

## LAB 1, 2 e 3 - IOT

Nome: João Marcos de Almeida Senna

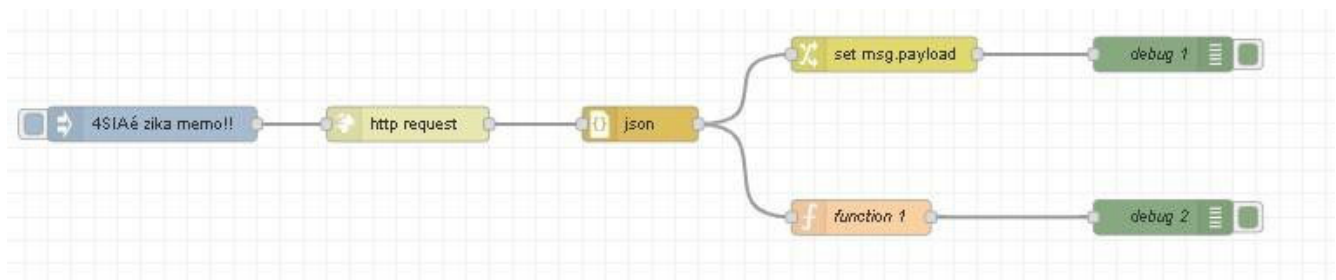
RM: 82832

Sala: 4SIA

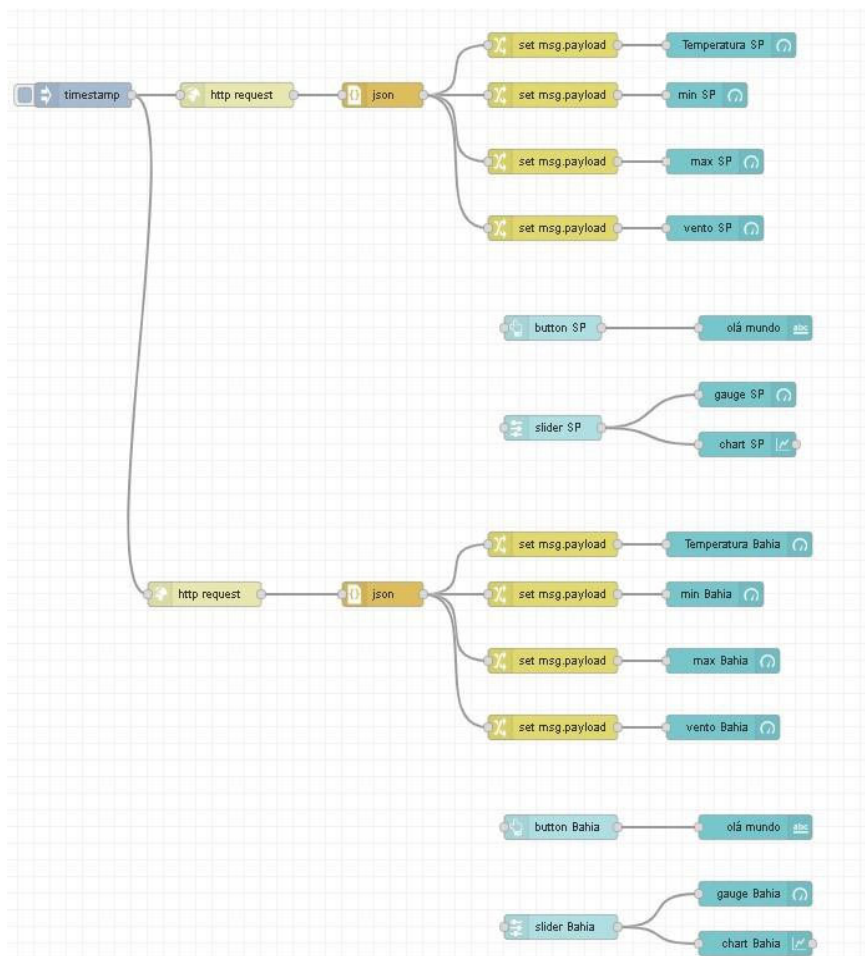
Link dos flows:

[https://drive.google.com/file/d/1ngG0kT0fJ0oZZcGG\\_31KCrIHE3Qy1K88/view?usp=sharing](https://drive.google.com/file/d/1ngG0kT0fJ0oZZcGG_31KCrIHE3Qy1K88/view?usp=sharing)


Desafio 1:



Desafio 2:




### Desafio 3:

**HIVEMQ**  
CLOUD

Need a fully managed MQTT broker?  
Get your own Cloud broker and connect up to 100 devices for free.

Get your free account

---

**Connection** connected 

Host  
mqtt-dashboard.com

Port  
8884

ClientID  
clientId-W5GR9f6rc6

Disconnect

Username

Password

Keep Alive  
60

SSL  
x

Clean Session  
x

Last-Will Topic

Last-Will QoS  
0

Last-Will Retain  
☐

Last-Will Message

**Publish** 

Topic  
arnaldoAVianaJr/fiap/teste2

QoS  
0

Retain  
☐

Publish

Message  
Olá mundo!!**Subscriptions** 

Add New Topic Subscription

Qos: 0

arnaldoAVianaJr/fiap...

x

**Messages** 

2023-08-16 19:19:48    Topic: arnaldoAVianaJr/fiap/teste2    Qos: 0

Olá mundo!!

### Desafio 4:

#### 1) QoS (Qualidade de Serviço) em mensagens MQTT:

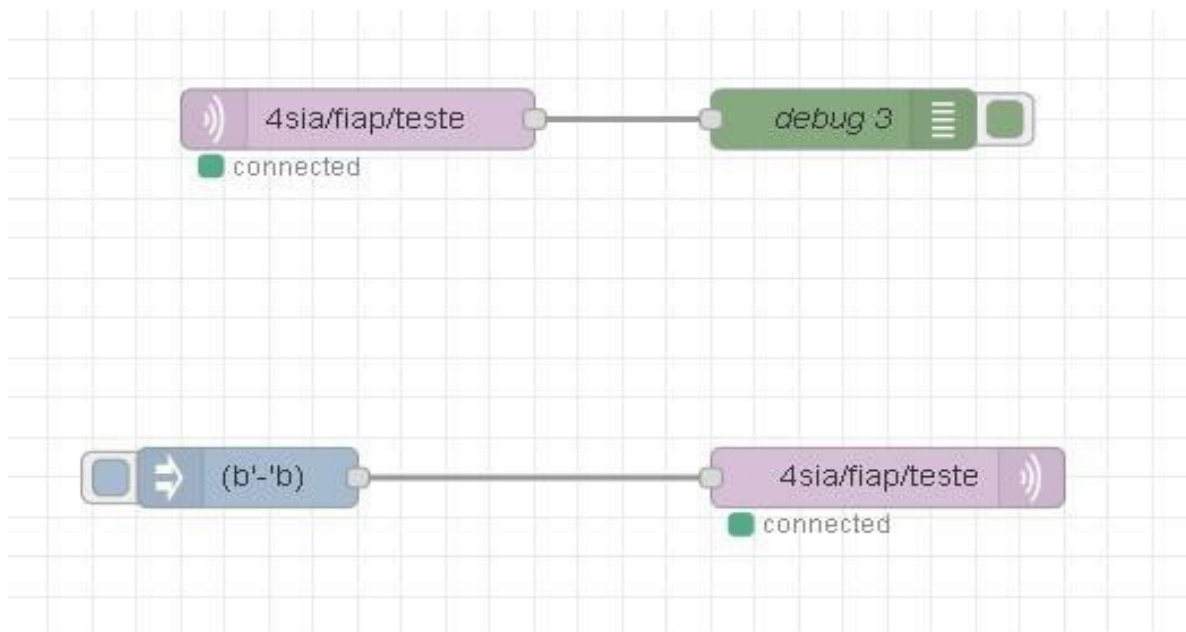
- **QoS 0 (Qualidade de Serviço 0):** Nesse nível, as mensagens são entregues somente uma vez, sem confirmações ou verificações. Não há garantias de entrega, e retransmissões não ocorrem. Esse nível é adequado para cenários onde a perda de mensagens não é crítica, como atualizações de status substituíveis por novas mensagens.
- **QoS 1 (Qualidade de Serviço 1):** Neste nível, a mensagem é entregue pelo menos uma vez. O cliente que recebe a mensagem envia um pacote de confirmação (PUBACK) ao servidor, indicando a recepção. Se o servidor não receber a confirmação, ele reenviará a mensagem. Isso assegura pelo menos uma entrega da mensagem, podendo resultar em duplicação se houver falha na confirmação.

- QoS 2 (Qualidade de Serviço 2): Nesse nível, a mensagem é garantida para ser entregue exatamente uma vez, sem duplicações. O processo envolve quatro etapas: o cliente envia um pacote de confirmação (PUBREC), o servidor envia um pacote de liberação (PUBREL), o cliente responde com um pacote de confirmação (PUBCOMP) e, por fim, o servidor envia uma confirmação de entrega ao cliente. Isso assegura a entrega única da mensagem, mesmo em situações de falha.

## 2) Caracteres coringa na subscrição de tópicos:

- a) Níveis múltiplos (multi-level) com '#': O caractere '#' é usado como coringa para corresponder a qualquer número de níveis em um tópico. Por exemplo, ao se inscrever em "casa/quarto/#", você receberá mensagens dos tópicos "casa/quarto/escritinha", "casa/quarto/cama", "casa/quarto/janela", etc. Isso permite inscrever-se em hierarquias completas de tópicos.
- b) Nível simples (single-level) com '+': O caractere '+' é usado como coringa para corresponder a um único nível em um tópico. Se inscrevendo em "casa/quarto/", você receberá mensagens de tópicos como "casa/quarto/escritinha" e "casa/quarto/cama", mas não de "casa/quarto/escritinha/luz" ou "casa/quarto/cama/travesseiro".

### Desafio 5:



Nome: João Marcos de Almeida Senna

RM: 82832

Sala: 4SIA

**Link dos flows:**

**Código usado no arduíno:**

[https://drive.google.com/file/d/1Fwpf7QpzVI51vIQbTO2K1b70BnpOYds2/view?usp=drive\\_link](https://drive.google.com/file/d/1Fwpf7QpzVI51vIQbTO2K1b70BnpOYds2/view?usp=drive_link)

```
#include <ArduinoJson.h> #include <dht.h>
```

```
#define DHTPIN 2 // Pino do sensor DHT11
```

```
#define LIGHTPIN A0 // Pino analógico para o sensor de luminosidade #define LEDPIN 13 // Pino do LED
```

```
dht DHT;
```

```
void setup() {
```

```
Serial.begin(9600); pinMode(LEDPIN, OUTPUT);
```

```
}
```

```
void loop() { delay(2000);
```

```
float temperature = DHT.readTemperature(); float humidity = DHT.readHumidity();
```

```
int lightValue = analogRead(LIGHTPIN); // Lê o valor do sensor de luminosidade
```

```
if (isnan(temperature) || isnan(humidity)) {
```

```
Serial.println("Erro ao ler o sensor DHT!"); return;
```

```
}
```

```
StaticJsonDocument<200> jsonDoc; jsonDoc["temperatura"] = temperature; jsonDoc["umidade"] =
```

```
humidity; jsonDoc["luminosidade"] = lightValue;
```

```
String jsonString;
```

```
serializeJson(jsonDoc, jsonString);
```

```
Serial.println(jsonString);
```

```
if (temperature > 30 || humidity < 40 || lightValue < 500) { digitalWrite(LEDPIN, HIGH);
```

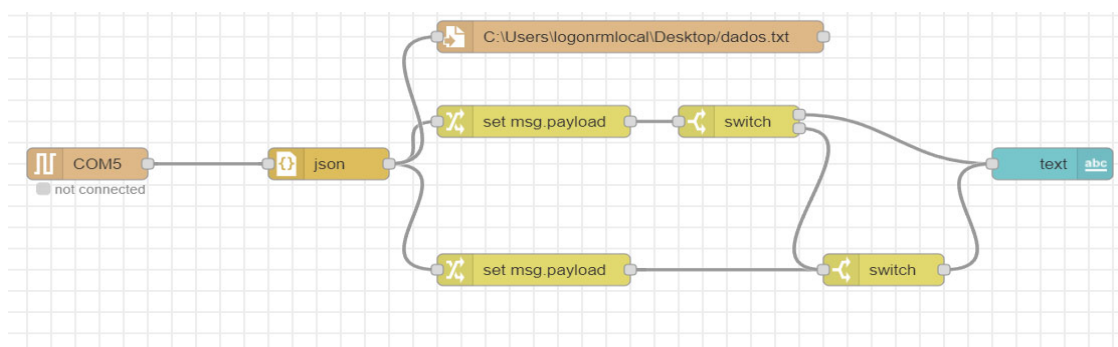
```
} else {
```

```
digitalWrite(LEDPIN, LOW);
```

```
}
```

```
}
```

**Desafios 1 e 2**



Desafio 3 adicional

