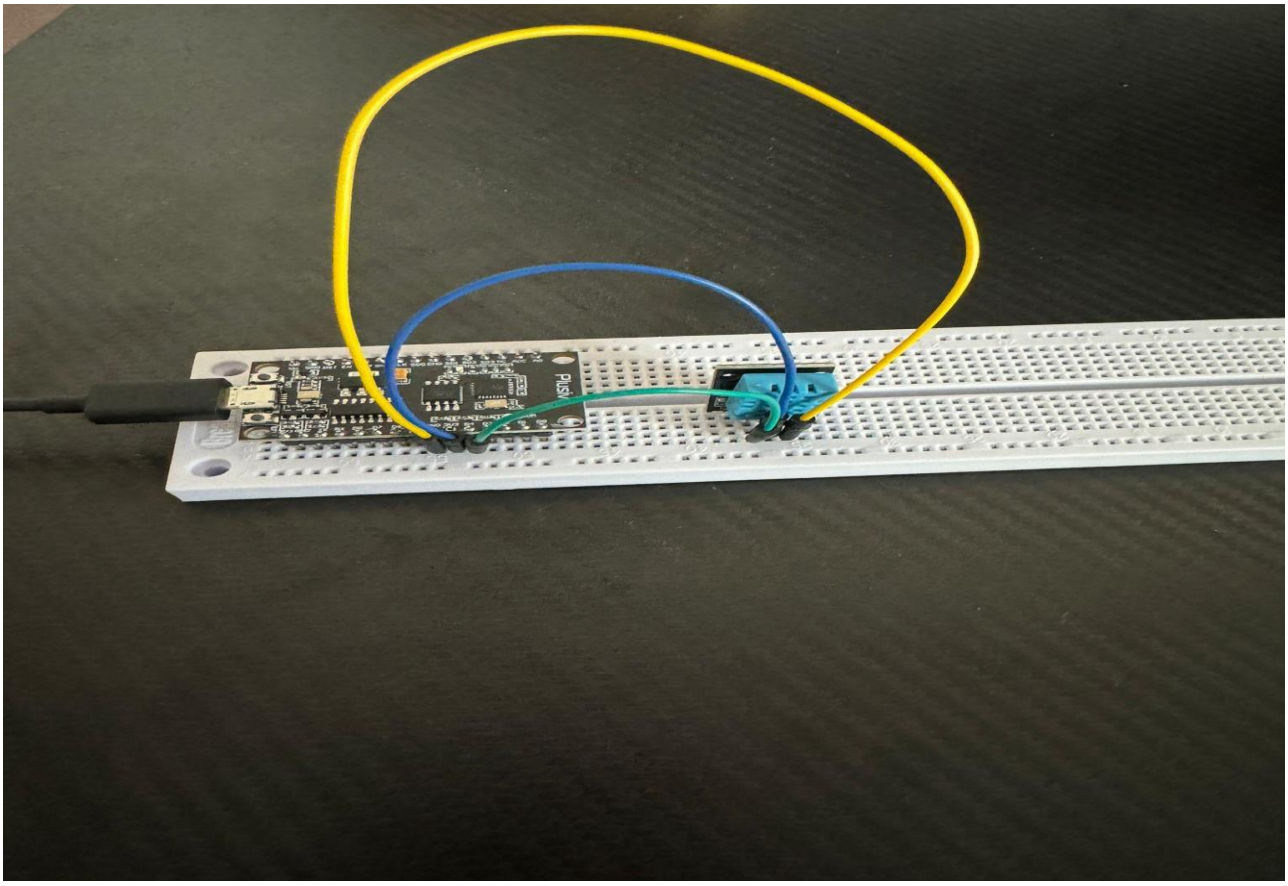
PRE-PRACTICA TASK

- Beia Consult International -

Author: Văduva Matias

My pre-practice project for Beia Consult International is a temperature and humidity IoT using a DHT11 wired to a Plusivo Micro board that is compatible with the Arduino ESP8266 board add-on.

This is how my project looks :



I have used a breadboard to easily make the circuit and created the connections using wires.

NodeMCU 1.0 (ESP-12...

sketch_jun25a.ino

```
#include <DHT11.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>

#define DEVICE "ESP8266"
#define INFLUXDB_URL "http://[redacted]:8086"
#define INFLUXDB_TOKEN "tlX2VfwqN8ptqQV2LDGDxIF0RGamZjSyVI82F8ew8PGHC_EH2ZaEK3hznqKxh_tJPqoQTCPmoKj2MRr_AfT96Q=="
#define INFLUXDB_ORG "35d9c13a2d7a42e2"
#define INFLUXDB_BUCKET "beia"
#define INFLUXDB_MEASUREMENT "sensor_data"
#define TZ_INFO "EET-2EEST,M3.5.0/3,M10.5.0/4"


InfluxDBClient Client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);
Point sensor("Temp-Hum");

const char* wifi_ssid = "DIGI-ft6c";
const char* wifi_password = "SuEUtY4hts";

const char* mqtt_server = "mqtt.beia-telemetrie.ro";
const char* mqtt_topic = "/training/device/matias-vaduva/data";

WiFiClient espClient;
PubSubClient client(espClient);

DHT11 dht11(D4);

void setup() {
```

Output Serial Monitor ×

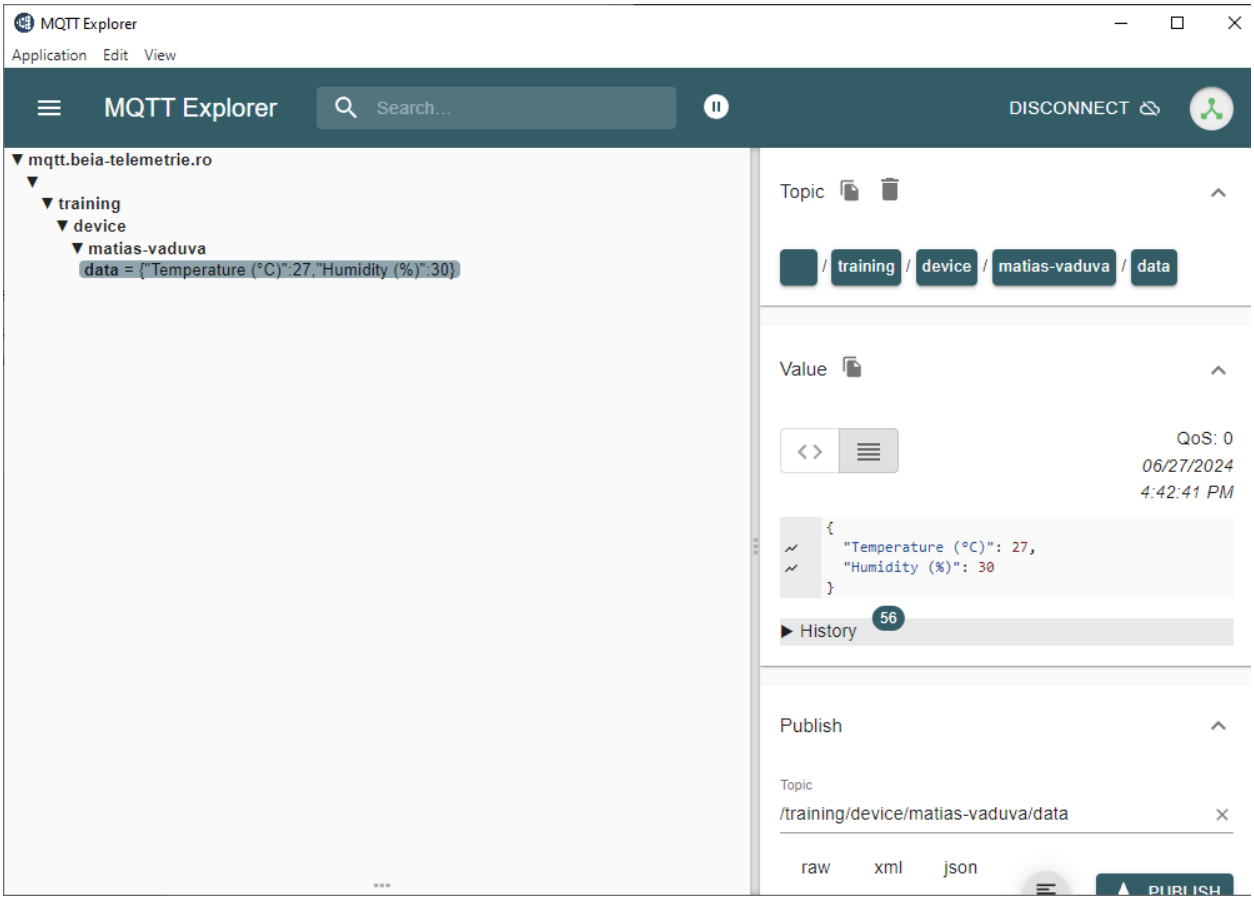
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3')

```
.Wifi - connected
Syncing time.
Synchronized time: Thu Jun 27 14:17:18 2024

Connected to InfluxDB: http[redacted]:8086
Connected to MQTT Broker!

The temperature is : 28°C

The humidity in the air is : 38 %
```

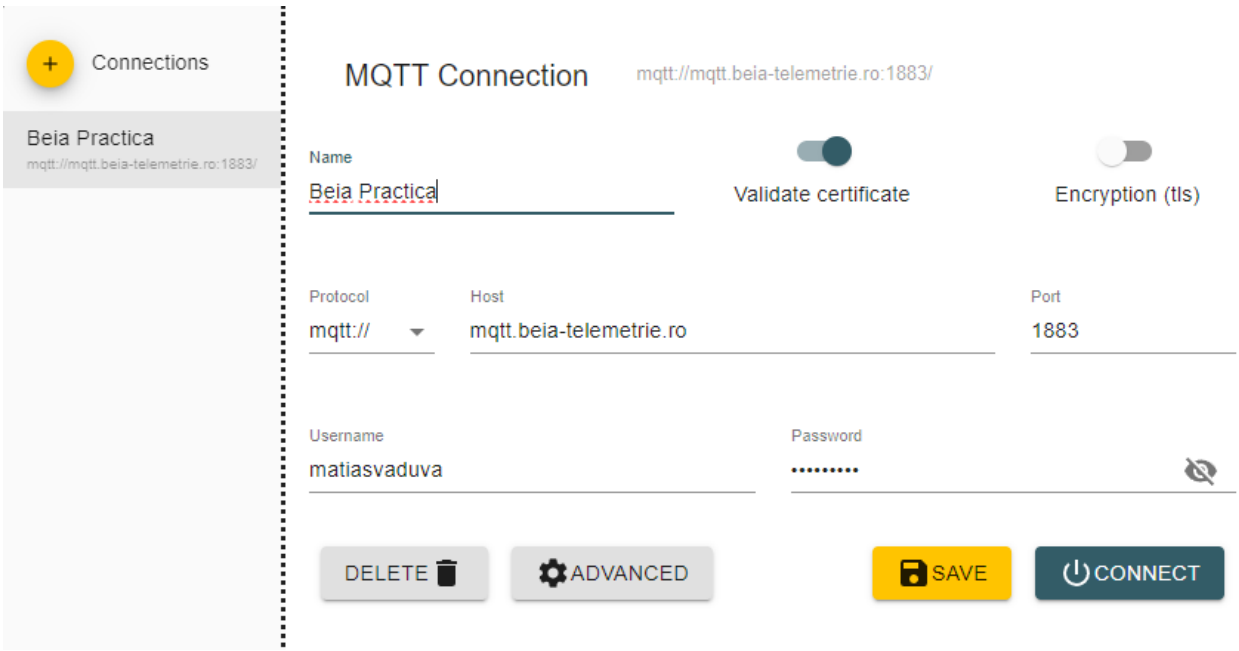
The DHT11 sensor is a bit faulty, it was firstly used for a project at my faculty and it ocassionaly doesn't read the temperature and humidity values.

As you can see, I connected my device to the WiFi and also the MQTT Broker + InfluxDB database (I will get to this later on in the presentation) and it sends accurate readings. The full code will be posted on my github.

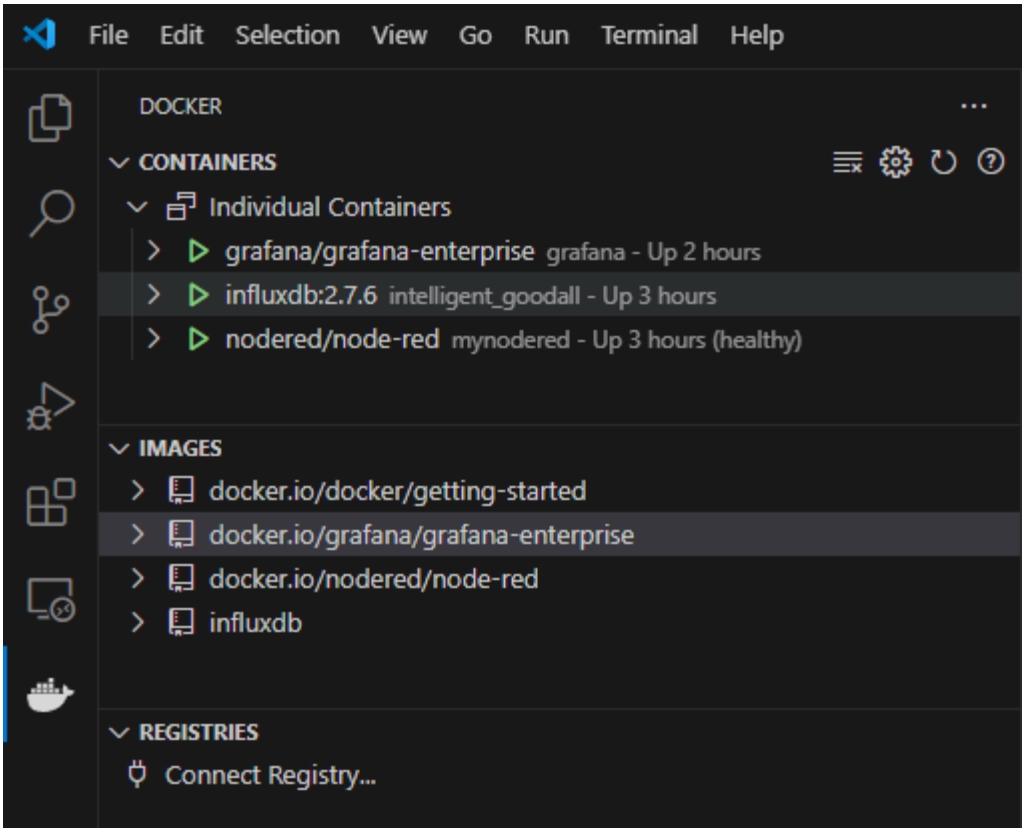
For the MQTT, I used MQTT EXPLORER where I structured the topics. It reads the data from my sensor once every 5 seconds.



Of course, I used the Beia server for this to happen

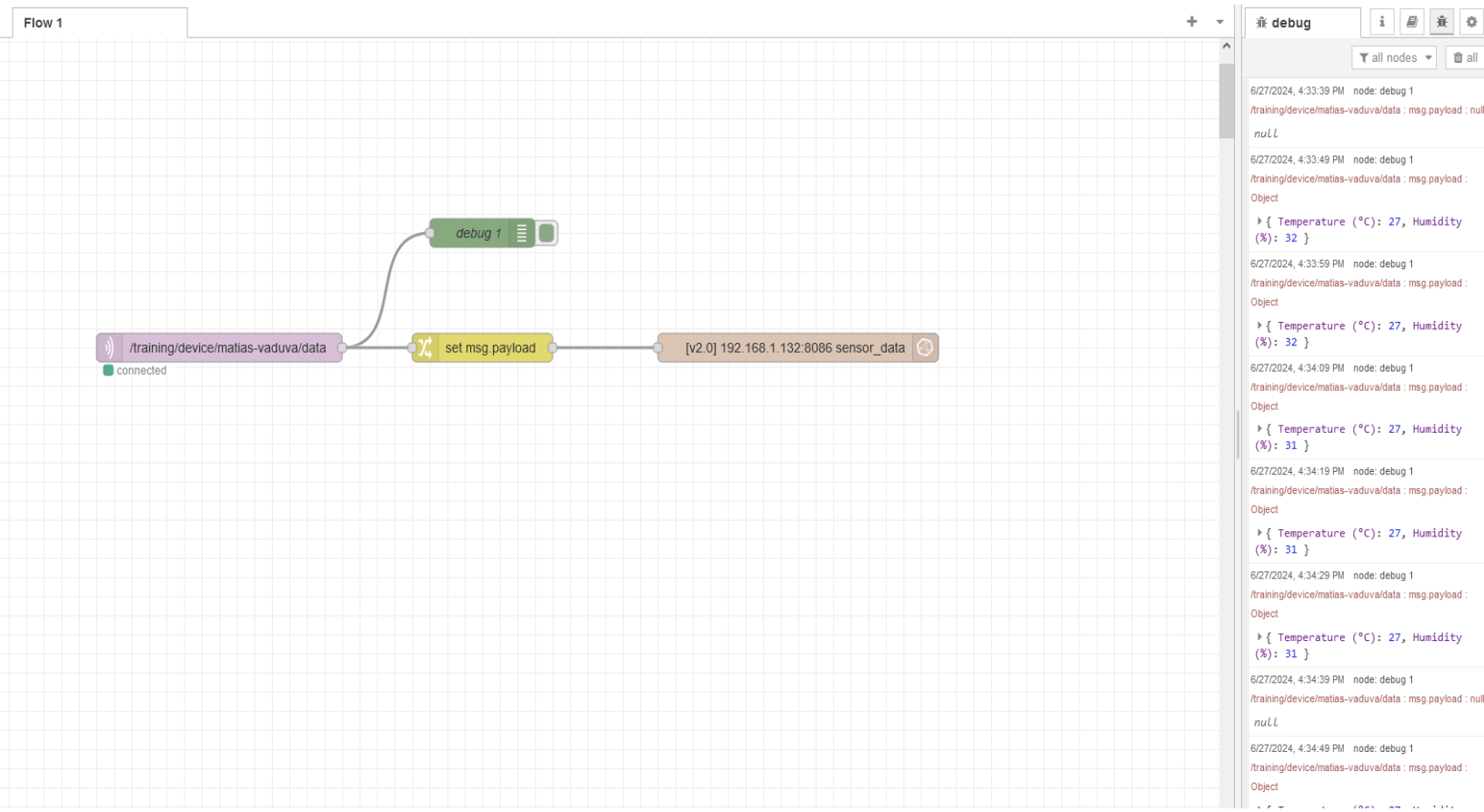


Next, I implemented docker and containers using Visual Studio Code, since I am operating on Windows 10.



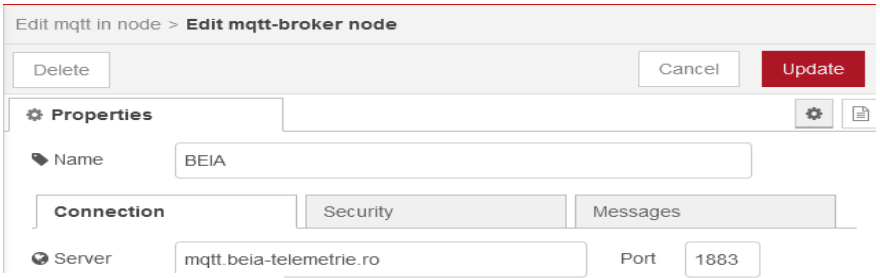
The containers are for Node-RED, Influxdb and Grafana, and they are all working without problems.

Next, I created a flow in Node-RED and subscribed it to my topic from the MQTT server using a broker node.



This is the full picture, including all the nodes and the temperature readings

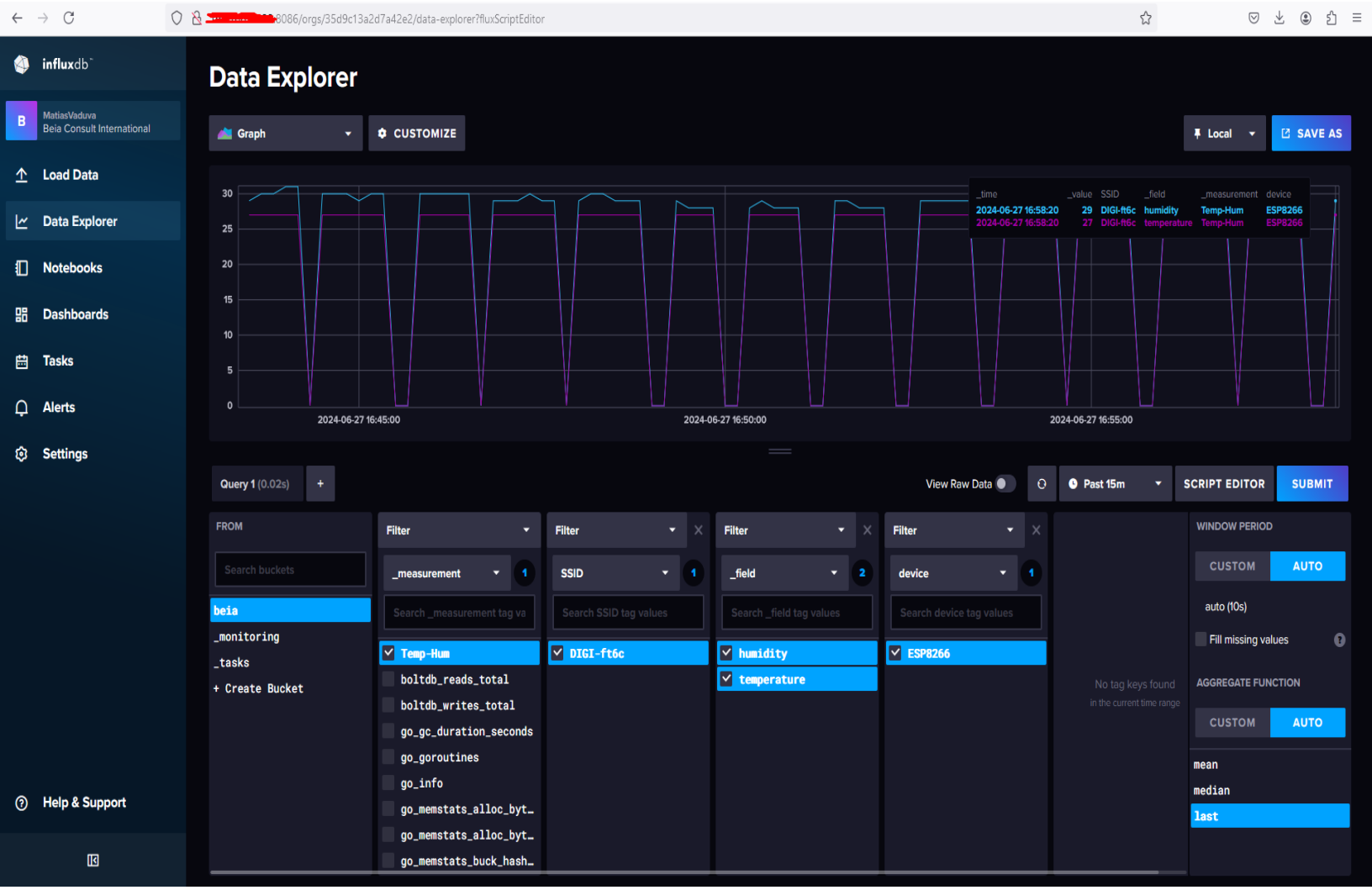
Also, this is the configuration node, and the of the MQTT data.




```
{
    "temperature" : msg.payload.t,
    "humidity" : msg.payload.h
}
```

For the InfluxDB Cloud database, I firstly added the requirements in my code, the site url, token etc. The rest is in the setup and loop functions. The site is locally accessible, using the ipv4 address with the port “8086” at the end.

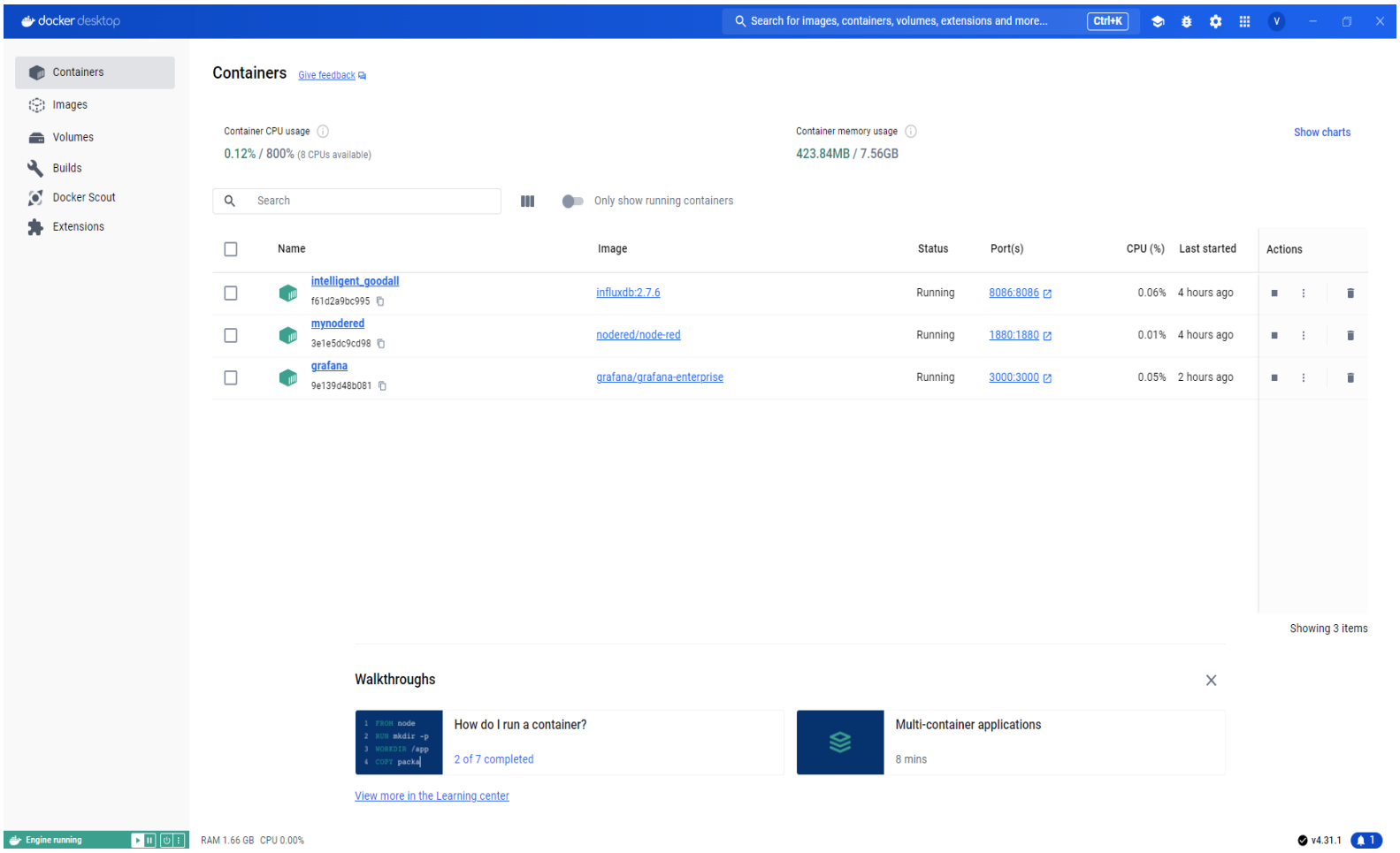
```
#define DEVICE "ESP8266"
#define INFLUXDB_URL "http://[redacted]:8086"
#define INFLUXDB_TOKEN "tlX2VfwqN8ptqQV2LDGDxIF0RGamZjSyVI82F8ew8PGHC_EH2ZaEK3hznqKxh_tJPqoQTCPmoKj2MRr_AfT96Q=="
#define INFLUXDB_ORG "35d9c13a2d7a42e2"
#define INFLUXDB_BUCKET "beia"
#define INFLUXDB_MEASUREMENT "sensor_data"
#define TZ_INFO "EET-2EEST,M3.5.0/3,M10.5.0/4"


InfluxDBClient Client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);
Point sensor("Temp-Hum");
```



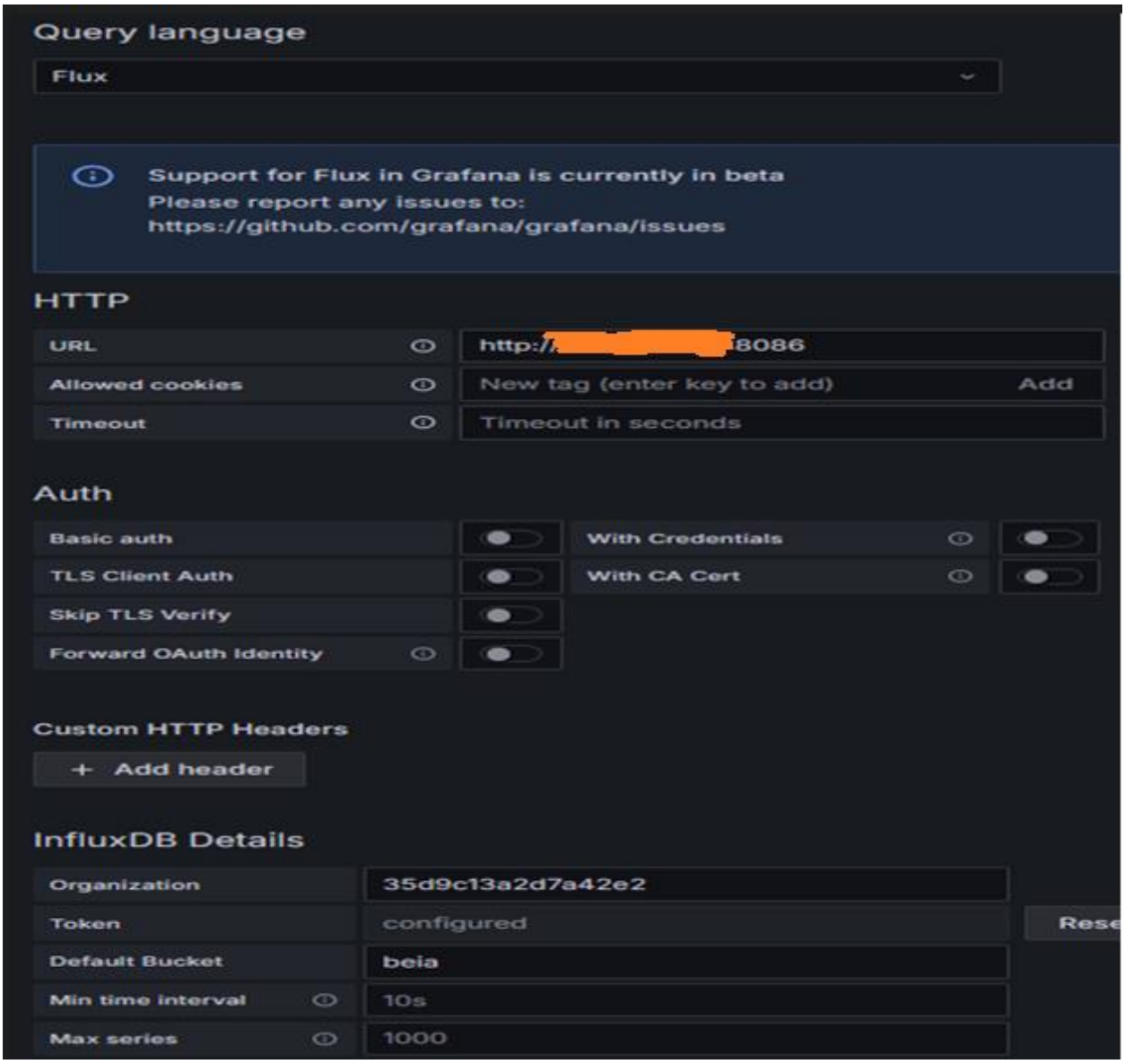
And these are the temperature and humidity readings, the 0 values are because of the sensor’s occasional missreadings.

I also created an alert for when the temperature exceeds the 35 °C threshold.

And these are all the containers also on the docker desktop application

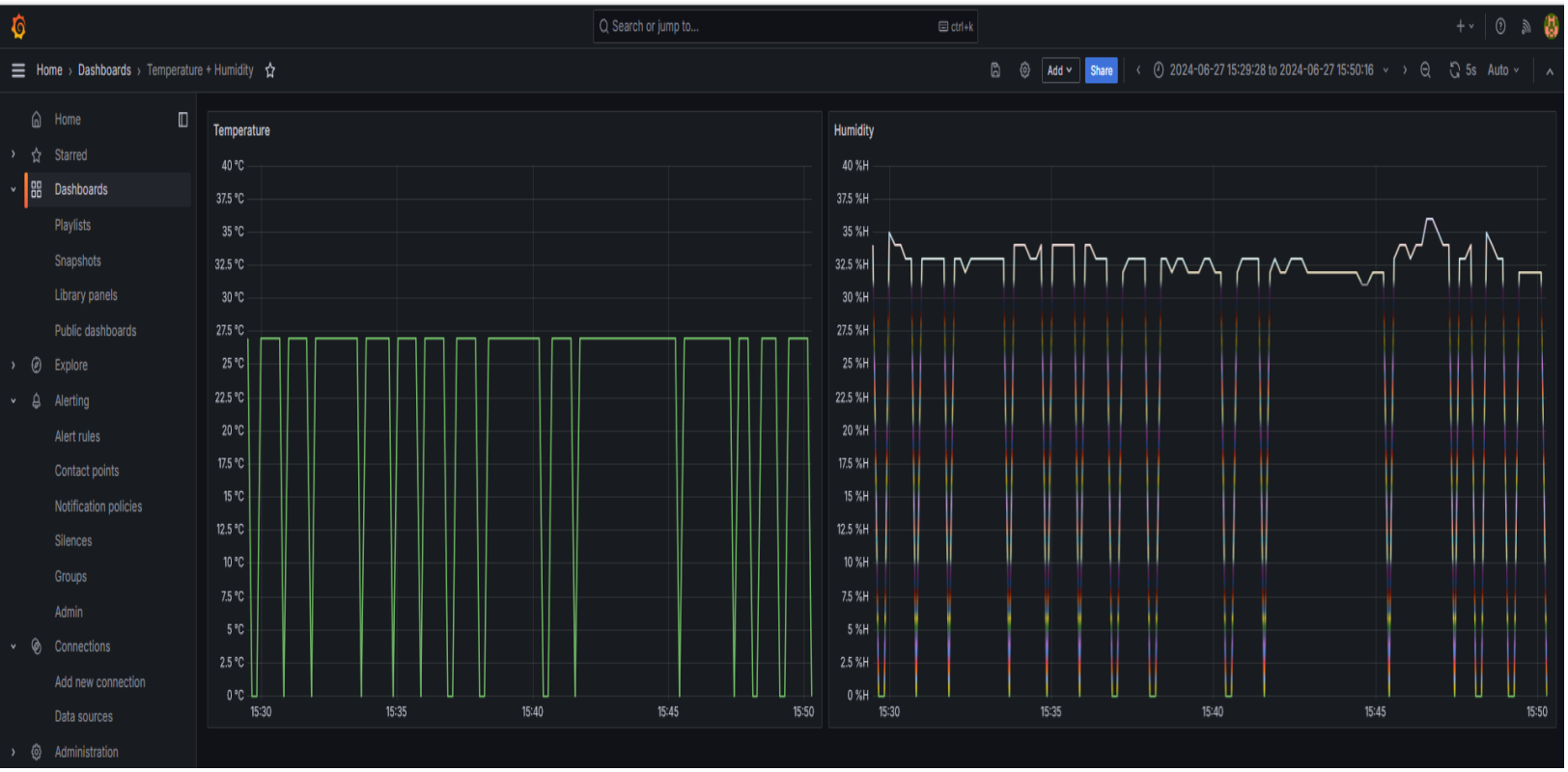


For Grafana, I couldn’t create a Data Source for the beia account so I created my own local instance of it and imported the data.



It uses the influxdb token and bucket I created for the program, and also the site URL.

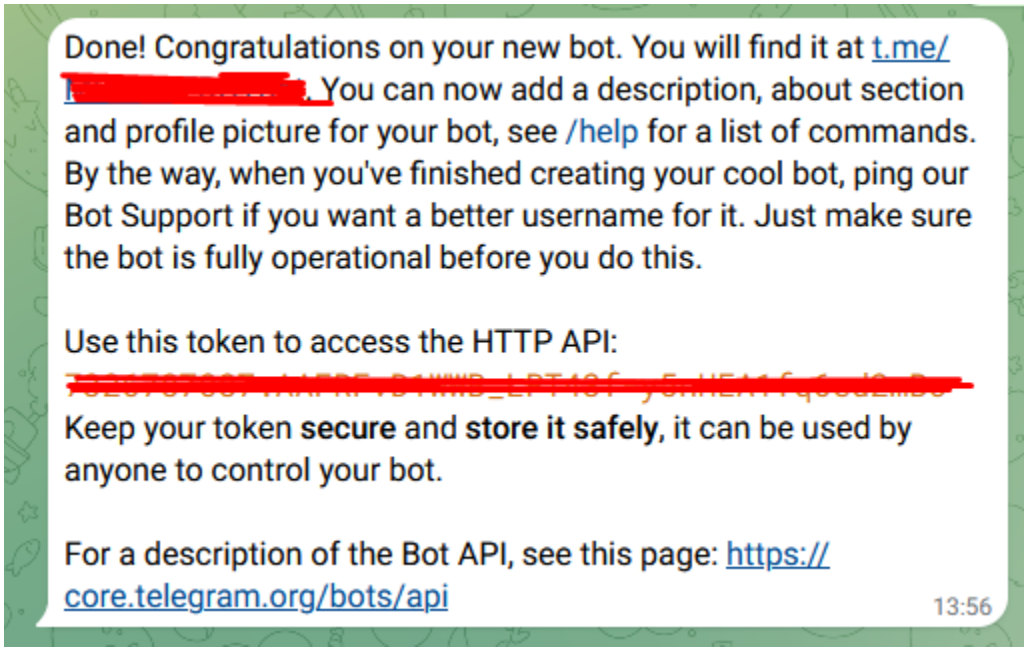
These are the graphs for the temperature and the humidity!

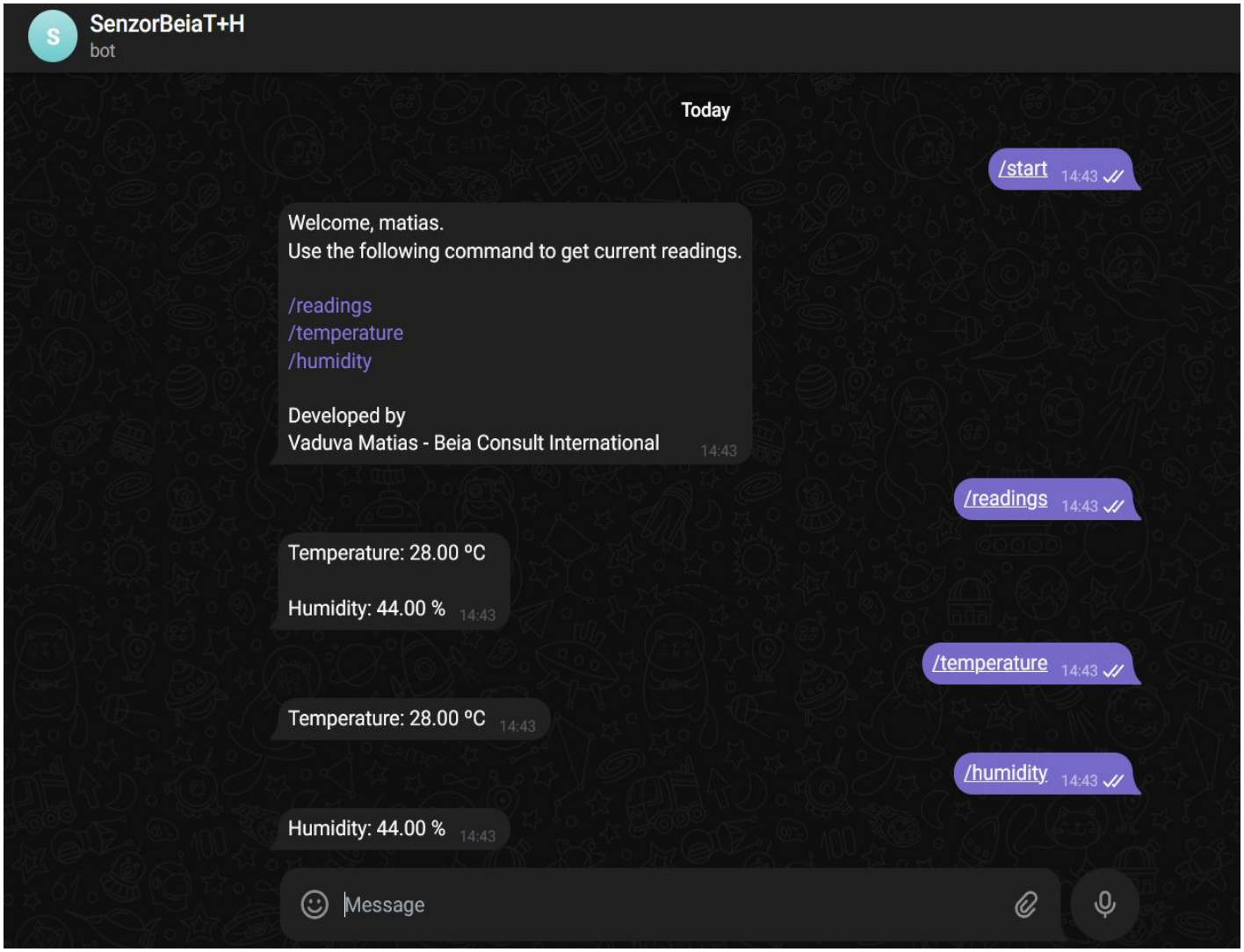


The next tasks are the chatbot and blockchain technology.

For the chatbot, I decided to create a Telegram Bot, it reads the temperature and humidity from my sensor and displays it in the chat. The code is written in ARDUINO.

Telegram gave me an API token and I used it with the chat id in the code to connect my system.



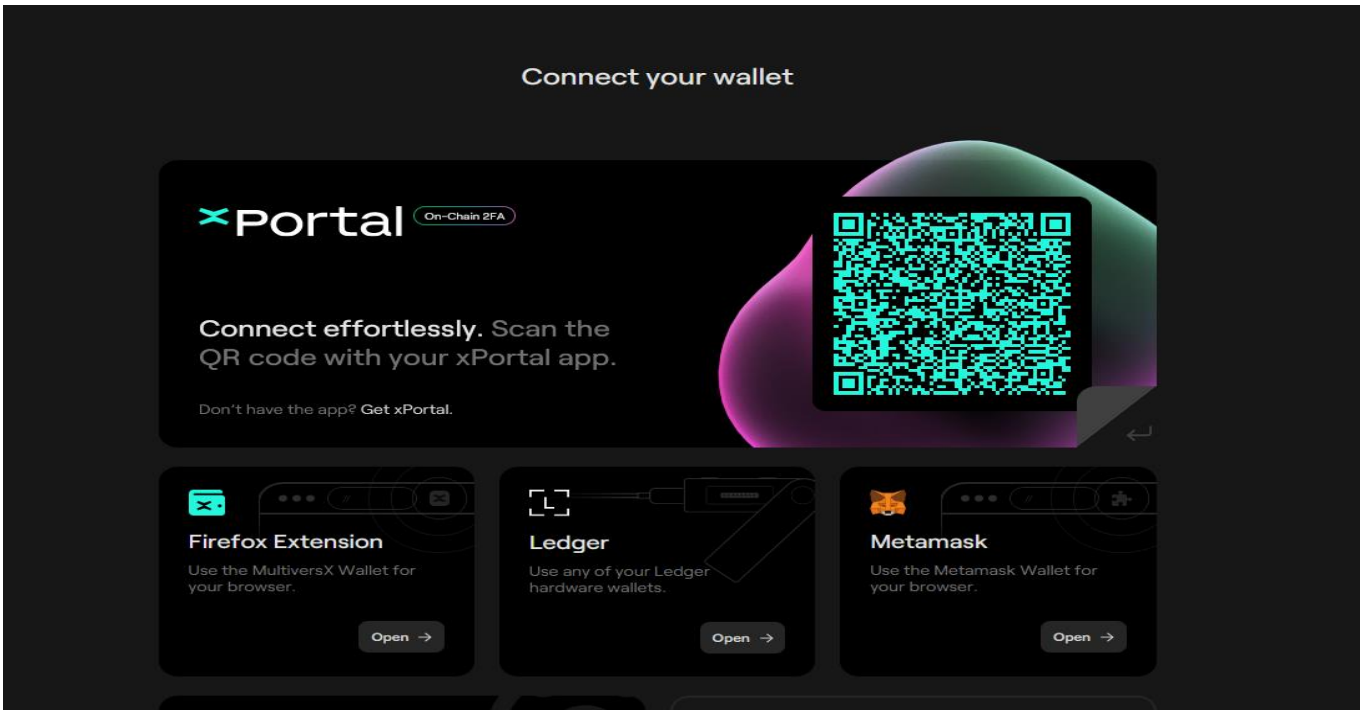


As you can see, I configured the bot to run with the /start command. I also added a little “welcome” message where you can see the available commands.

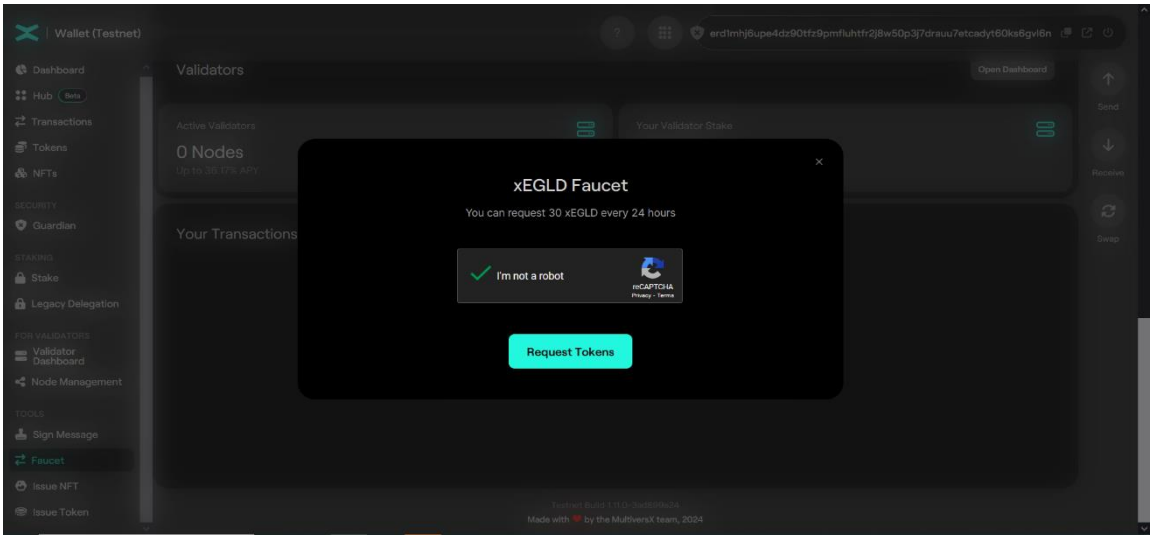
If you type /readings, it will show both the temperature and humidity values, and I also made it possible to access them separately with their corresponding names.

For the Blockchain technology, we had some help from one of the senior colleagues. lockchain technology is an innovative framework that has captured widespread interest in the digital age.

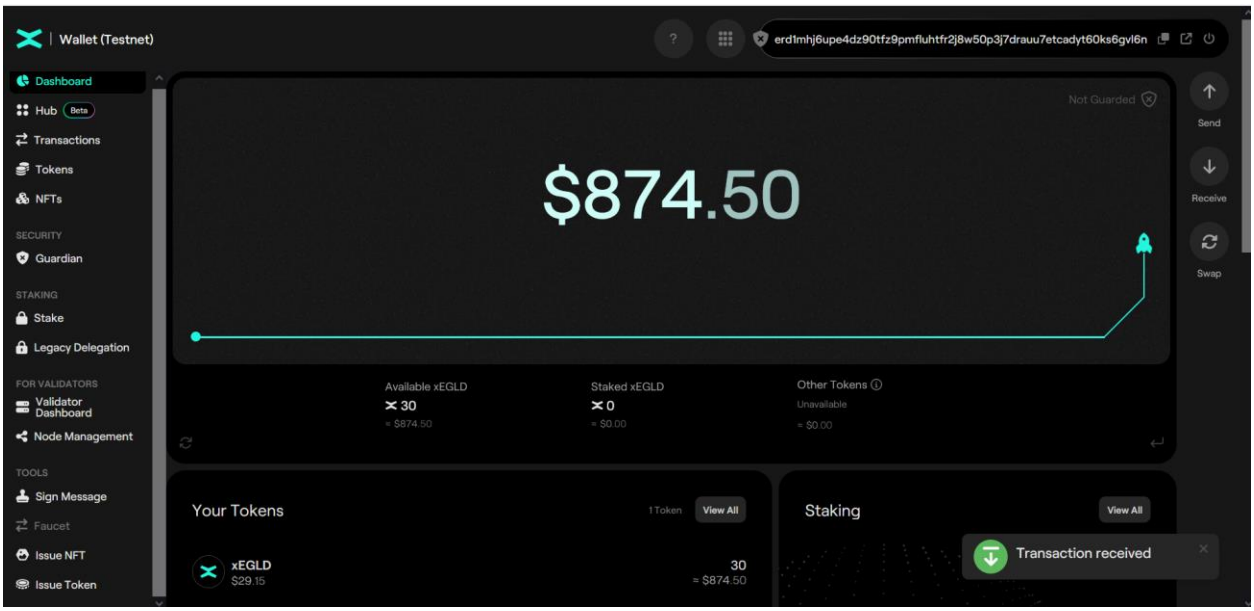
Firstly, I downloaded X portal on my phone and made an account.



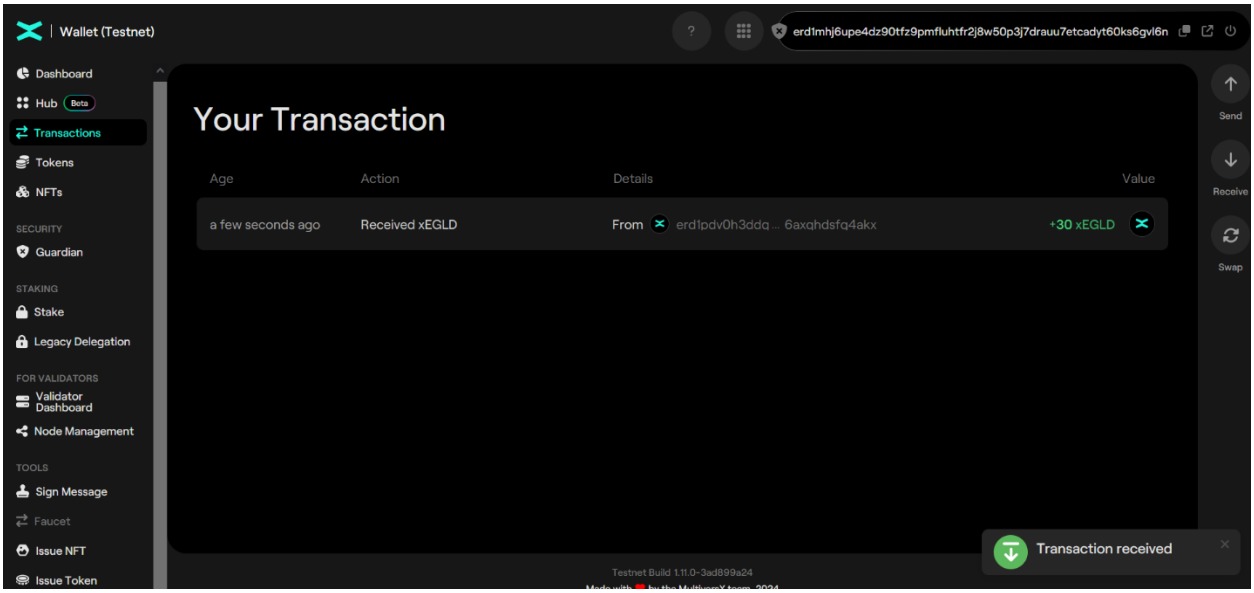
The website used is https://testnet-wallet.multiversx.com/unlock.



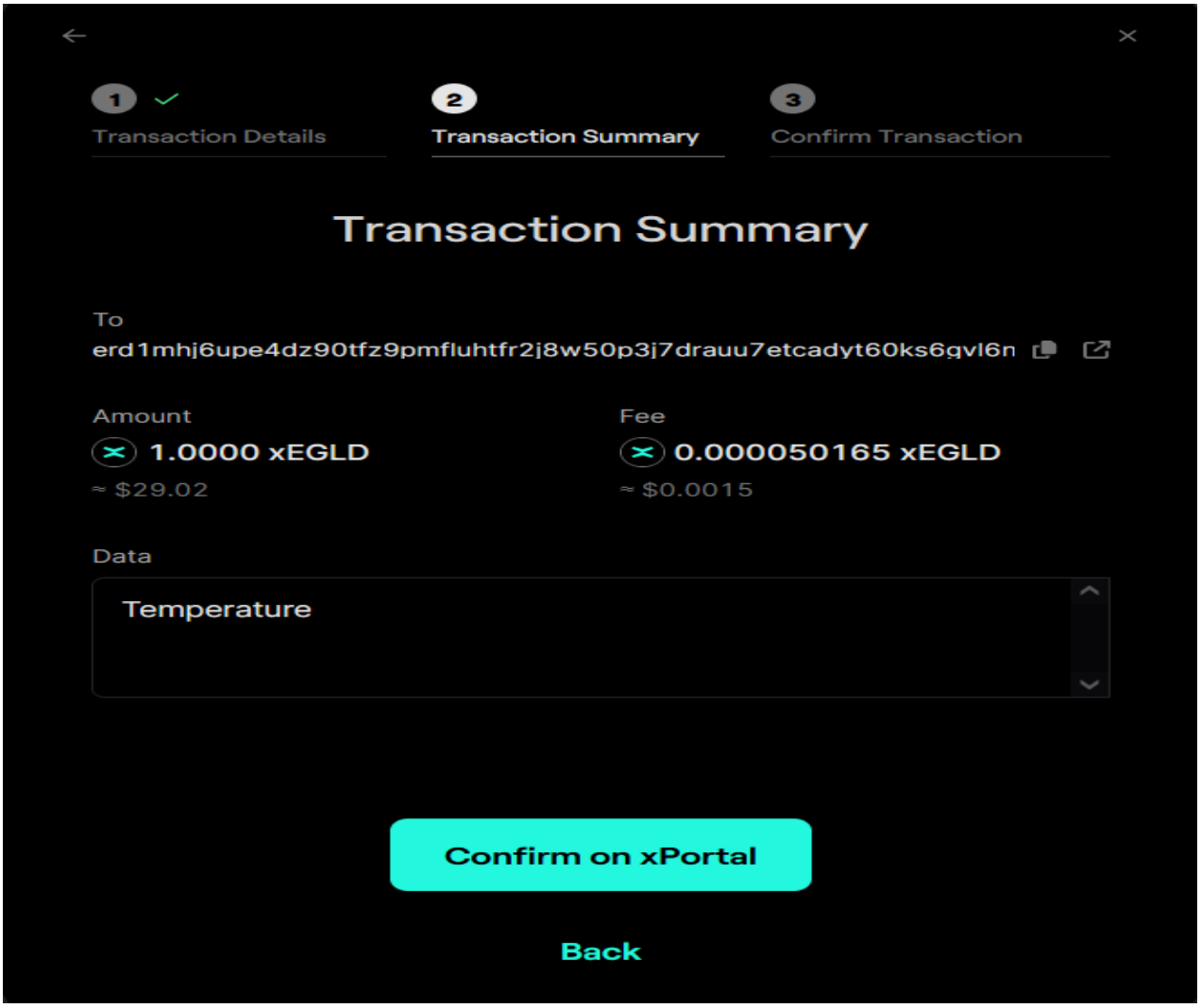
Next, I used the Faucet option and I can generate 30 xEGLD currency every 24 hours.



It transformed the currency in a more commonly used ones, in this case the American Dollar.



I then made a transaction to myself, the data sent being the sensor reading which could also be automated.



This is the link to my github with the Arduino code :

https://github.com/MatiasVaduva/Cod-Pre-Practica