INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Educação

INTERNET DAS COISAS

TP3 - Aplicações para manipulação e armazenamento de dados

- Ano Letivo 2022/2023 –

A48530 – Diogo Brandão Ferreira
A48542 – José Nuno Marinho Carvalho

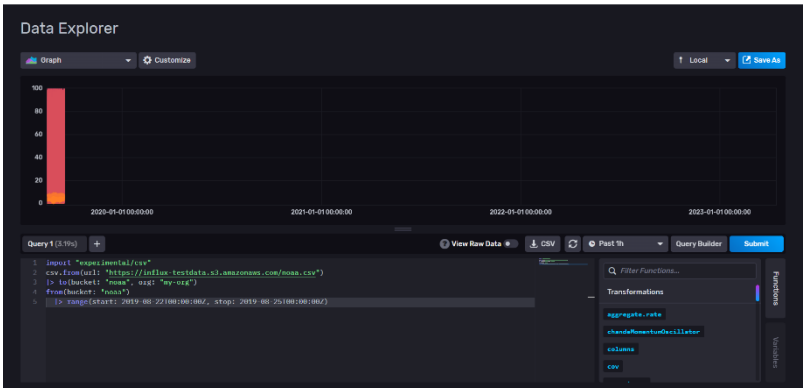
2. (i) O InfluxDB é uma base de dados de séries temporais (time-series database), em open source. Foi criado para armazenar, consultar e processar grandes volumes de dados de séries temporais em tempo real. É amplamente utilizado em aplicações IoT, monitoramento de infraestruturas e análise de dados de negócios.

(ii) Time-series database, ou base de dados de séries temporais, é um tipo de base de dados que é otimizado para armazenar e consultar dados que mudam ao longo do tempo. Esse tipo de banco de dados é perfeito para armazenar dados de sensores, eventos de log, dados de aplicativos de monitoramento e outras informações que precisam ser analisadas em relação ao tempo.

(iii) Flux é uma linguagem de consulta e processamento de dados, projetada para trabalhar com dados de séries temporais no banco de dados InfluxDB. É uma linguagem declarativa que permite que os usuários definam fluxos de dados que podem ser executados em tempo real ou em lotes. Ele suporta operações como filtragem, agregação, junção e transformação de dados. É altamente extensível e pode ser integrado com outras ferramentas de análise de dados e visualização.

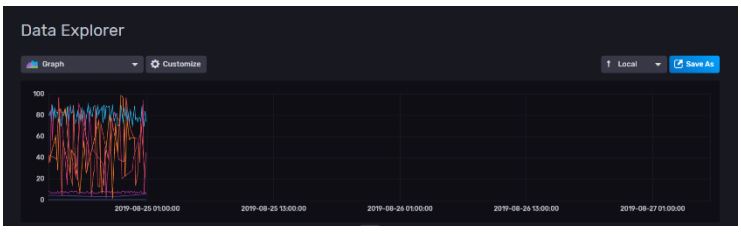
3.c)

```
from(bucket: "noaa")
  |> range(start: 2019-08-22T00:00:00Z, stop: 2019-08-25T00:00:00Z)
```



3.d)

```
from(bucket: "noaa")
  |> range(start: 2019-08-22T00:00:00Z, stop: 2019-08-25T00:00:00Z)
  |> filter(fn: (r) => r.location == "santa_monica")
```

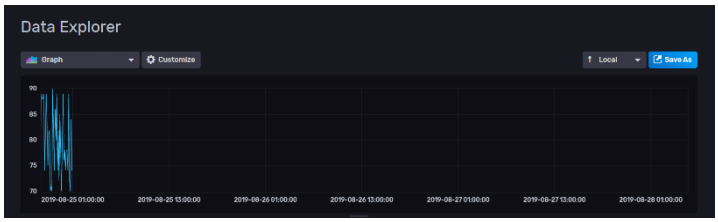


3.e) from(bucket: "noaa")

|> range(start: 2019-08-22T00:00:00Z, stop: 2019-08-25T00:00:00Z)

|> filter(fn: (r) => r.location == "santa_monica")

|> filter(fn: (r) => r._measurement == "average_temperature")



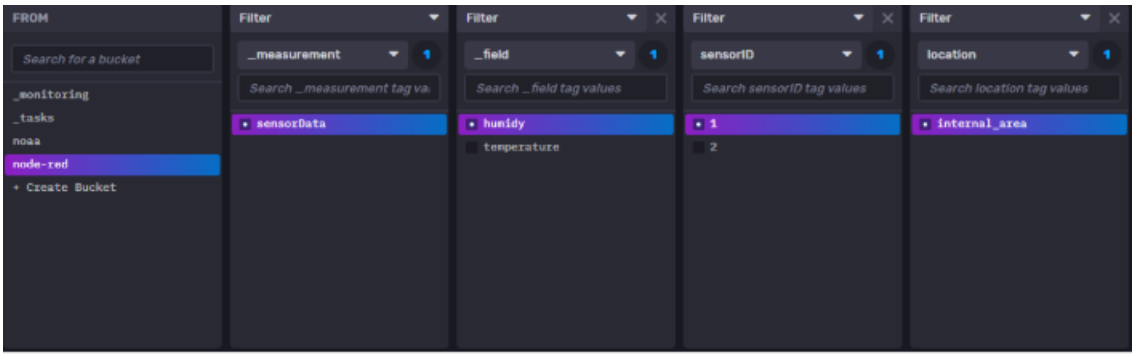
3.f) from(bucket: "noaa")

|> range(start: 2019-08-22T00:00:00Z, stop: 2019-08-25T00:00:00Z)

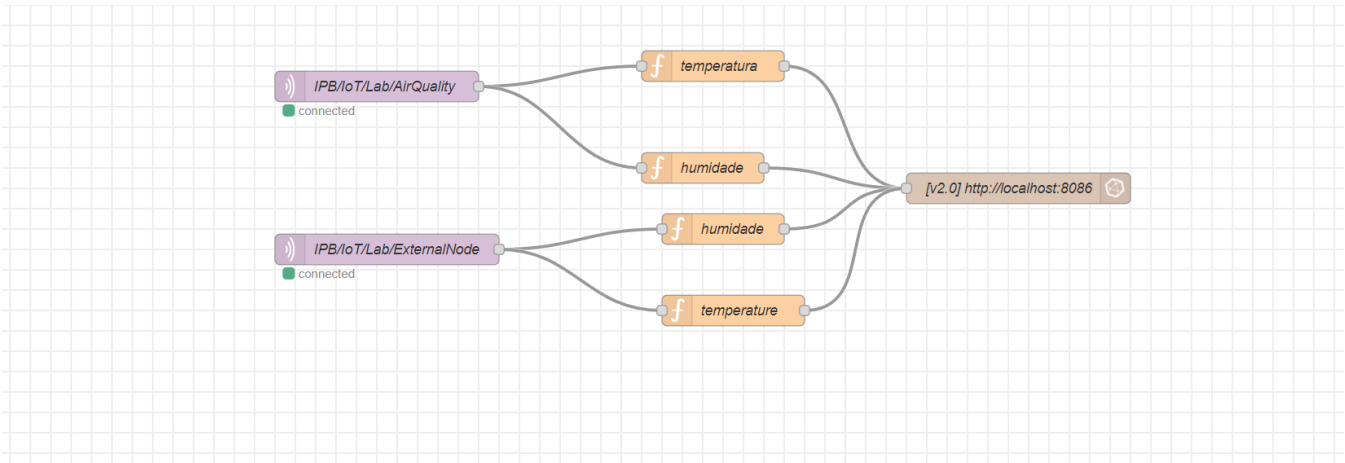
|> filter(fn: (r) => r._measurement == "average_temperature" and r.location == "santa_monica")

|> aggregateWindow(every: 1h, fn: max)

7.)



(Filtros Node-Red)



```
msg.payload = [
{
measurement: "sensorData",
fields: {
temperature:msg.payload.temp,
},
tags: {
sensorID: 1,
location: "internal_area"
}
}
];
return msg;
```


```
msg.payload = [
{
measurement: "sensorData",
fields: {
humidy:msg.payload.hum,
},
tags: {
sensorID: 2,
location: "external_area"
}
}
];
return msg ;
```

8.) No InfluxDB UI, crie uma dashboard contendo:

Gráfico que mostre os valores de temperatura

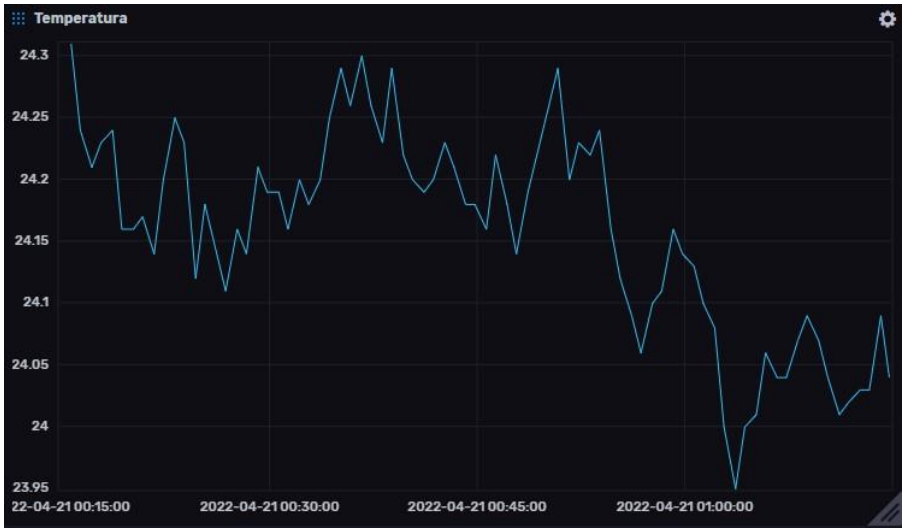


Gráfico que mostre os valores de humidade.

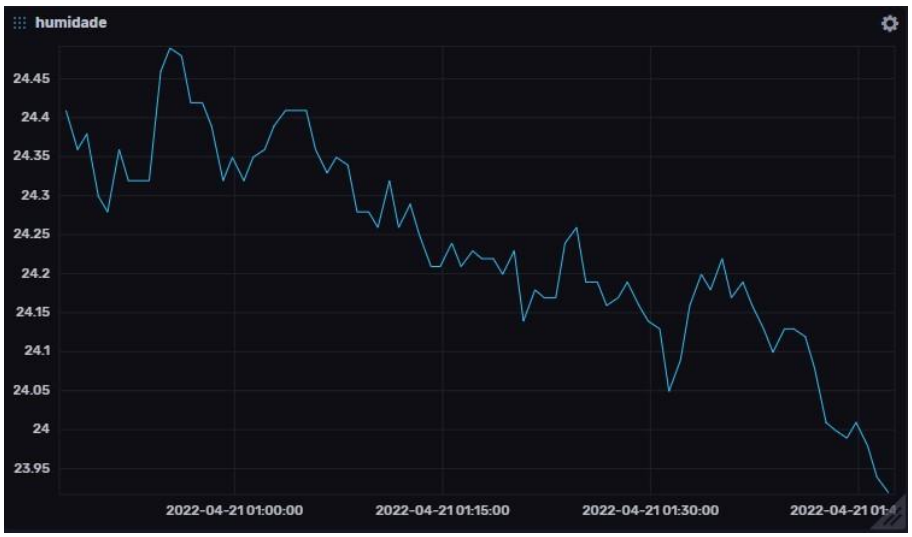
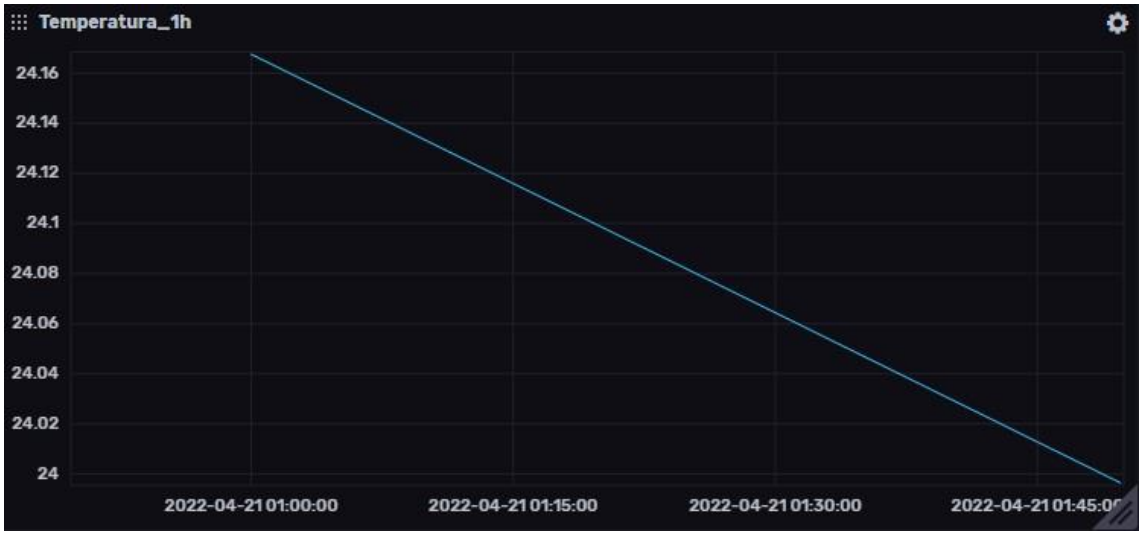


Gráfico que mostre a média de humidade a cada uma hora.

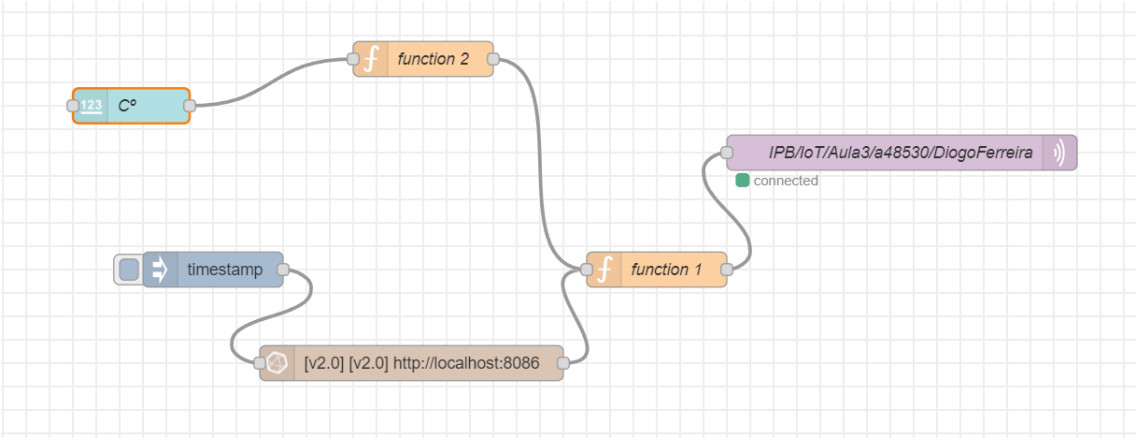


Single stats que mostrem os valores máximos e mínimos para temperatura e humidade da última uma hora.

MAX,MIN DE TEMPERATURA:

MAX,MIN DE HUMIDADE

9. Deseja-se fazer o controlo automático de uma janela com base na temperatura ambiente média da última uma hora e a temperatura selecionada por um usuário. Para isso,considere os dados de temperatura armazenados no InfluxDB (realizado nos exercícios anteriores).



Function 1:

```
if (msg.payload && msg.payload.length > 0 && msg.payload[0]._value !== undefined) {
    var meanValue = msg.payload[0]._value;
    var userValue = flow.get("userInput") || 0;
    var newMsg;
    if (meanValue <= userValue) newMsg = "Janela Aberta";
    else newMsg = "Janela Fechada";
    msg.payload = newMsg;
} else {
   msg.payload = "Error: Please wait 30 sec ";
}
return msg;
```

Function 2:

```
flow.set('userInput', msg.payload);
return msg;
```

InfluxDB:

```
from(bucket: "red-noaa")
|> range(start:-1h, stop: now())
|> filter(fn: (r) => r["_measurement"] == "sensorData")
|> filter(fn: (r) => r["_field"] == "humidy" or r["_field"] == "temperature")
|> filter(fn: (r) => r["location"] == "internal_area")
|> filter(fn: (r) => r["sensorID"] == "1")
|> aggregateWindow(every: 1h, fn: median, createEmpty: false)
|> yield(name: "median")
```