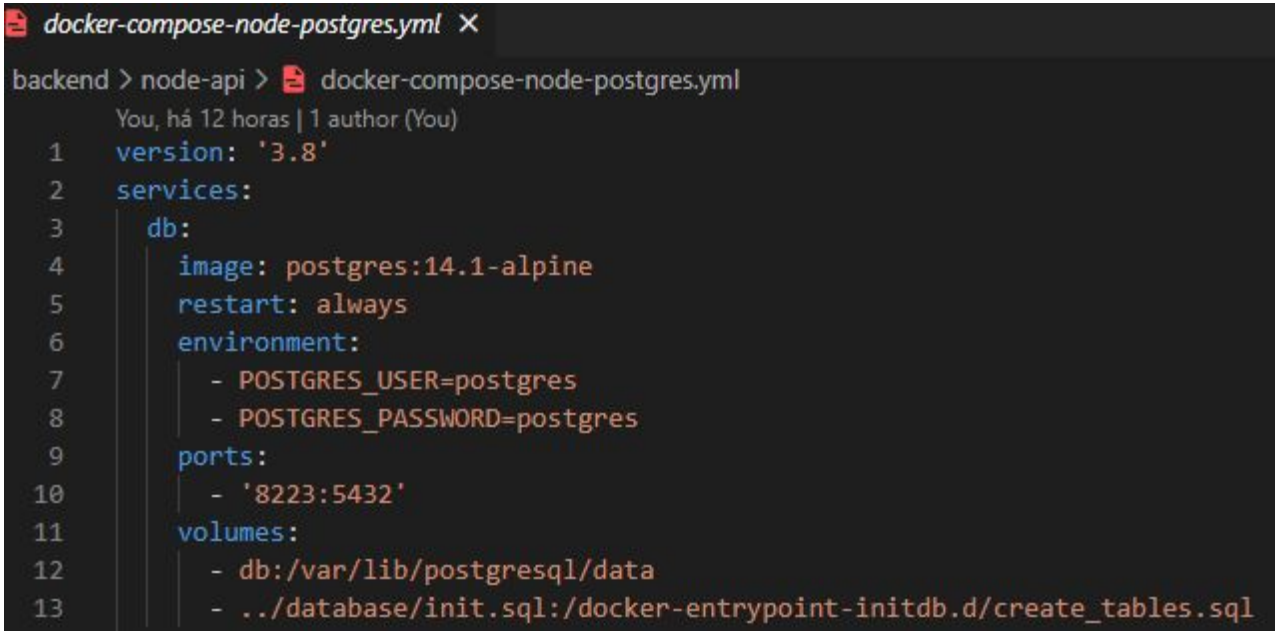


Rotas

```
module.exports = (app) => {
  //Test
  app.get("/", function (req, res) {
    res.send({ response: "Service is up" });
  })

  //Authentication
  app.post("/signin", app.src.api.auth.signin)
  app.post("/validateToken", app.src.api.auth.validateToken)

  //Users
  app
    .route("/users")
    .all(shouldAuthenticate ? app.src.config.passport.authenticate() : (req, res, next) => next())
    .post(admin(app.src.api.user.save))
    .get(admin(app.src.api.user.get))

  app
    .route("/users/:id")
    .all(shouldAuthenticate ? app.src.config.passport.authenticate() : (req, res, next) => next())
    .put(admin(app.src.api.user.save))
    .delete(admin(app.src.api.user.remove))

  //Sensors
  app
    .route("/temperature")
    .all(shouldAuthenticate ? app.src.config.passport.authenticate() : (req, res, next) => next())
    .get(app.src.api.sensors.getTemperature);

  //Humudity
  app
  .route("/humidity")
  .all(shouldAuthenticate ? app.src.config.passport.authenticate() : (req, res, next) => next())
  .get(app.src.api.sensors.getHumidity);

  //Logs
  app
    .route("/logs")
    .all(shouldAuthenticate ? app.src.config.passport.authenticate() : (req, res, next) => next())
    .get(admin(app.src.api.log.get))
};      You, há 2 meses • First release …
```

Postgresql

Docker compose



Postgresql

Schema

init.sql

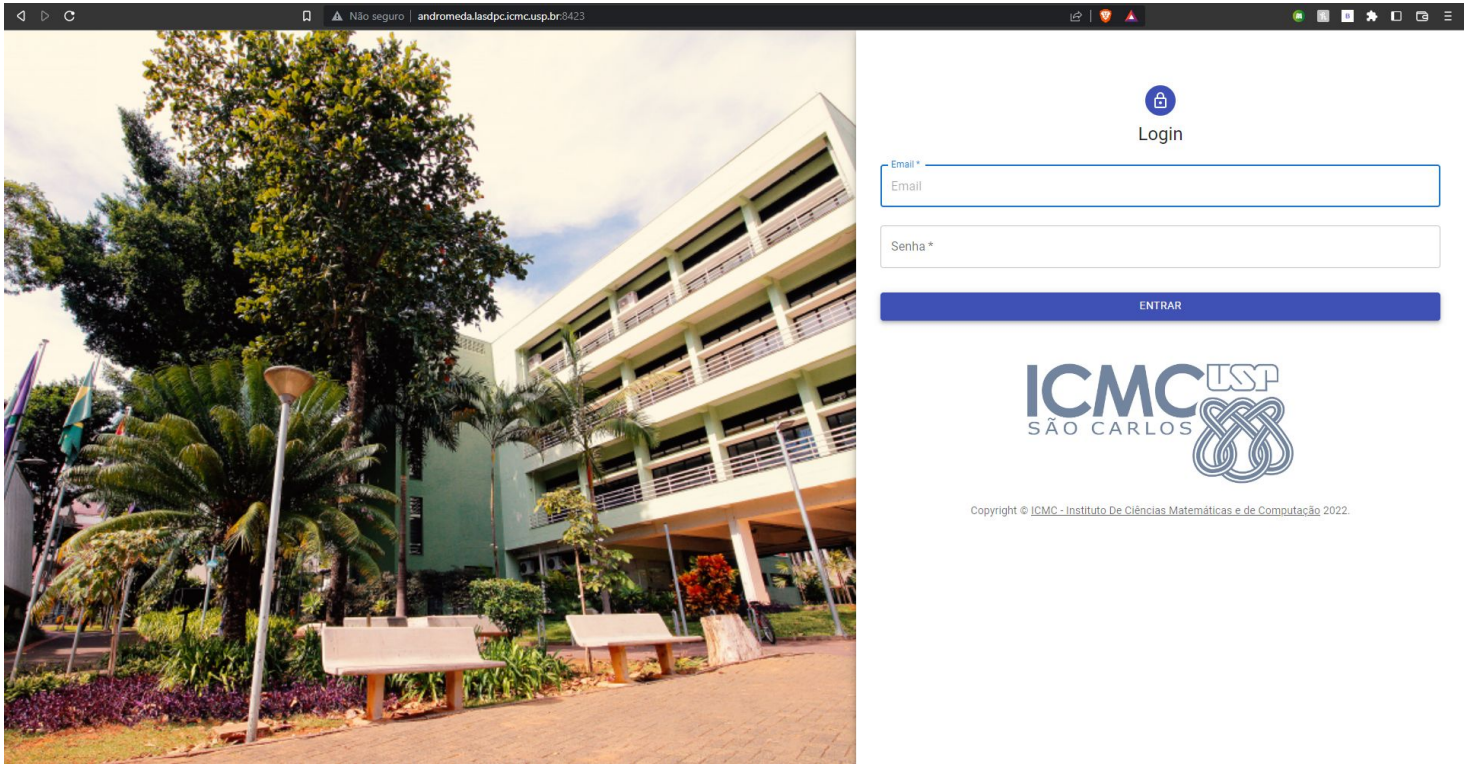
backend > database > init.sql

You, há 20 horas | 1 author (You) | Run on active connection | Select block

```sql
CREATE TABLE users (
    id SERIAL PRIMARY KEY NOT NULL,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE not NULL,
    password VARCHAR(256) NOT NULL,
    admin BOOLEAN DEFAULT FALSE NOT NULL
);

CREATE TABLE logs (
    id SERIAL PRIMARY key,
    user_id INTEGER NOT NULL,
    log_action VARCHAR(14) NOT NULL,
    log_type VARCHAR(10) NOT NULL,
    message TEXT NOT NULL,
    timestamp timestamp,

    CONSTRAINT fk_log_user FOREIGN KEY (user_id) REFERENCES users (id),
    CONSTRAINT ck_log_action CHECK (log_action IN ('GET_TEMP', 'GET_HUM', 'CREATE', 'GET', 'UPDATE', 'DELETE')),
    CONSTRAINT ck_log_type CHECK (log_type IN ('SUCCESS', 'FAILURE'))
);

INSERT INTO users (name, email, password, admin) VALUES ('Admin', 'admin@admin.com', '$2a$10$i/EDYJ9eFYEAG4QpXETBkeK089VVb9zAu6vwZ1J4dXA.sjchn6C9y', TRUE);
```

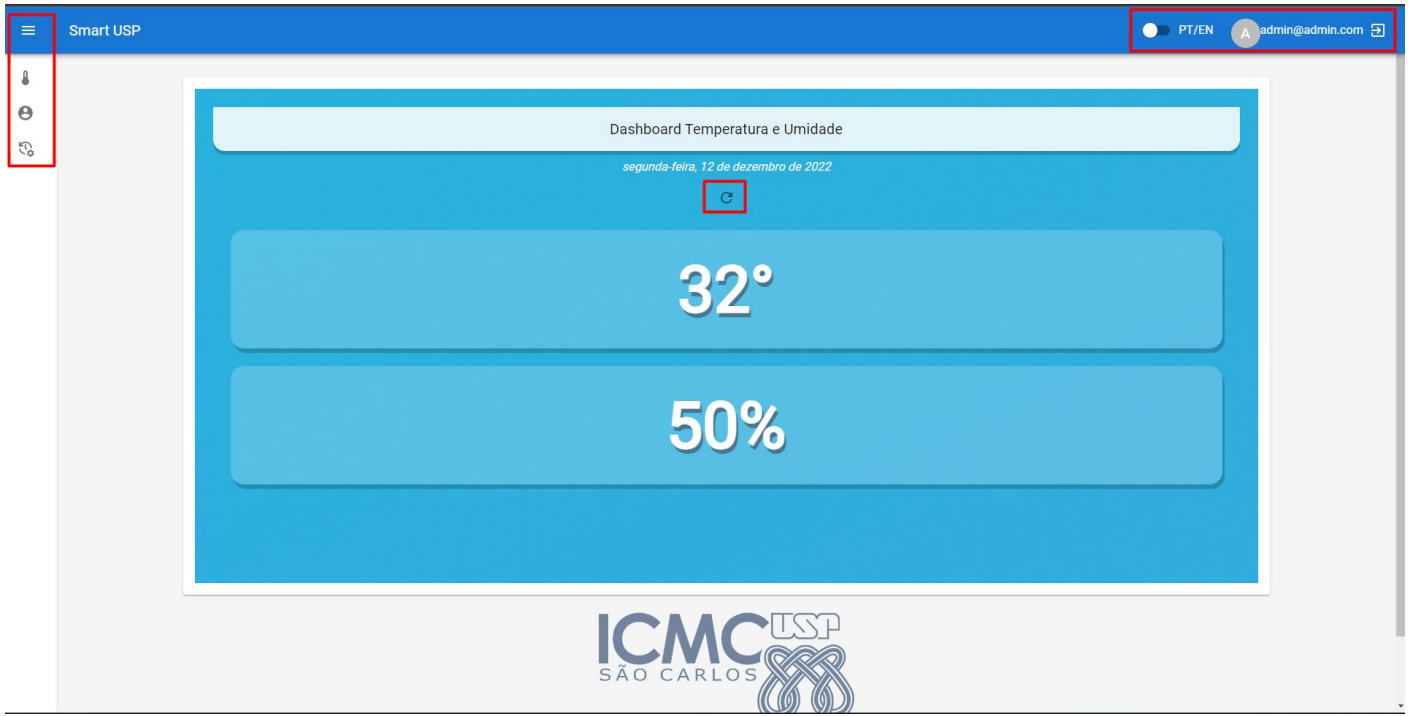
Reactjs

Login



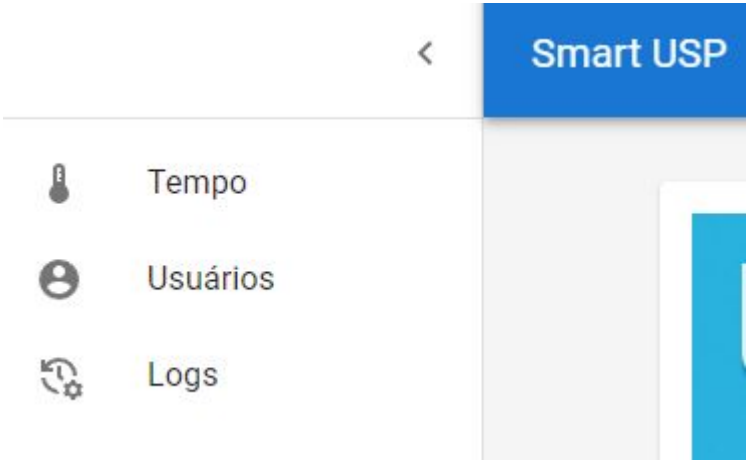
Reactjs

Home



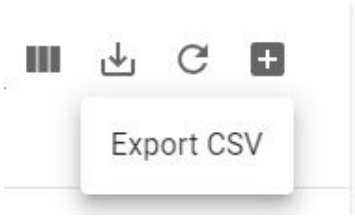
Reactjs

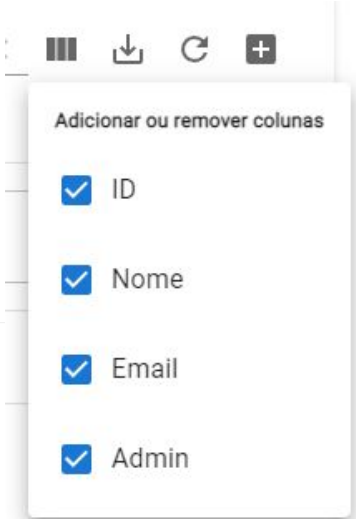
Rotas

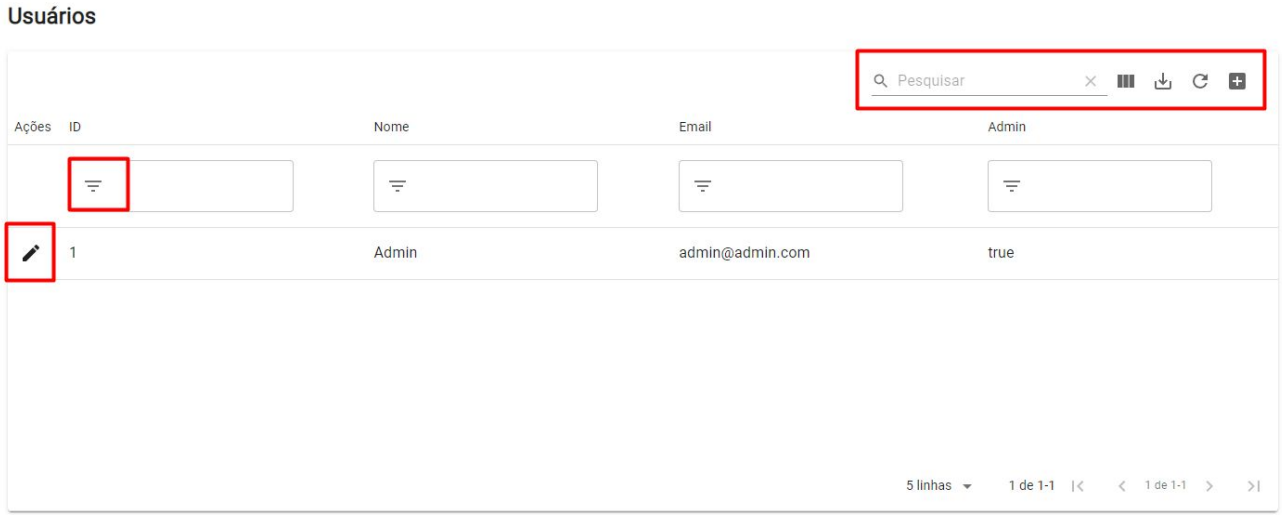


Reactjs

Usuários

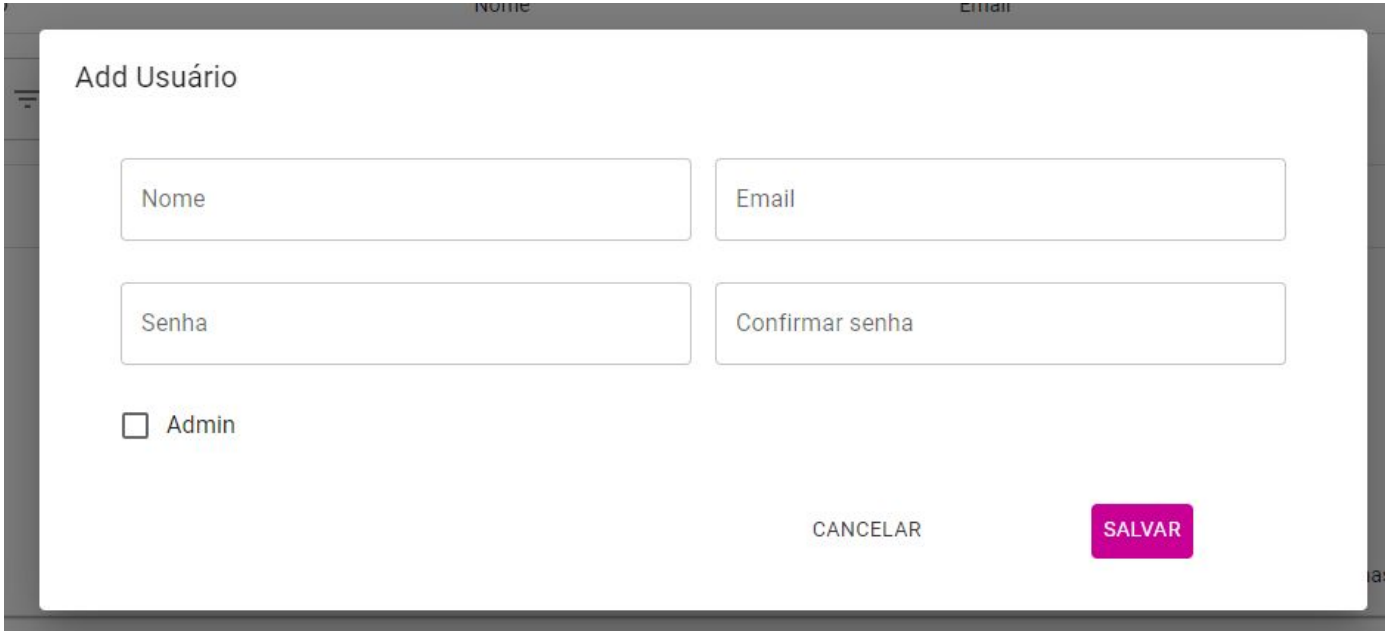






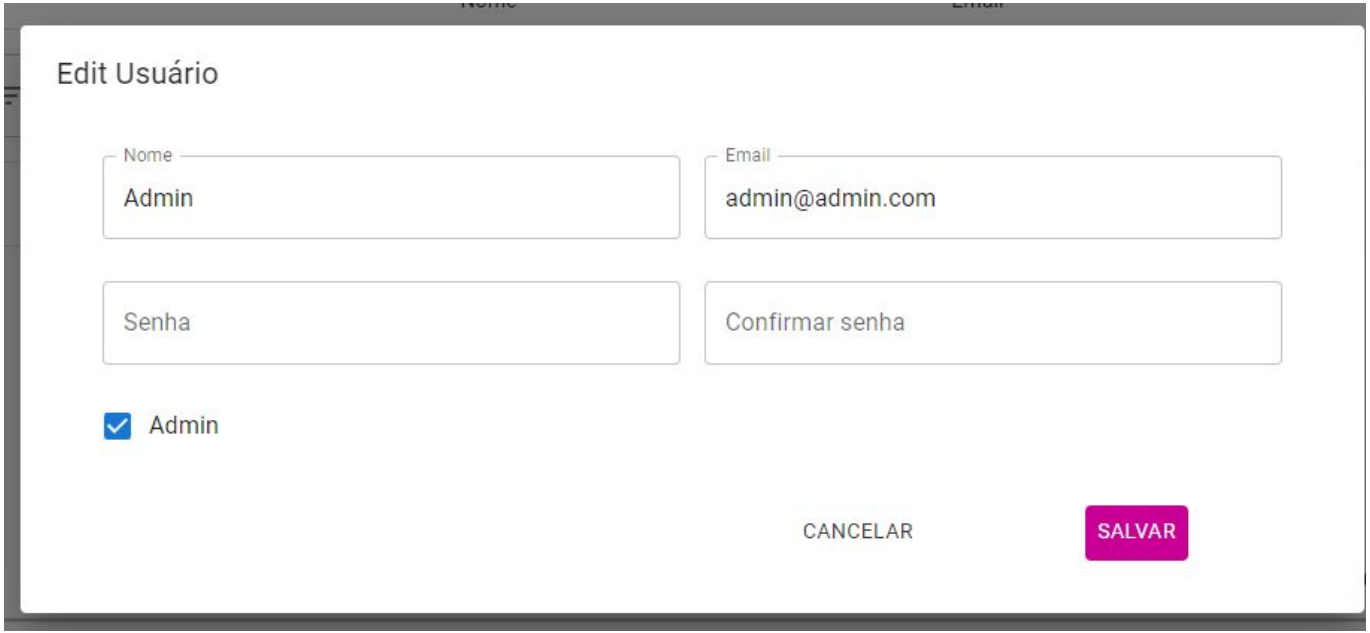
Reactjs

Usuários



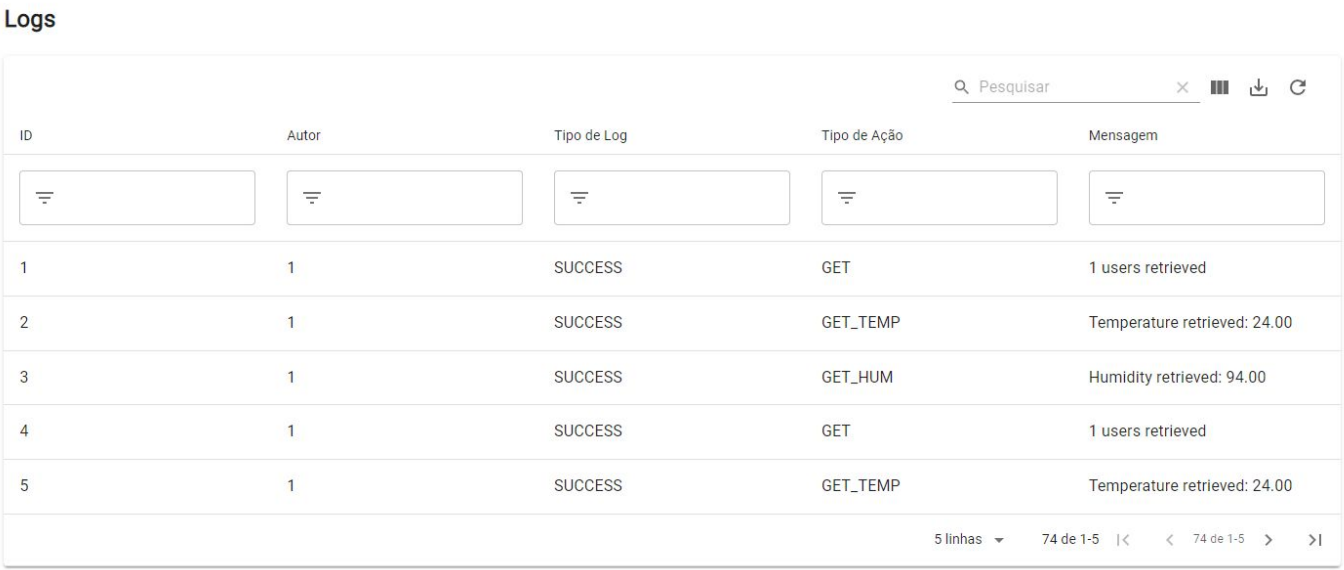
Reactjs

Usuários



Reactjs

Logs

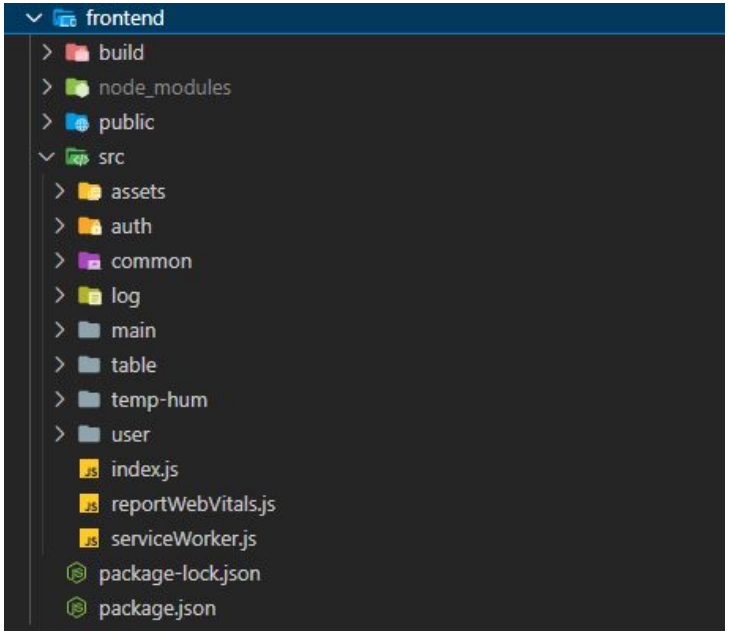


Reactjs

Bibliotecas

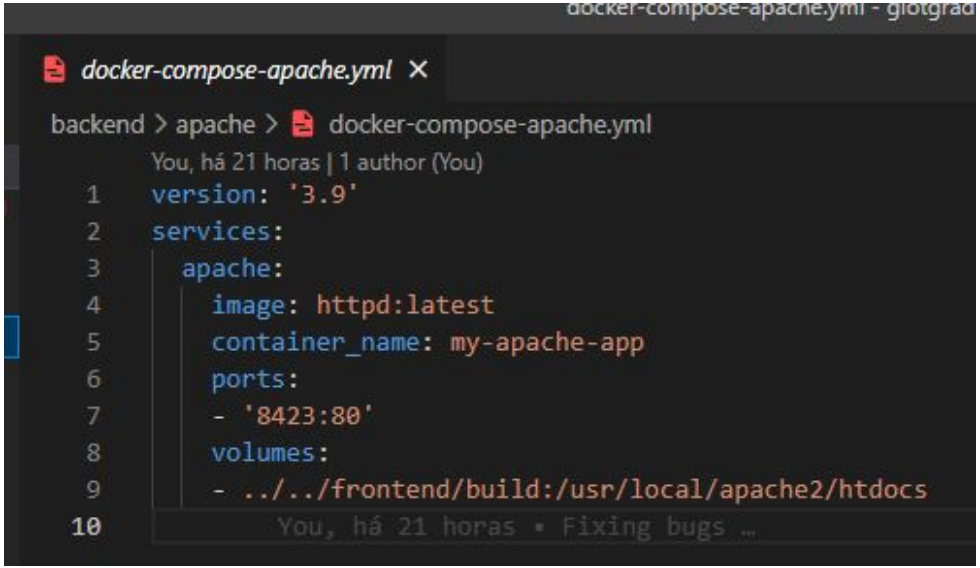
```
"dependencies": {
  "@emotion/react": "^11.7.0",
  "@emotion/styled": "^11.6.0",
  "@material-table/core": "^0.2.9",
  "@material-table/exporters": "^1.0.7",
  "@material-ui/core": "^5.0.0-beta.5",
  "@mui/icons-material": "^5.2.1",
  "@mui/material": "^5.2.3",
  "@mui/styles": "^5.2.3",
  "@testing-library/jest-dom": "^5.11.4",
  "@testing-library/react": "^11.1.0",
  "@testing-library/user-event": "^12.1.10",
  "axios": "^0.24.0",
  "clsx": "^1.1.1",
  "react": "^17.0.2",
  "react-avatar": "^4.1.0",
  "react-dom": "^17.0.2",
  "react-redux": "^7.2.6",
  "react-redux-toastr": "^7.6.6",
  "react-router-dom": "^6.0.2",
  "react-scripts": "4.0.3",
  "recompose": "^0.30.0",
  "redux": "^4.1.2",
  "redux-form": "^8.3.8",
  "redux-multi": "^0.1.12",
  "redux-promise": "^0.6.0",
  "redux-thunk": "^2.4.1",
  "web-vitals": "^1.0.1"
},
```

Estrutura



Apache

Docker compose



Resultado Final

```
giotgrad09@tau02-vm3:~$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                                                  NAMES
3a32aff6db8d   node-api_api         "docker-entrypoint.s…"   12 hours ago   Up 12 hours   0.0.0.0:8323->8323/tcp, :::8323->8323/tcp              node-api
b3d36fe2cc18   postgres:14.1-alpine "docker-entrypoint.s…"   12 hours ago   Up 12 hours   0.0.0.0:8223->5432/tcp, :::8223->5432/tcp              node-api-db-1
16b7d9ef78e3   eclipse-mosquitto    "/docker-entrypoint.…"   13 hours ago   Up 13 hours   1883/tcp, 0.0.0.0:8123->8123/tcp, :::8123->8123/tcp    mqtt
227565058c73   httpd:latest         "httpd-foreground"       21 hours ago   Up 21 hours   0.0.0.0:8423->80/tcp, :::8423->80/tcp                  my-apache-app
giotgrad09@tau02-vm3:~$
```

```
giotgrad09@tau02-vm3:~$ docker image ls
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
node-api_api        latest    b137b660a829   20 hours ago     216MB
httpd               latest    157dcdf23d6c   6 days ago       145MB
eclipse-mosquitto   latest    09ef04ba0be0   4 weeks ago      11.9MB
```

Instalação

Instalação

1. Clone o repositório
2. Construa o container do broker

Crie a imagem do broker e suba o container

```
docker compose -f giotgrad09/backend/broker/docker-compose-mosquitto.yaml up -d --remove-orphans
```

```
docker exec -it mqtt /bin/sh
```

```
mosquitto_passwd -U mosquitto/config/password.txt
```

```
vi mosquitto/config/mosquitto.conf
```

Descomente a linha 15 do arquivo `mosquitto.conf` e salve as alterações. (INSERT -> descomente -> ESC -> :wq)

```
exit
```

Restart o container

```
docker restart mqtt
```

3. Sete informações de rede e credenciais no arquivo `giotgrad09/backend/esp32-ino/esp32-temp-hum.ino`

Se estiver na conexão da eduroam, deixe a variável shouldConnectToEduroam = true, do contrário, deixe false e sete as informações da rede local em HOME_WIFI_SSID e HOME_WIFI_PASSWORD. Se estiver na eduroam: Sete o define EDUROAM_EAP_IDENTITY com seu número USP Sete o define EDUROAM_EAP_PASSWORD com sua senha única

4. Descarregue o código no ESP32
5. Crie os outros containers

```
docker compose -f giotgrad09/backend/node-api/docker-compose-mosquitto.yaml up -d --remove-orphans
```

```
docker compose -f giotgrad09/backend/apache/docker-compose-apache up -d --remove-orphans
```

6. Acesse http://andromeda.lasdpc.icmc.usp.br:8423/
7. Utilize o login "admin@admin" e senha "12345Admin*" para se autenticar na interface