



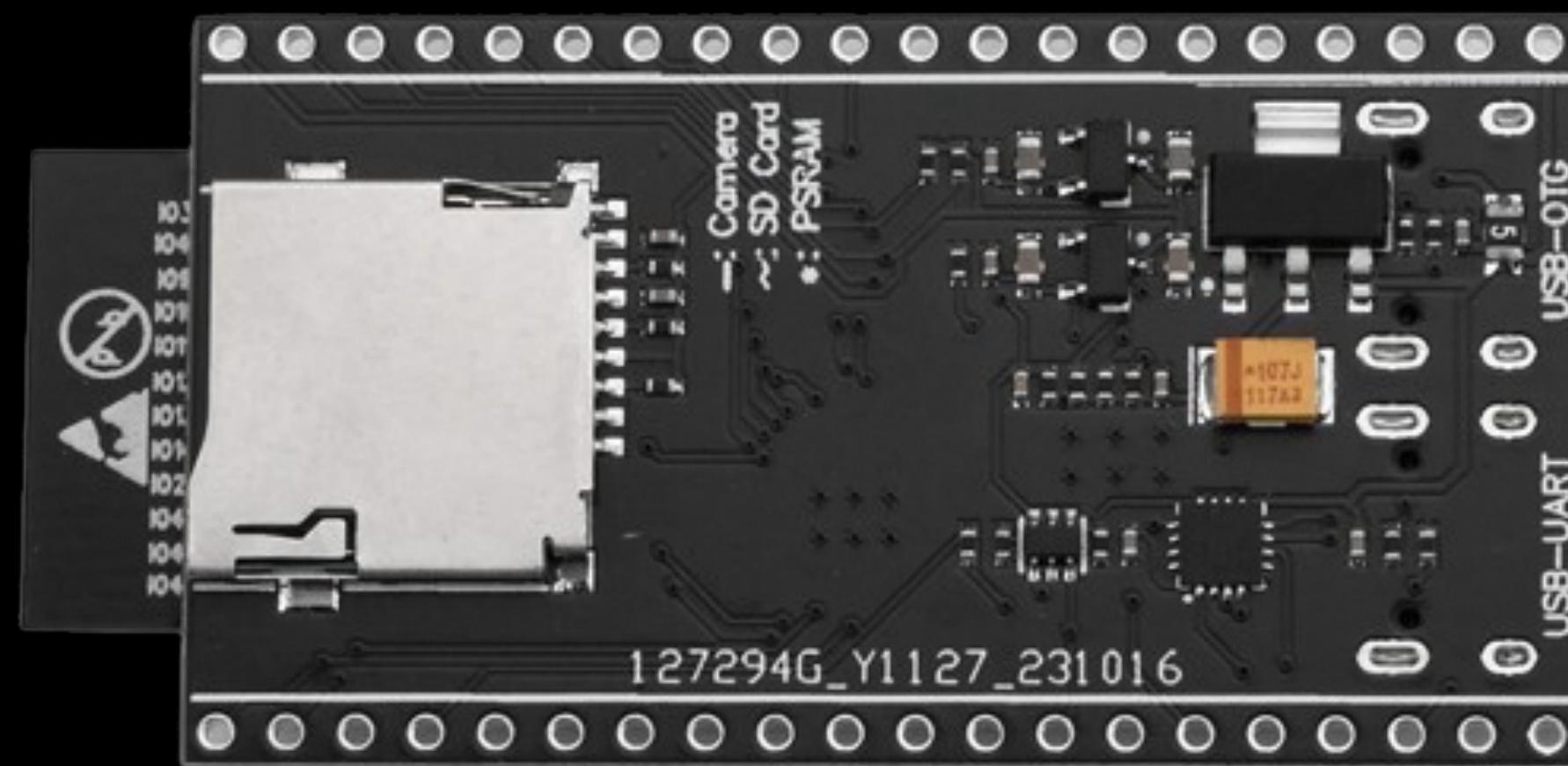
# Projeto 03

## Coisas Monitoradas – Teoria

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG4051 – Projeto Internet das Coisas

Coisas



```
#include <SD_MMC.h>

void setup () {
    SD_MMC.setPins(39, 38, 40); // ATENÇÃO À ORDEM EXATA!
    if (!SD_MMC.begin("/sdcard", true)) {
        Serial.println("Falha na inicialização do cartão SD");
        while (true) {} // trava programa aqui
    }
}

// em alguma função do código...
File arquivo = SD_MMC.open("/arquivo.txt", FILE_READ);
if (!arquivo) {
    Serial.println("Falha ao abrir o arquivo txt");
    while (true) {} // trava programa aqui
}

String textoDoArquivo = arquivo.readString();
arquivo.close();
```

```
#include <SD_MMC.h>

void setup () {
    SD_MMC.setPins(39, 38, 40); // ATENÇÃO À ORDEM EXATA!
    if (!SD_MMC.begin("/sdcard", true)) {
        Serial.println("Falha na inicialização do cartão SD");
        while (true) {} // trava programa aqui
    }
}

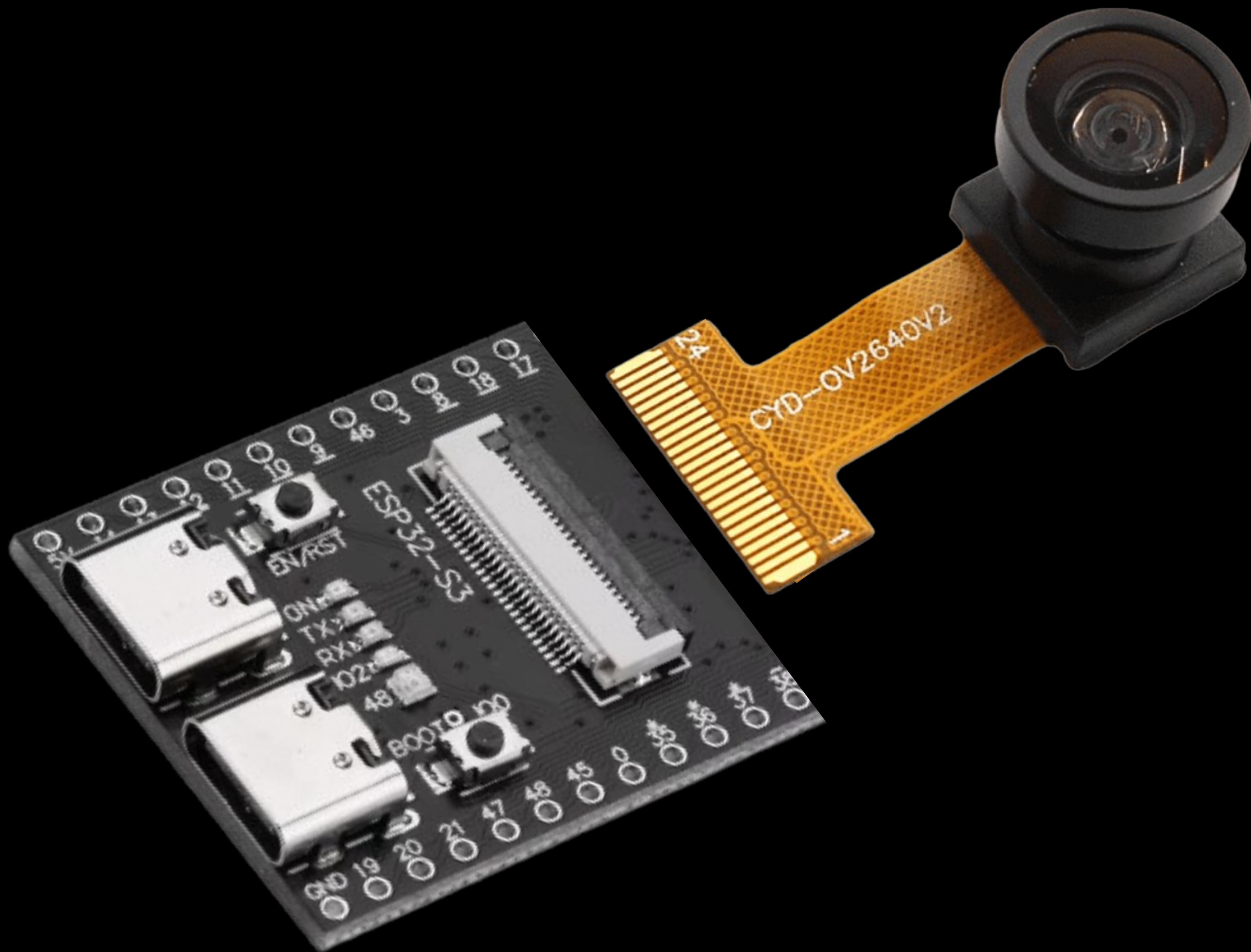
// em algum lugar do código...
File inputFile = SD_MMC.open("/arquivo.txt", FILE_WRITE);
inputFile.println("Olá!");
inputFile.close();
```

```
#include <WebServer.h>
WebServer servidor(80);

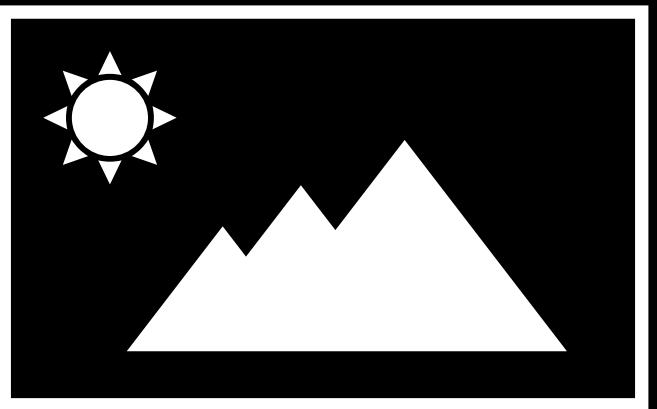
void mostrarFoto () {
    File arquivo = SD_MMC.open("/foto.jpeg", FILE_READ);
    if (!arquivo) {
        servidor.send(404, "text/plain", "Erro ao carregar foto!");
        return;
    }
    servidor.sendHeader("Content-Type", "image/jpeg");
    servidor.sendHeader("Content-Length", String(arquivo.size()));
    servidor.sendHeader("Connection", "close");
    servidor.streamFile(arquivo, "image/jpeg");
    arquivo.close();
}

void setup () {
    // inicia Serial, SD Card, WiFi, etc...

    servidor.on("/foto", HTTP_GET, mostrarFoto);
    servidor.begin();
}
// loop com reconectar WiFi, handleClient do servidor, etc..
```



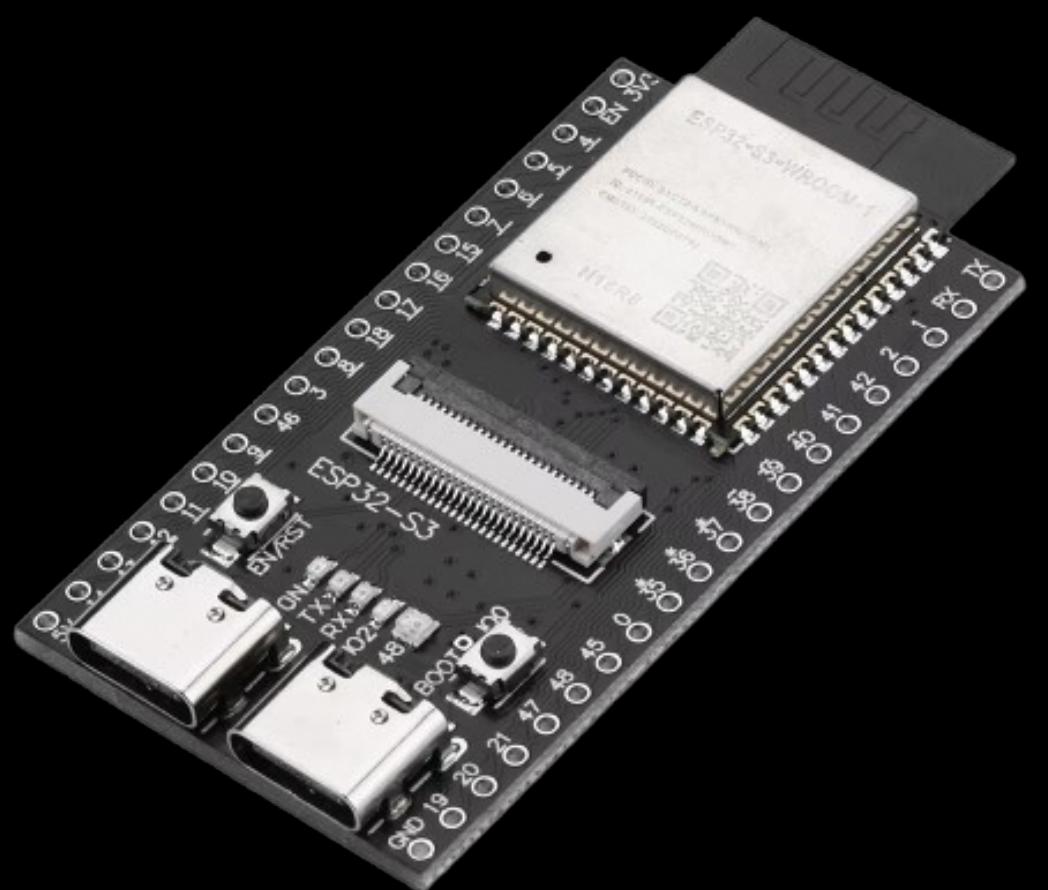
	STRB
	AGND
	SIO_D
	AVDD
	SIO_C
	RESETB
	VSYNC
	PWDN
	HREF
	DVDD
	DOVDD
	DATA9
	XCLK
	DATA8
	DGND
	DATA7
	PCLK
	DATA6
	DATA2
	DATA5
	DATA3
	DATA4
	DATA1
	DATA0



2 mega pixels → de ~300kB a ~800kB (JPEG)

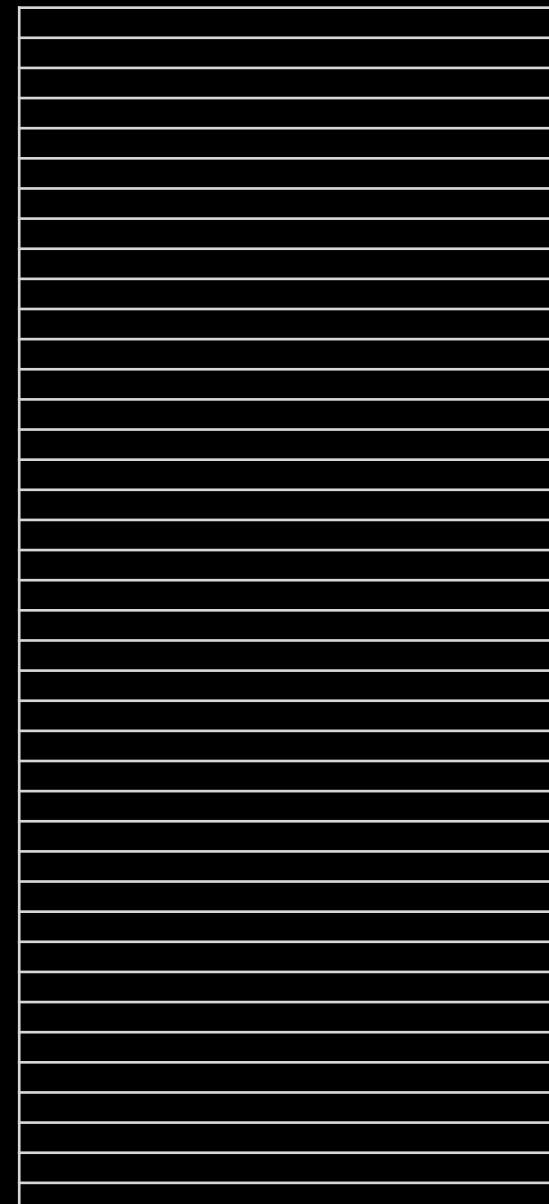
Memória SRAM (interna)

512kB

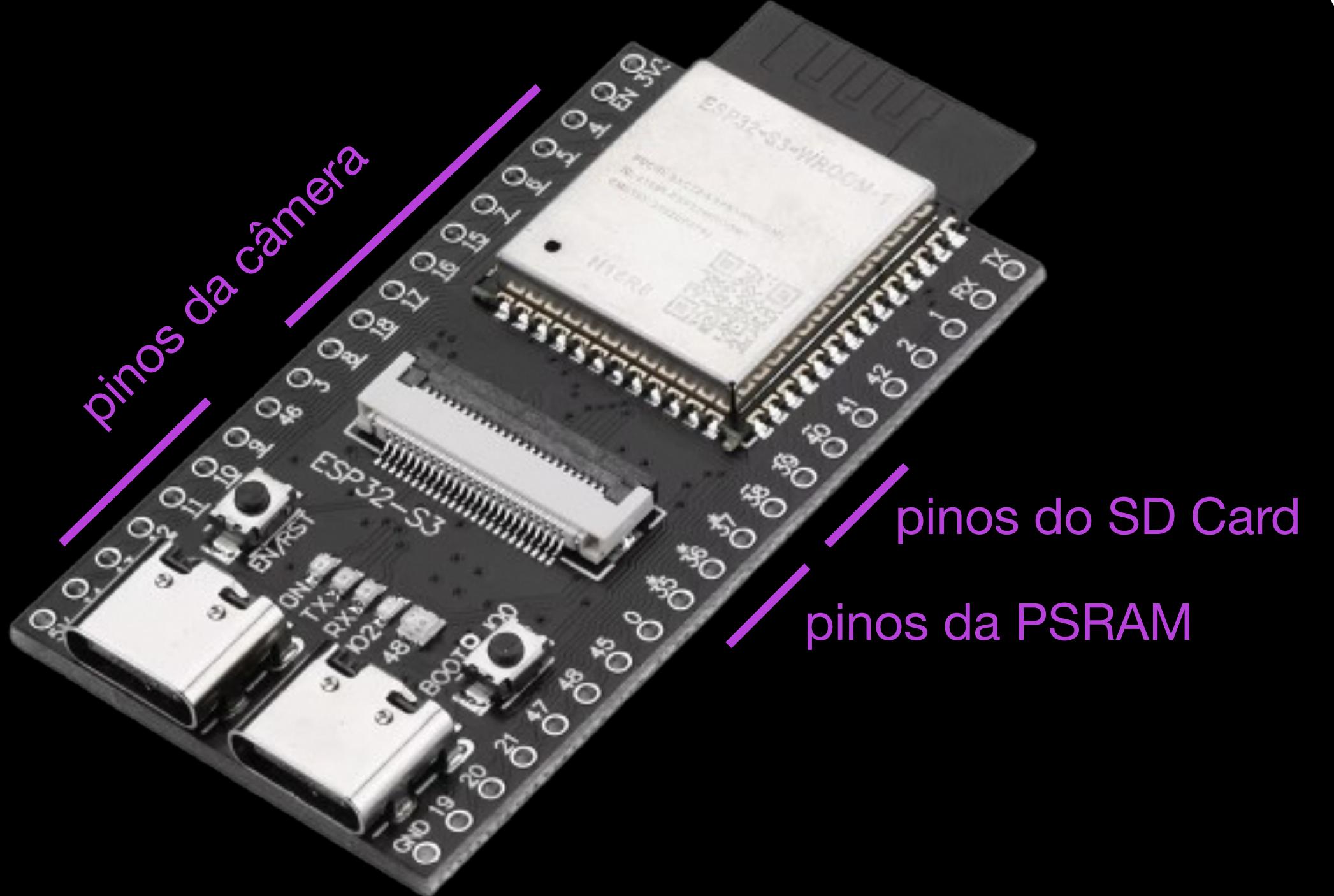


Memória PSRAM (Externa)

8MB



Não dá para conectar esses pinos em outros dispositivos quando usamos a câmera, a PSRAM e o SD Card



```
#include <esp_camera.h>

camera_config_t config = {
    .pin_pwdn = -1, .pin_reset = -1,
    .pin_xclk = 15, .pin_sscb_sda = 4, .pin_sscb_scl = 5,
    .pin_d7 = 16, .pin_d6 = 17, .pin_d5 = 18, .pin_d4 = 12,
    .pin_d3 = 10, .pin_d2 = 8, .pin_d1 = 9, .pin_d0 = 11,
    .pin_vsync = 6, .pin_href = 7, .pin_pclk = 13,
    .xclk_freq_hz = 20000000,
    .ledc_timer = LEDC_TIMER_0, .ledc_channel = LEDC_CHANNEL_0,
    .pixel_format = PIXFORMAT_JPEG, .frame_size = FRAMESIZE_SVGA,
    .jpeg_quality = 10, .fb_count = 2, .grab_mode = CAMERA_GRAB_LATEST
};

void setup () {
    // inicia Serial, WiFi, MQTT...

    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {
        Serial.printf("Falha ao inicializar a câmera: 0x%x", err);
        while (true) {} // trava programa aqui em caso de erro
    }
}

// código continua a seguir...
```

// continuação do código anterior...

```
void tirarFotoESalvarNoSDCard () {
    camera_fb_t* foto = esp_camera_fb_get();

    File arquivoFoto = SD_MMC.open("/foto.jpeg", FILE_WRITE);
    arquivoFoto.write(foto->buf, foto->len);
    arquivoFoto.close();

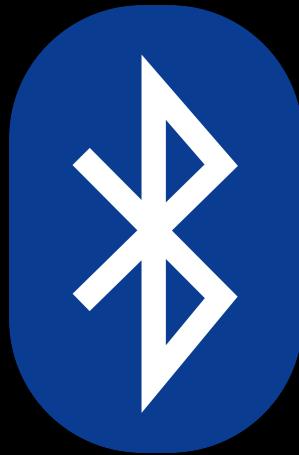
    esp_camera_fb_return(foto); // libera memória
}

void tirarFotoEEviarParaMQTT () {
    camera_fb_t* foto = esp_camera_fb_get();

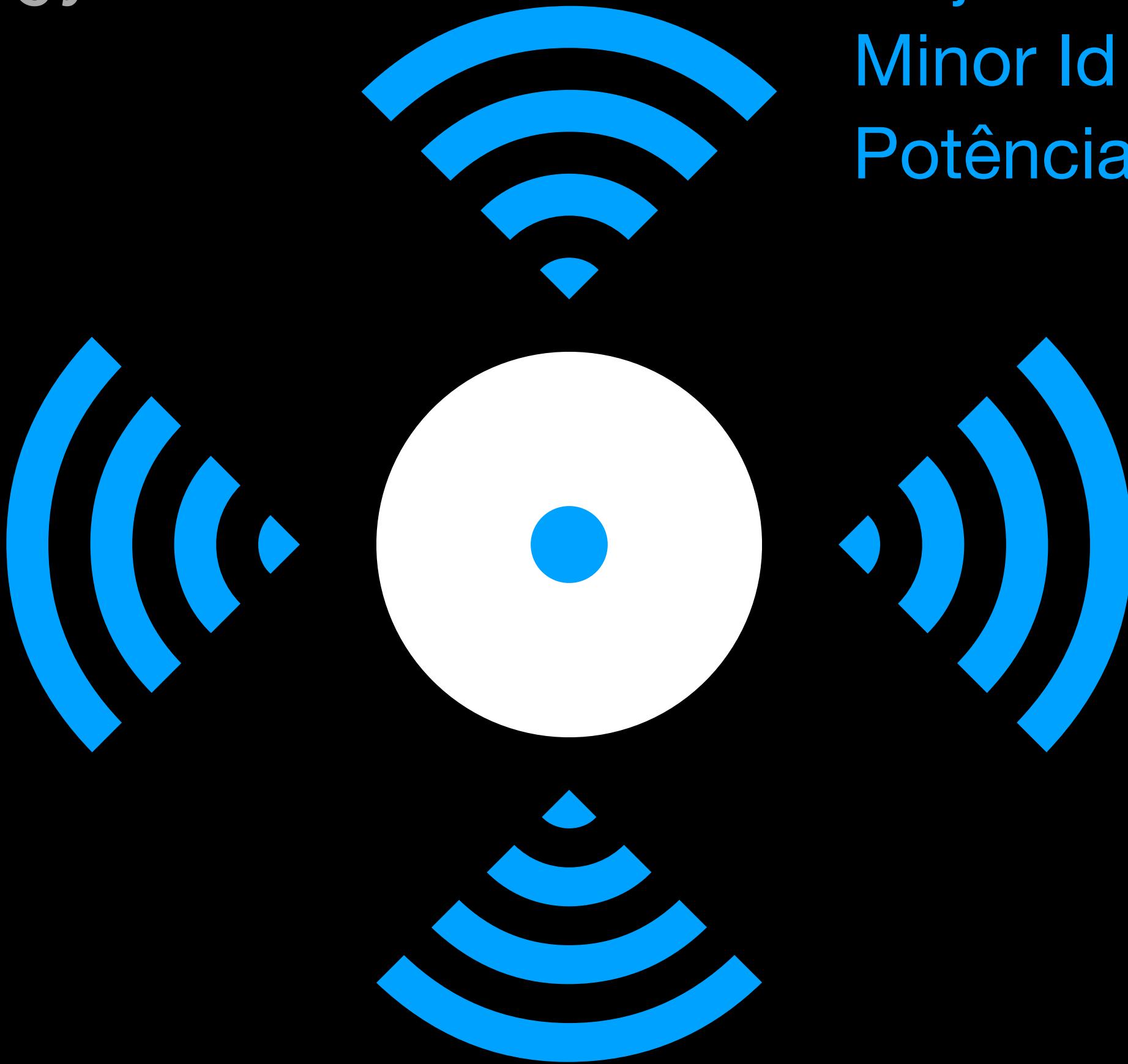
    if (mqtt.publish("topico", (const char*)foto->buf, foto->len)) {
        Serial.println("Foto enviada com sucesso");
    } else {
        Serial.println("Falha ao enviar foto");
    }

    esp_camera_fb_return(foto); // libera memória
}
```

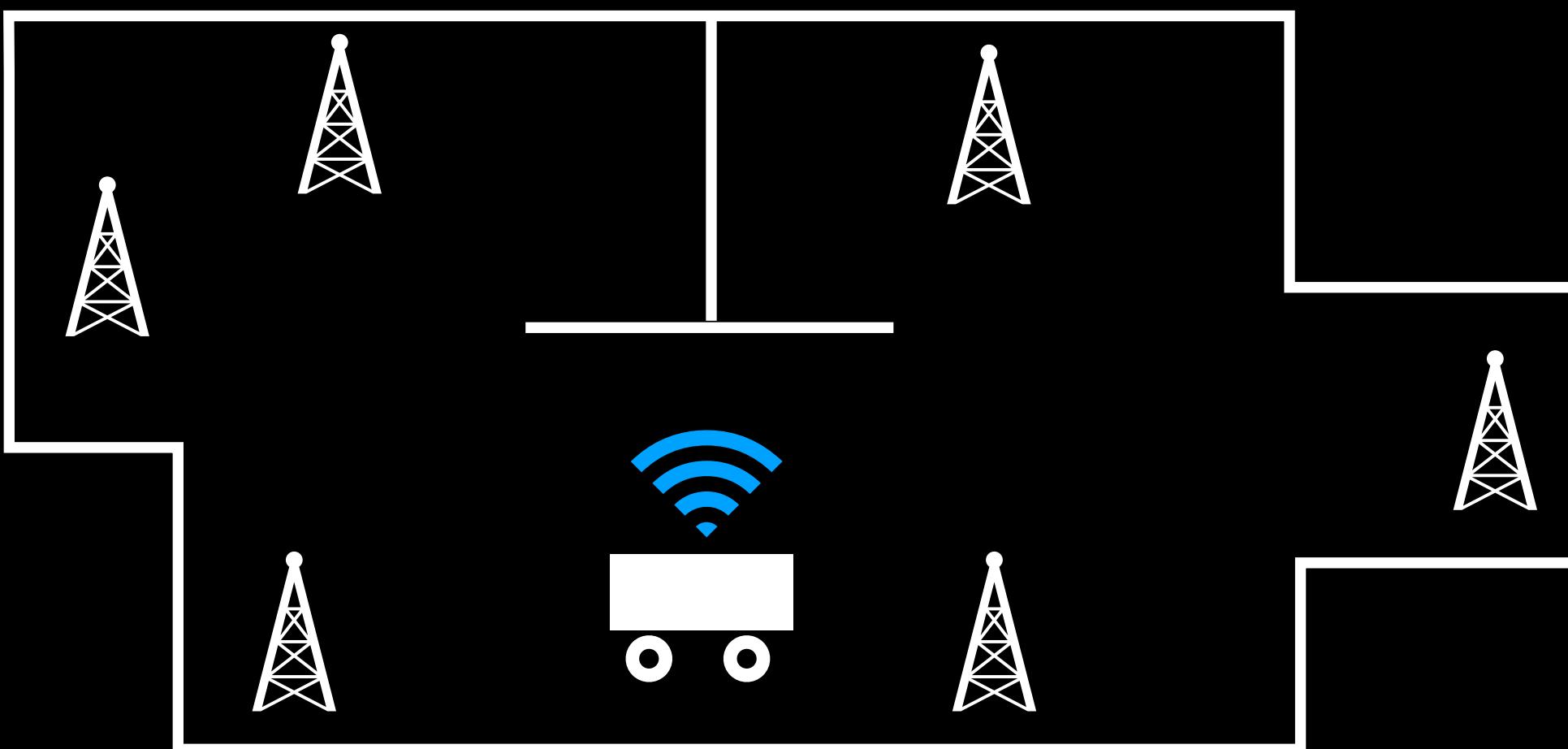
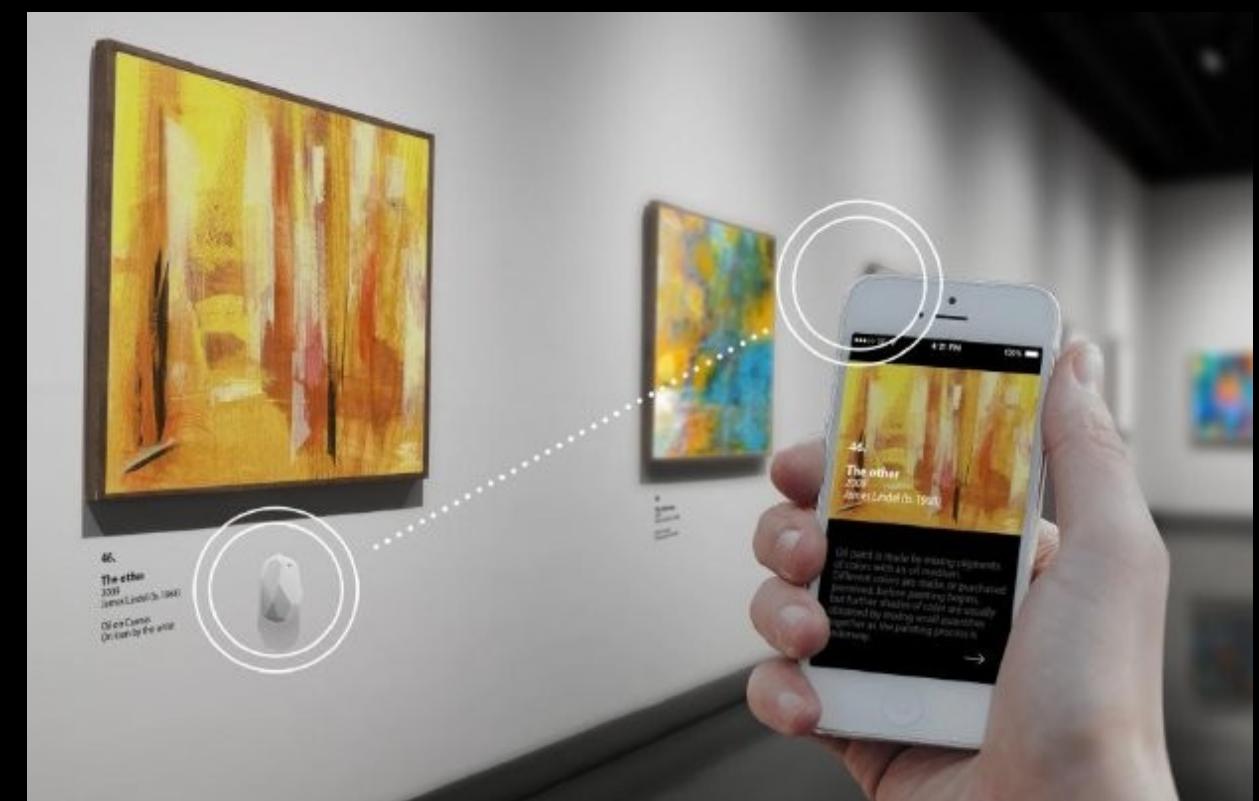
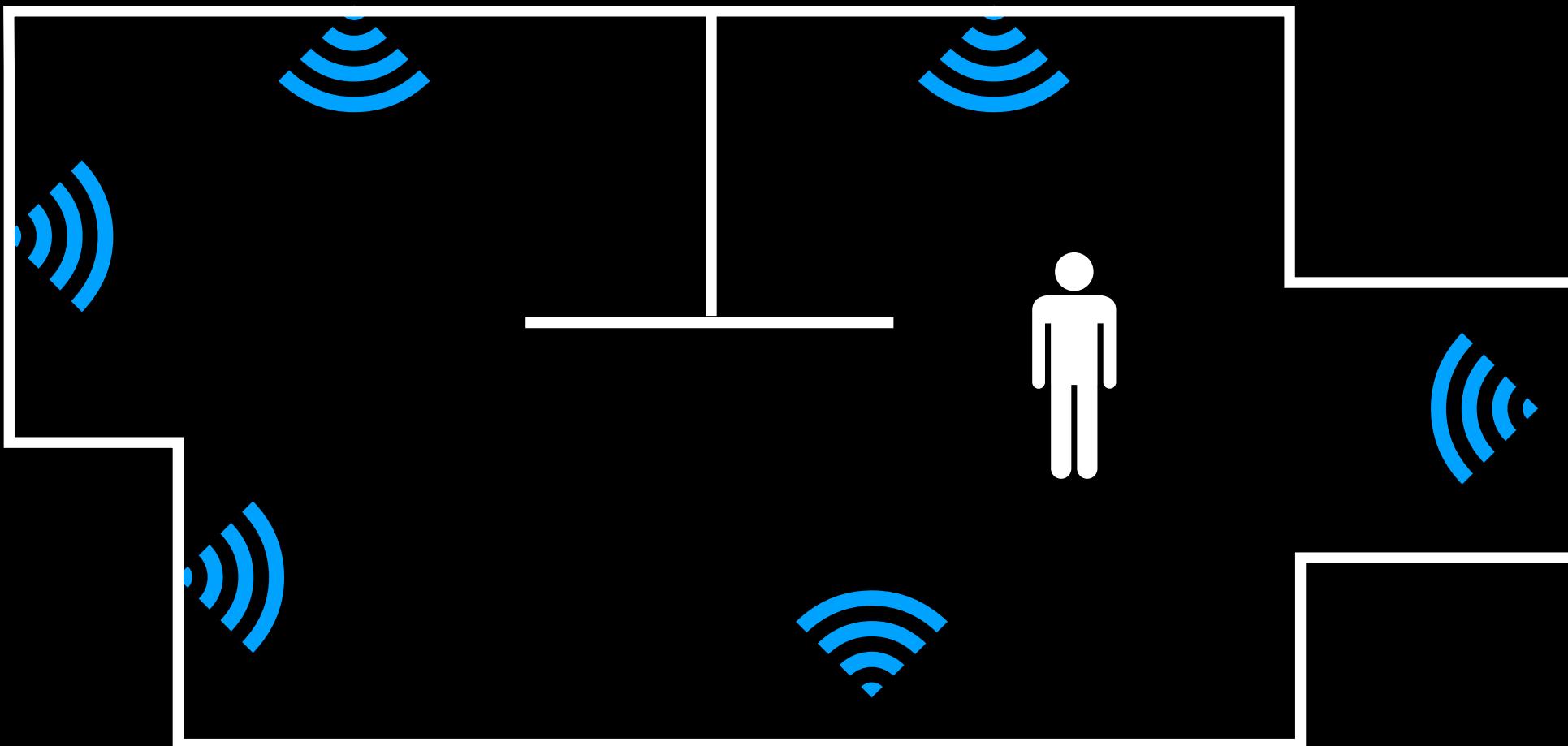


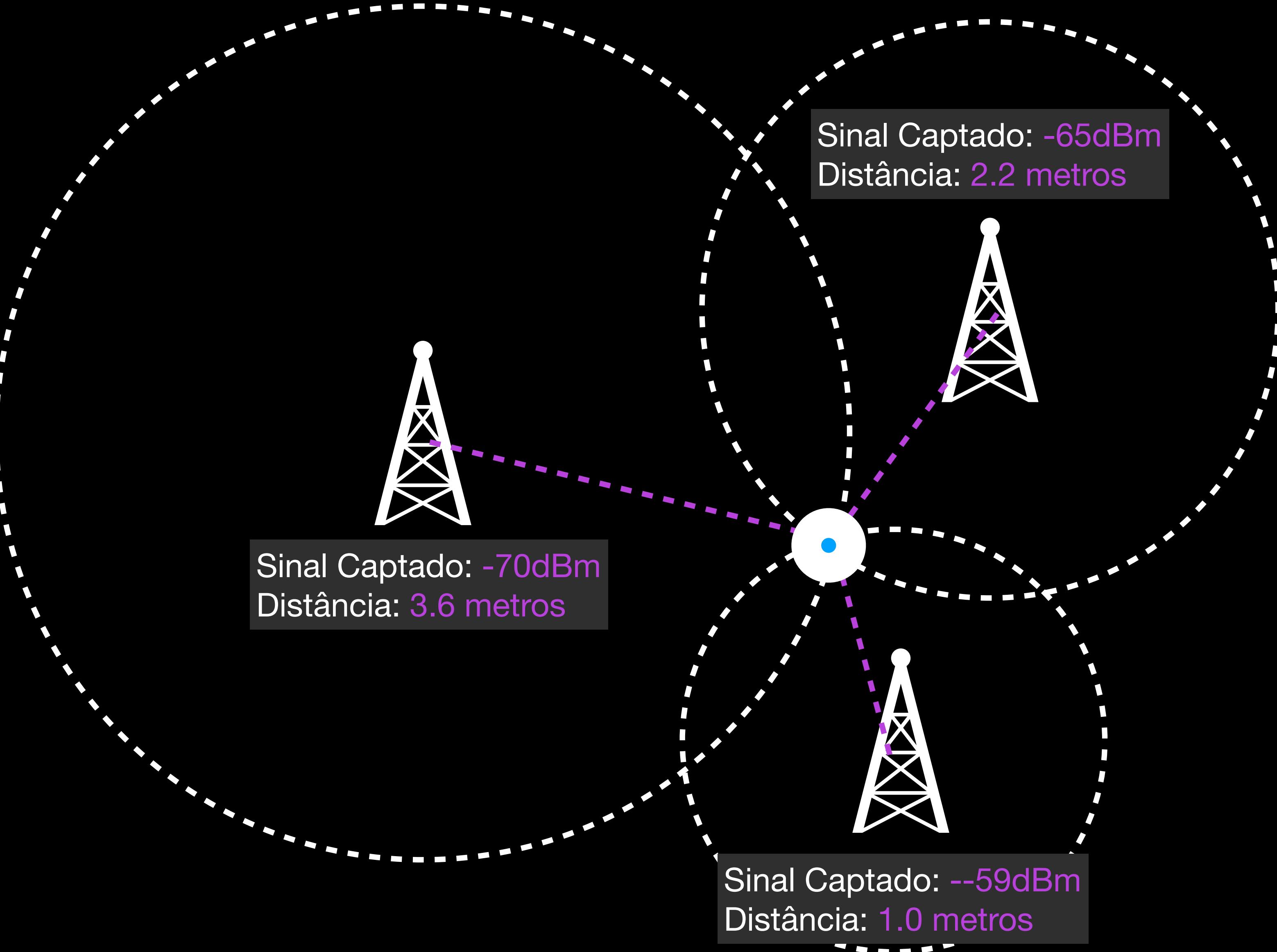


# Bluetooth Low Energy



Identificador Fabricante  
Major Id  
Minor Id  
Potência de Transmissão





```
#include <BLEDevice.h>
#include <BLEScan.h>
#include <BLEBeacon.h>

BLEScan* scannerBluetooth;
String idDoMeuBeacon = "ID DO SEU BEACON";

float calcularDistancia(int potenciaSinal) {
    if (potenciaSinal == 0) {
        return -1.0; // Não foi possível estimar a distância
    }

    int potenciaReferencia = -59; // dBm medido com distância de 1 metro
    float razao = potenciaSinal * 1.0 / potenciaReferencia;
    float distancia;
    if (razao < 1.0) {
        distancia = pow(razao, 10); // cálculo para distâncias < 1m
    } else {
        distancia = (0.89976) * pow(razao, 7.7095) + 0.111;
    }
    return (float)(int)(distancia * 10 + 0.5) / 10.0; // arredonda para 1 casa
}
// código continua a seguir...
```

```
// continuação do código anterior...

class MeuRastreador: public BLEAdvertisedDeviceCallbacks {
void onResult(BLEAdvertisedDevice dispositivoBluetooth) {
String dadosBeacon = dispositivoBluetooth.getManufacturerData();
if (dadosBeacon.length() != 25) {
    return; // não é Beacon
}

BLEBeacon oBeacon = BLEBeacon();
oBeacon.setData(dadosBeacon);
String idDispositivo = oBeacon.getProximityUUID().toString();

if (idDispositivo == idDoMeuBeacon) {
    scannerBluetooth->stop();

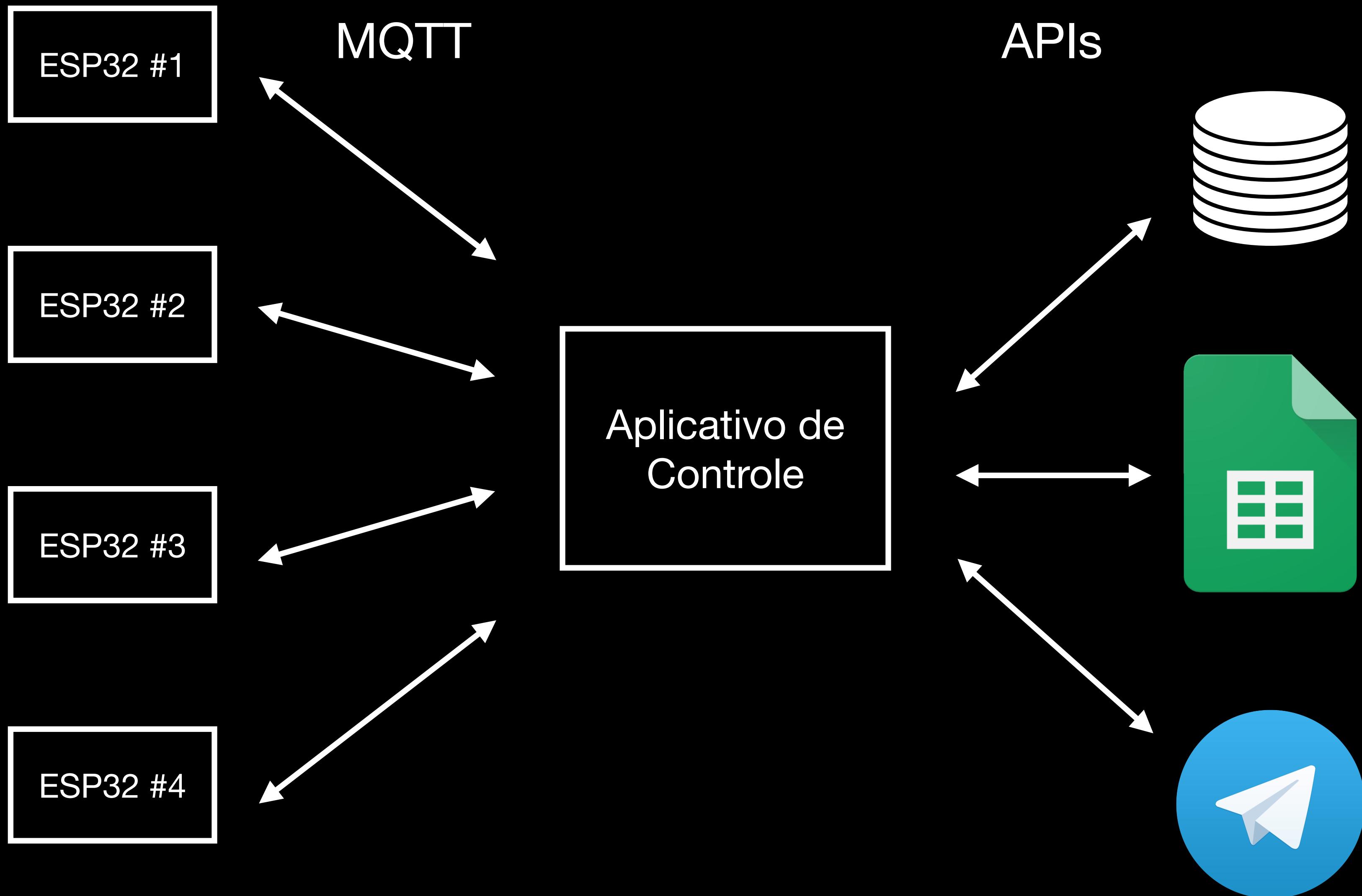
    int potenciaSinal = dispositivoBluetooth.getRSSI();
    float distancia = calcularDistancia(potenciaSinal);
    Serial.printf("Beacon encontrado a %.1f metros!\n", distancia);
}
}
};

// código continua a seguir...
```

```
// continuação do código anterior...
void setup() {
    BLEDevice::init("");
    scannerBluetooth = BLEDevice::getScan();
    scannerBluetooth->setAdvertisedDeviceCallbacks(new MeuRastreador());
    scannerBluetooth->setActiveScan(true); // captura mais informações
    scannerBluetooth->setInterval(100);
    scannerBluetooth->setWindow(99);
};

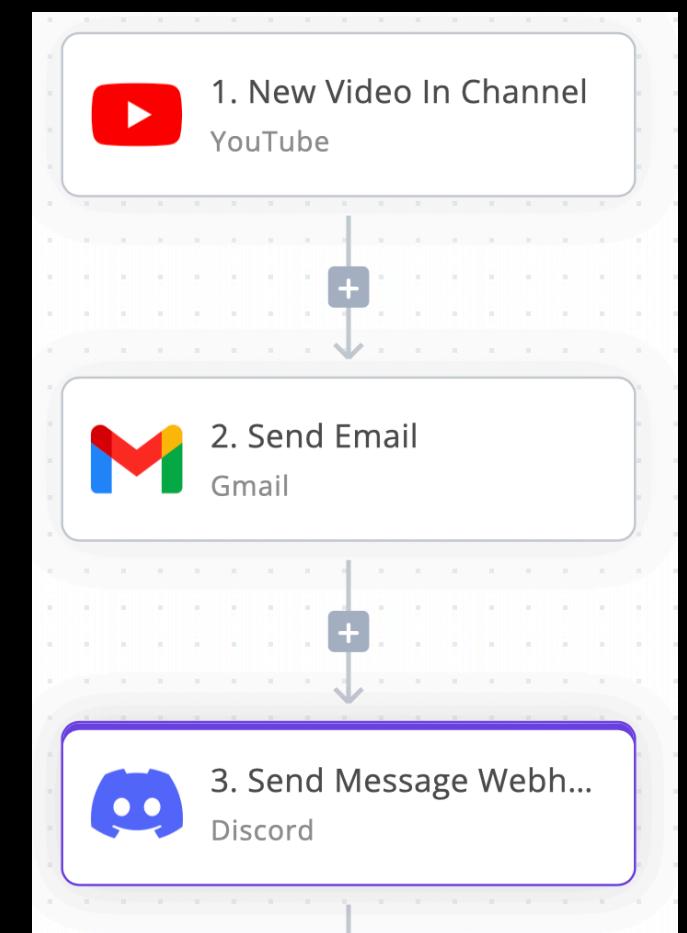
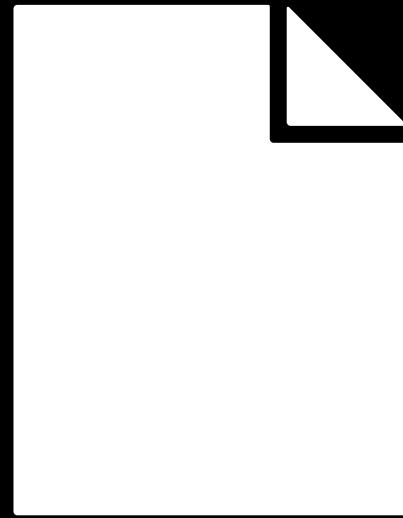
void loop() {
    scannerBluetooth->start(1, true); // escaneie durante 1s
    scannerBluetooth->clearResults();
}
```

# Internet

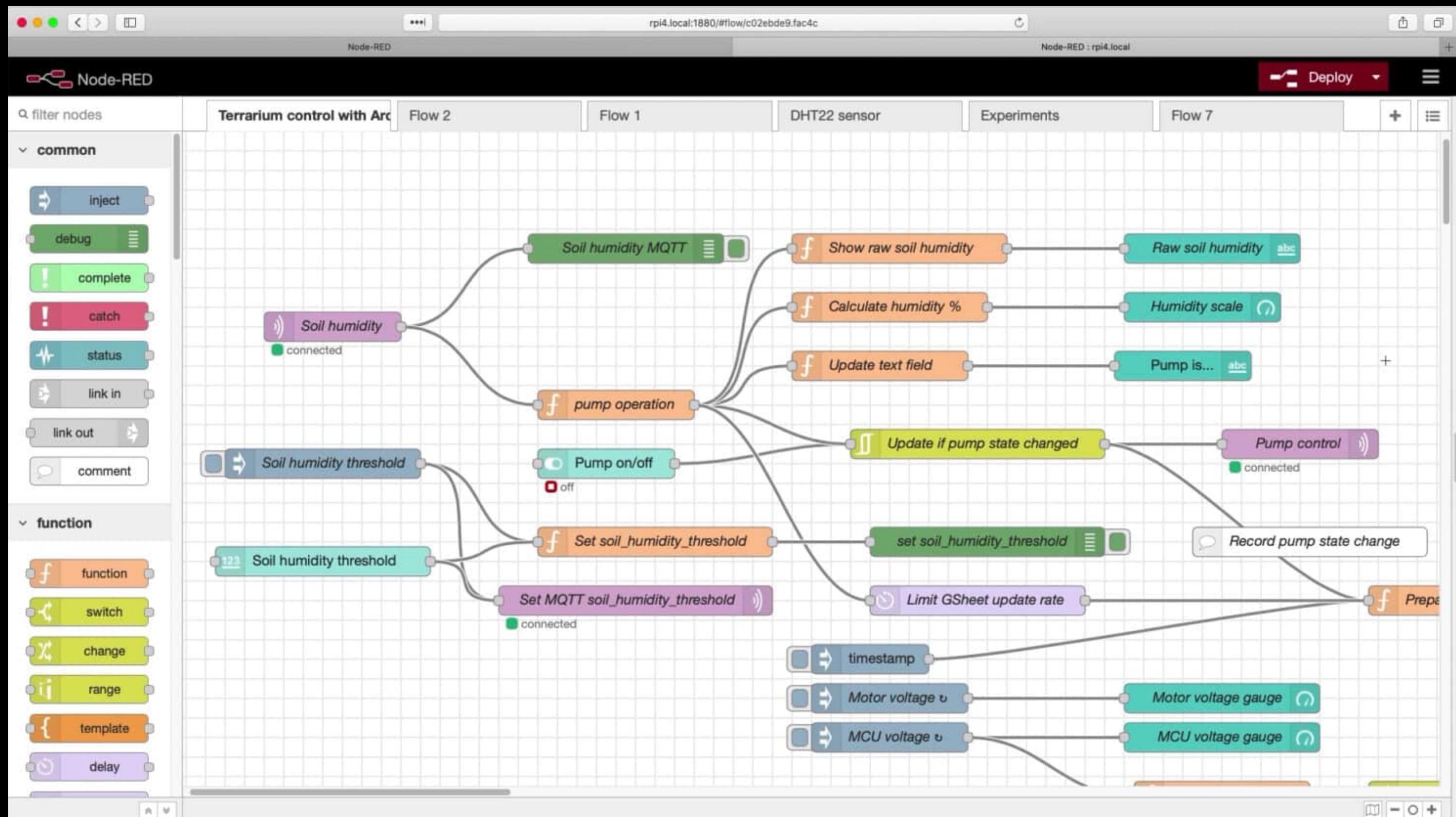
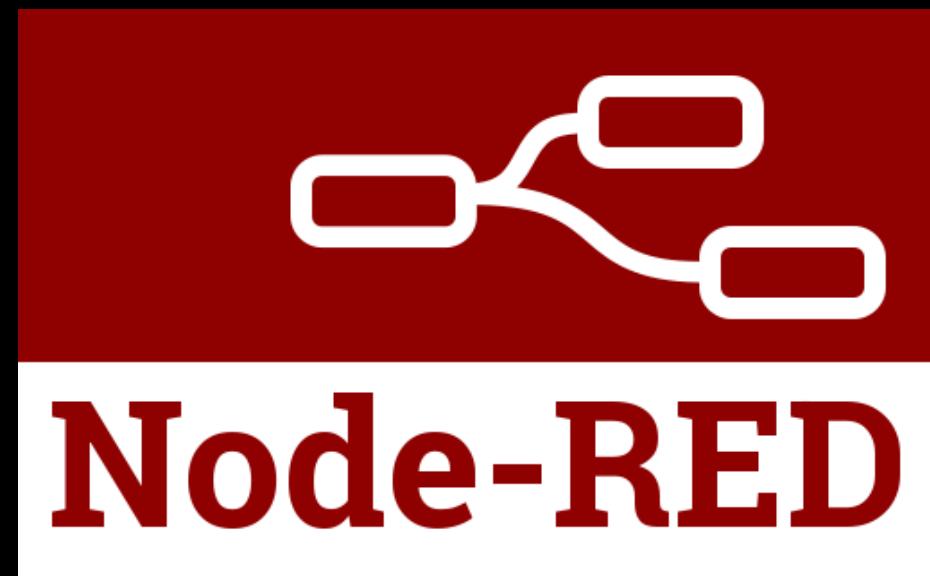


# Aplicativo de Controle

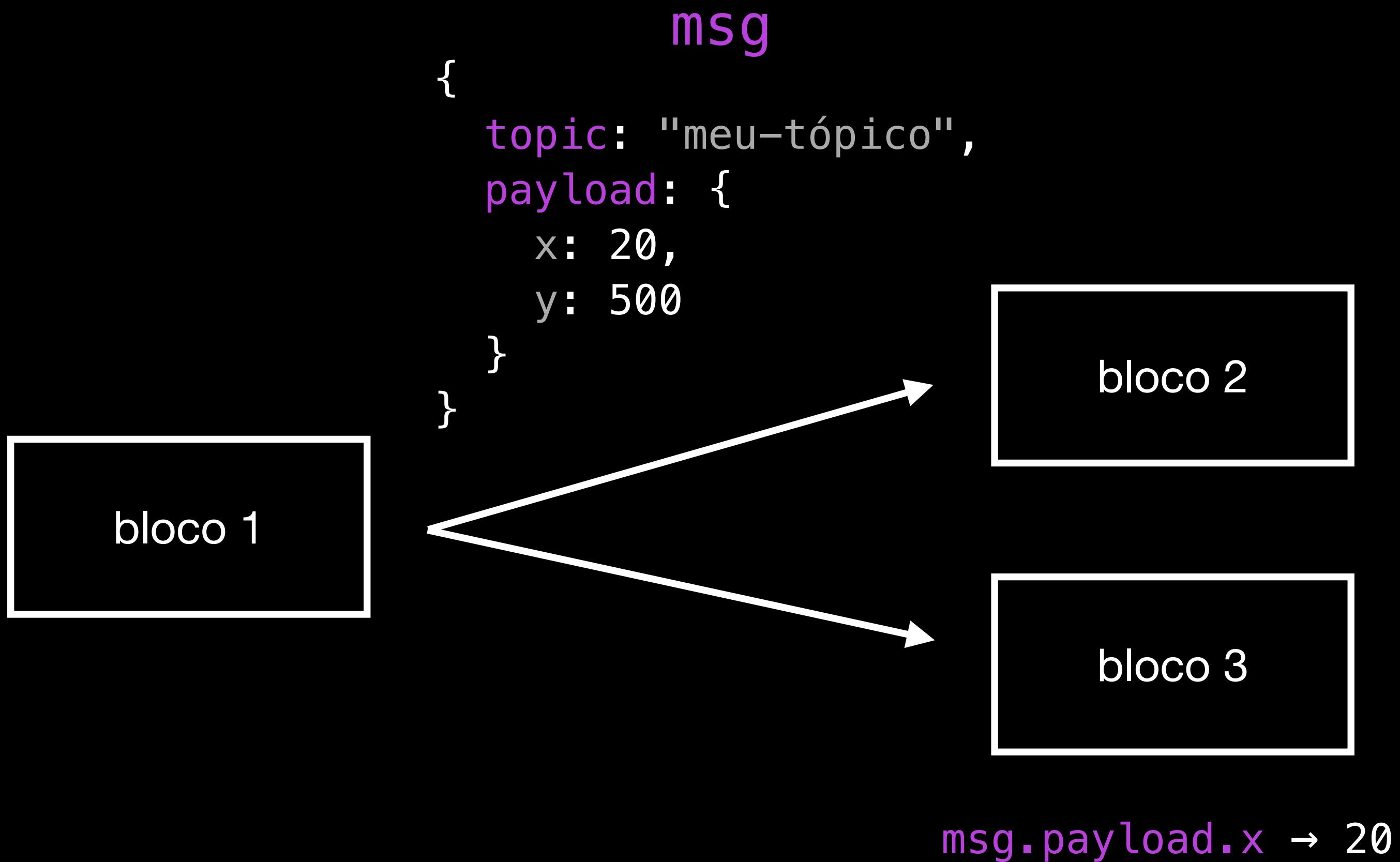
controle.py



- Bastante flexível e poderoso
- Configuração e manutenção mais complicadas
- Erros paralisam todo o aplicativo
- Menos flexível e menos poderoso
- Configuração e manutenção mais simples
- Erro numa sequência não paralisa outra



# Node-RED usa Javascript!



comum

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

função

- function
- switch
- change
- range
- template
- delay
- trigger
- exec
- filter

rede

- mqtt in
- mqtt out
- http in
- http response
- http request
- websocket in
- websocket out
- tcp in
- tcp out
- tcp request
- udp in
- udp out

sequencia

- split
- join
- sort
- batch

tópico1/10



Olá!



{

```
topic: "tópico1/10",  
payload: "Olá!"
```

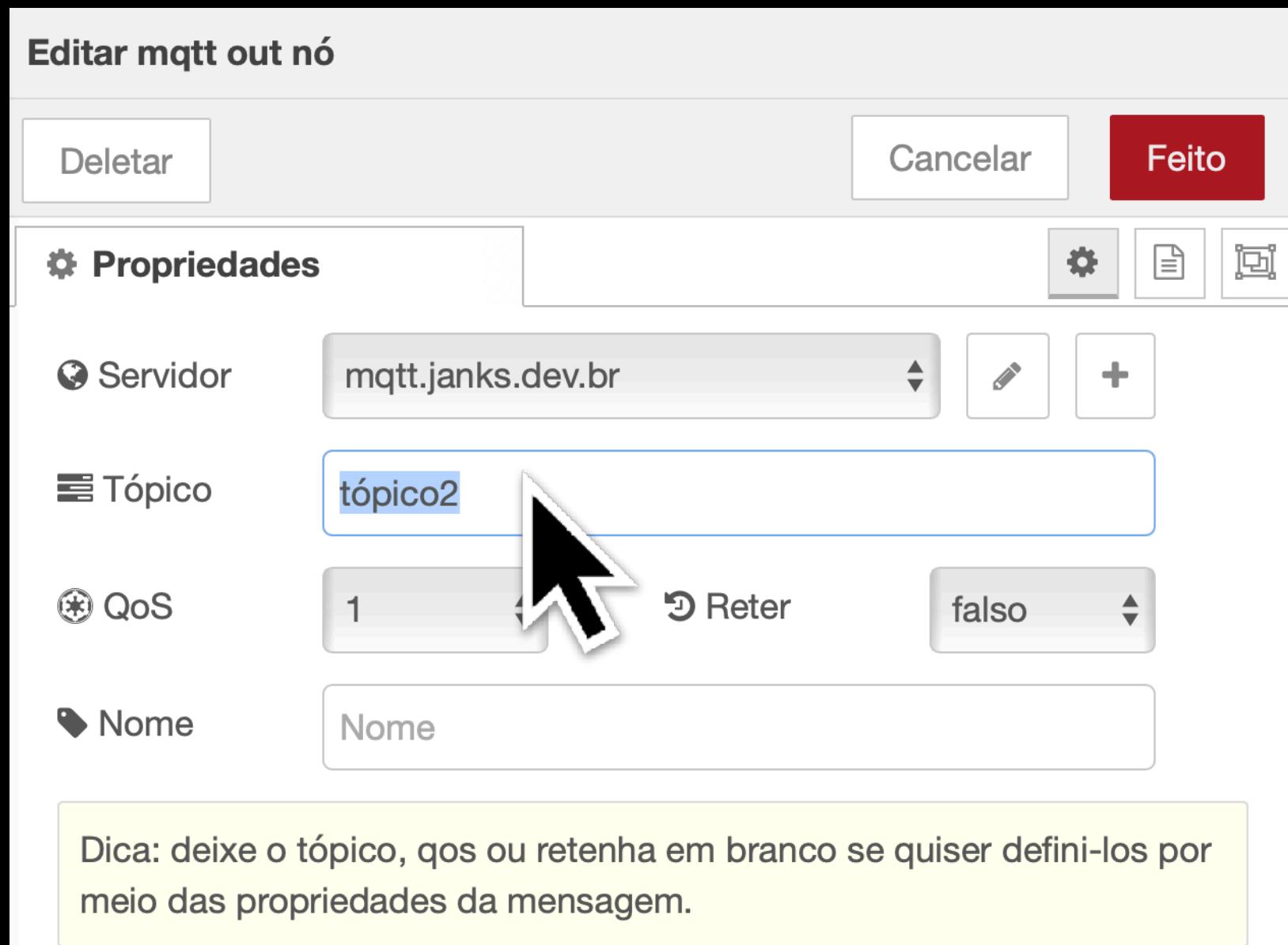
}

(clique duplo para configurar)

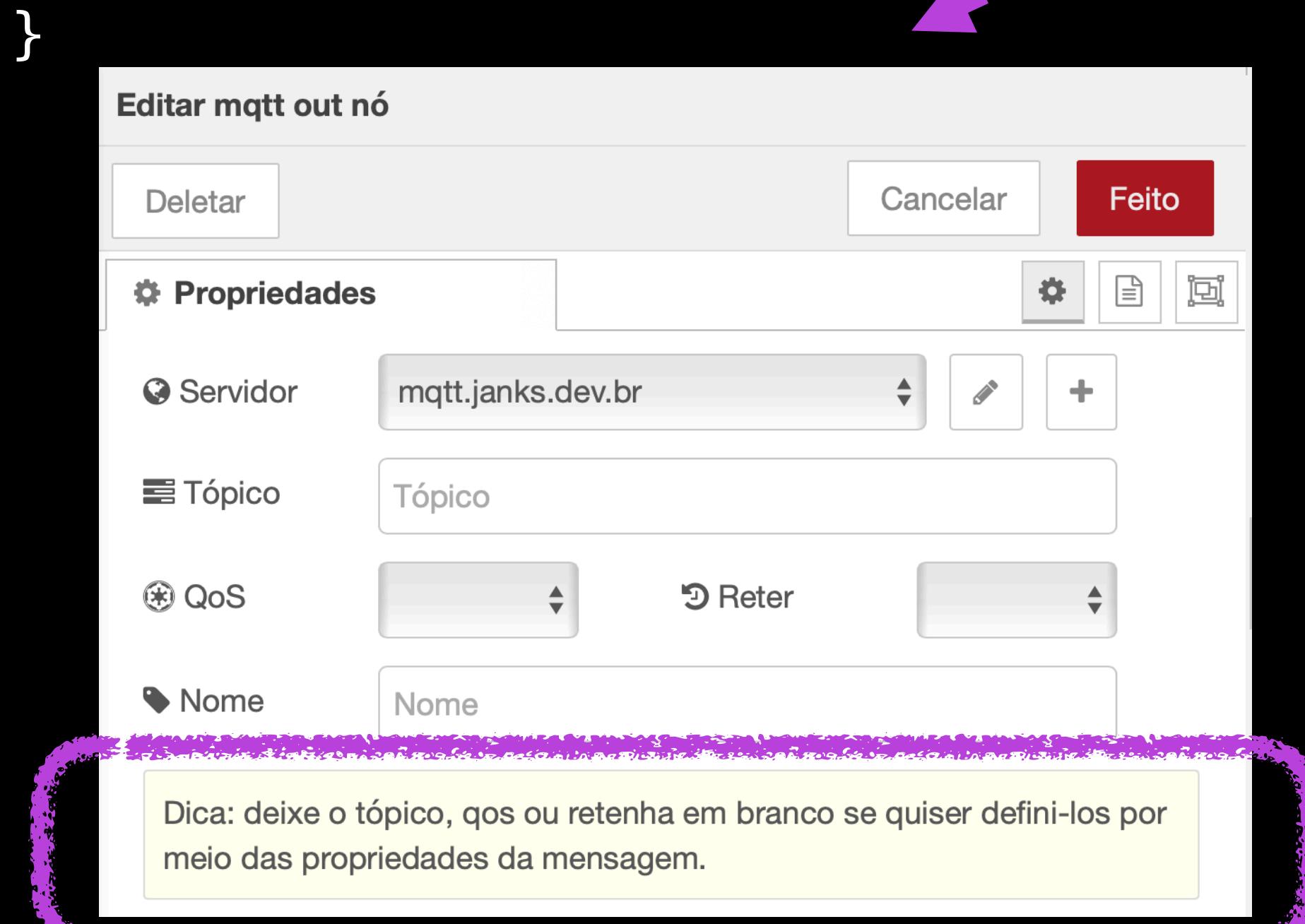
Editar mqtt in nó

Deletar		Cancelar	Feito
<b>Propriedades</b>			
Servidor	mqtt.janks.dev.br		
Ação	Assinar um tópico único		
Tópico	tópico1/+		
QoS	1		
Saída	auto-detecção(objeto JSON, cadeia de caracteres ou armazenamento)		
Nome	Nome		

```
{  
    payload: "OK!"  
}
```



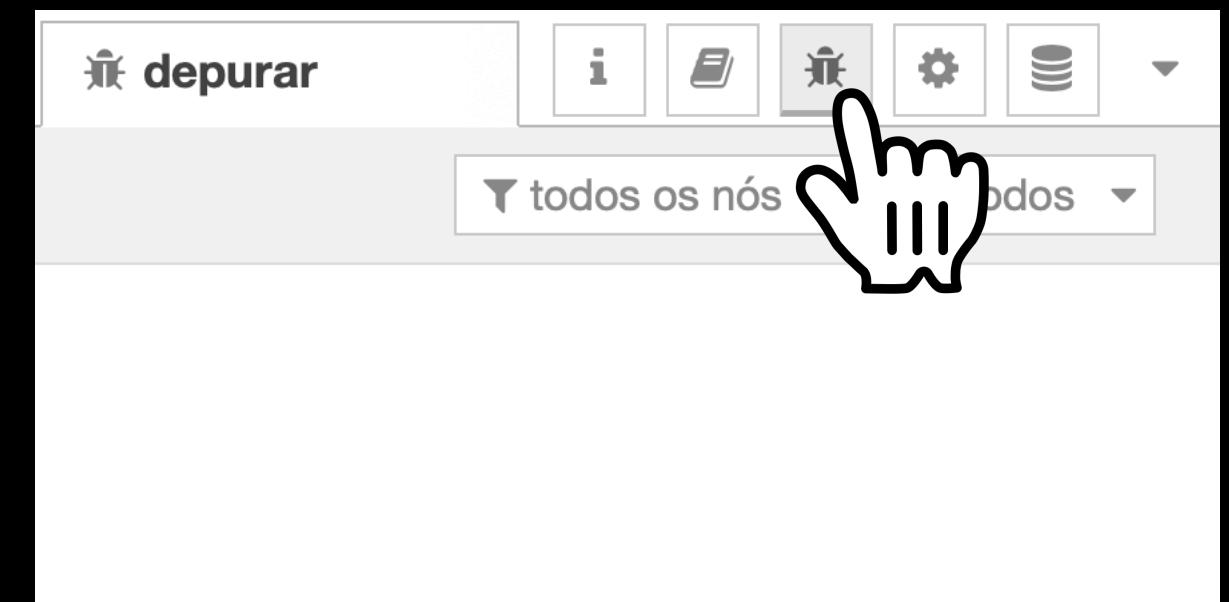
```
{  
    topic: "tópico2",  
    qos: 1,  
    retain: false,  
    payload: "conteúdo",  
}
```



```
{
  topic: "tópico4",
  qos: 0,
  retain: false
  payload: "Jan K. S.",
}
```



debug



**Propriedades**

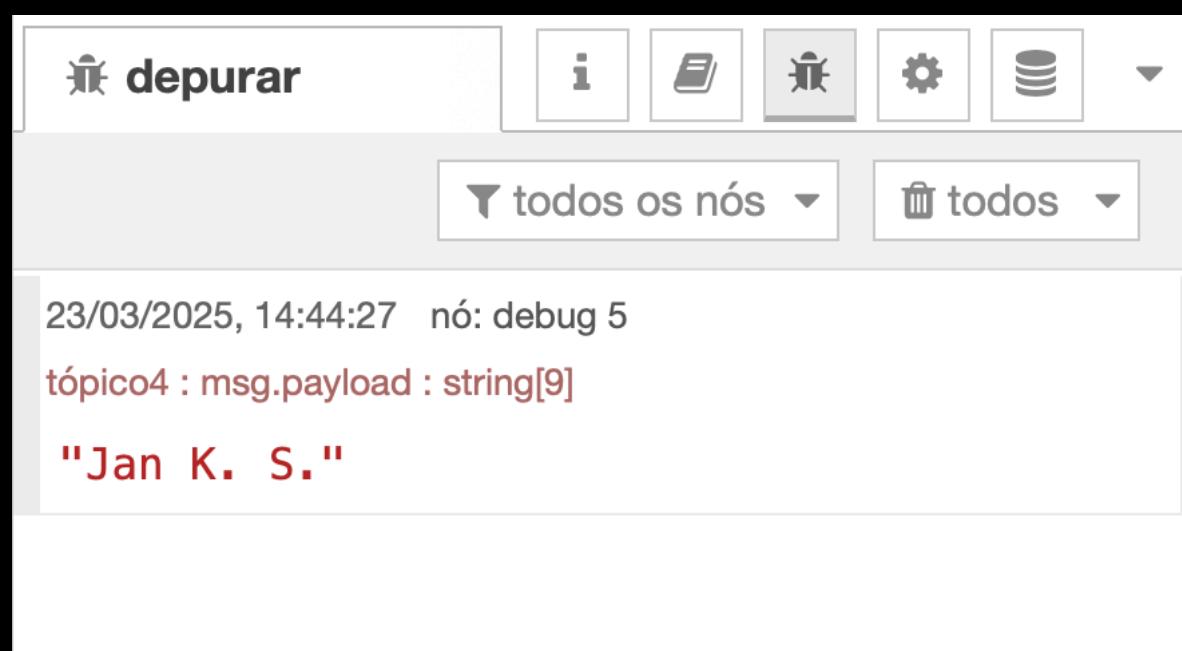
**Saída**  (highlighted with a purple box)

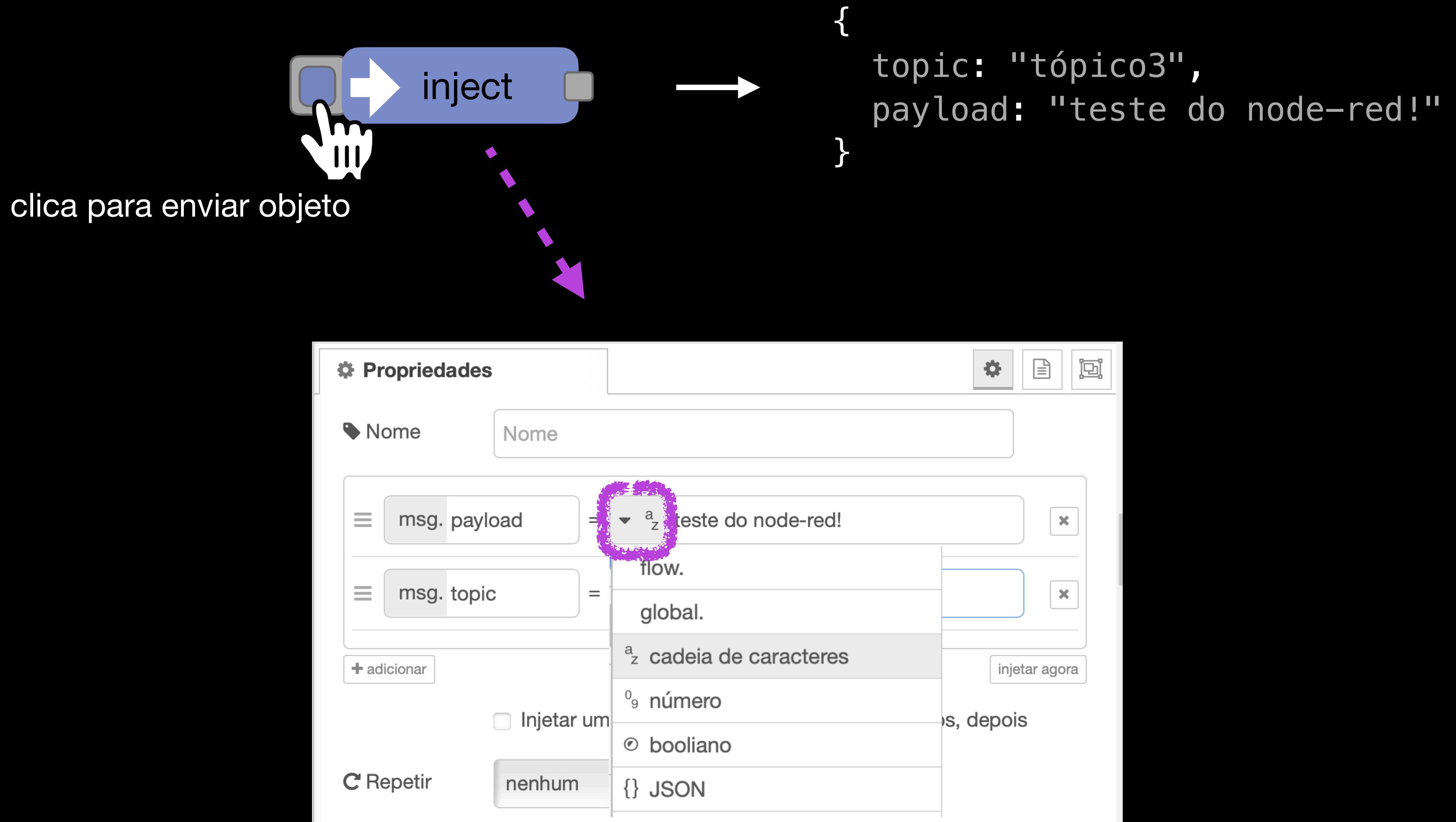
**Para**  janela de depuração  
 console do sistema  
 estado do nó (32 caracteres)

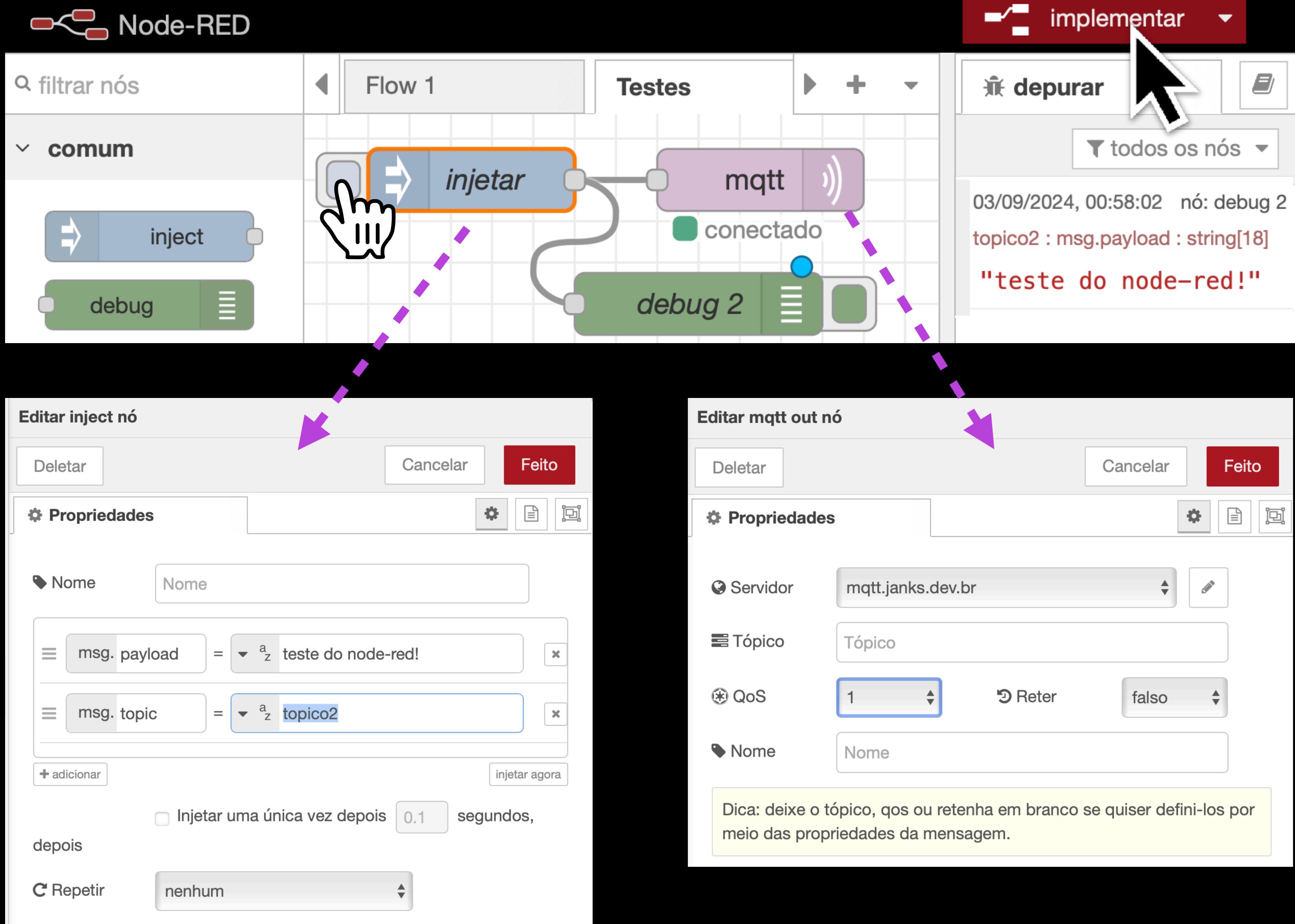
**Propriedades**

**Saída**  (highlighted with a purple box)

**Para**  janela de depuração  
 console do sistema  
 estado do nó (32 caracteres)







```
{  
  topic: "projeto",  
  payload: "micro"  
} → X change → {  
  topic: "projeto",  
  payload: "IoT"  
}
```

Regras

Definir msg. payload para o valor a\_z IoT

Regras

Definir msg. payload.x para o valor 42

Definir msg. payload.y para o valor 3.1415

Definir msg. payload.nome para o valor msg. topic

Valor de cópia profunda

```
{  
  topic: "dados",  
  payload: {  
    x: 13  
  }  
} → X change → {  
  topic: "dados",  
  payload: {  
    x: 42,  
    y: 3.1415,  
    nome: "dados"  
  }  
}
```

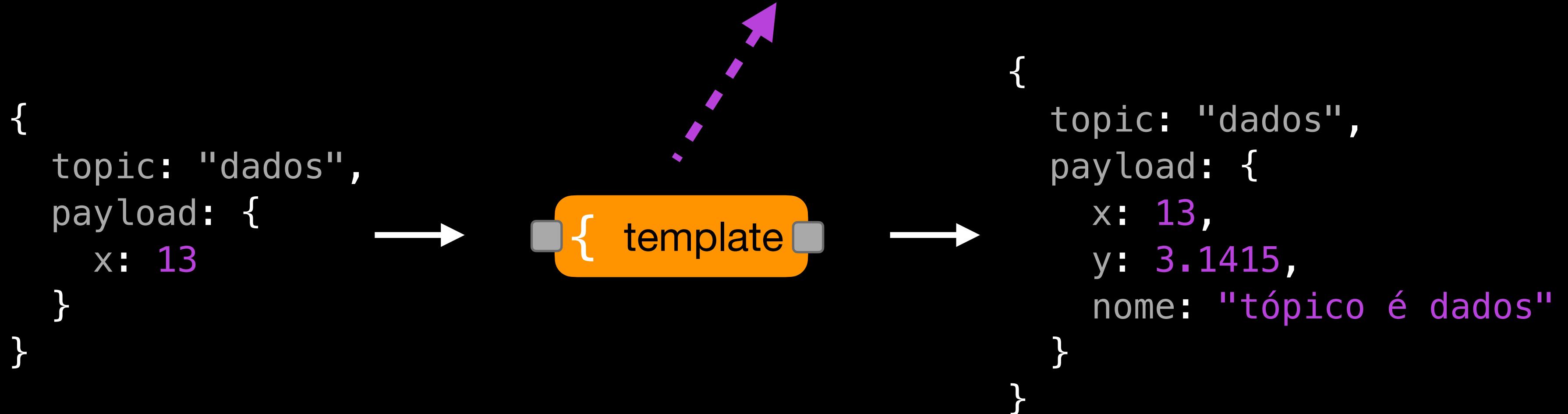
... Propriedade ▾ msg. payload

Modelo Destaque de sintaxe: mustache

```
1 {  
2   "x": {{payload.x}},  
3   "y": 3.1415,  
4   "nome": "tópico é {{topic}}"  
5 }
```

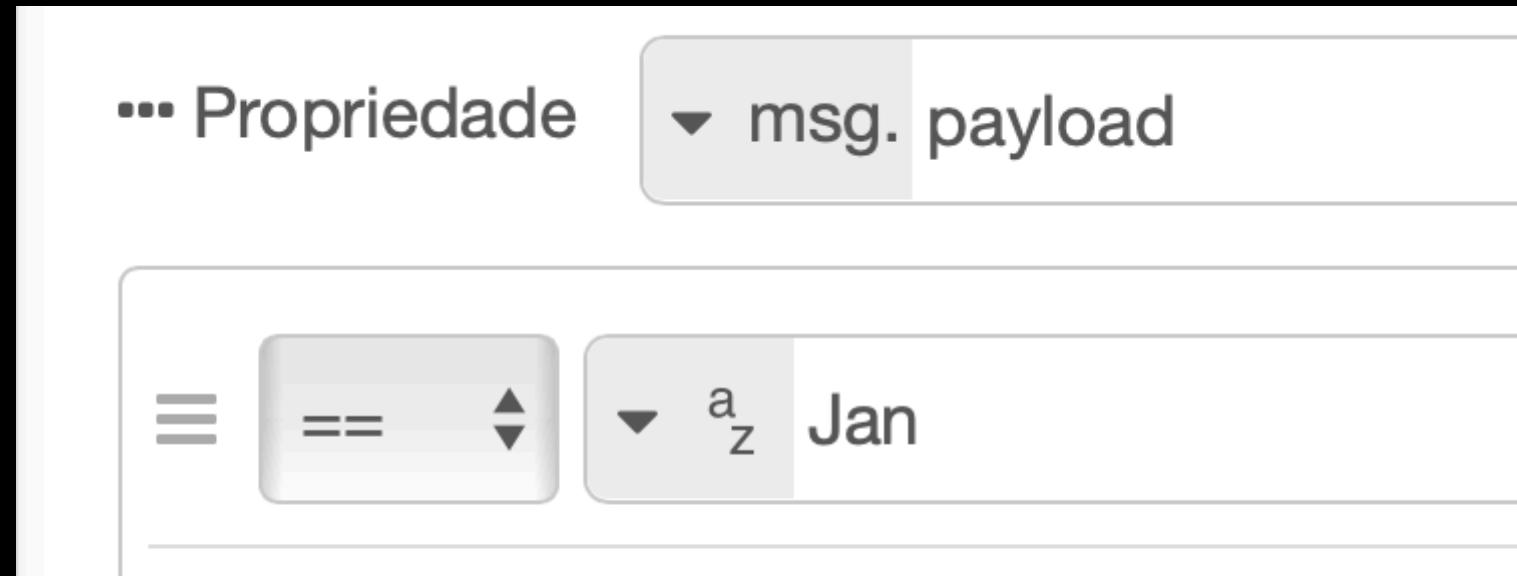
</> Formato Modelo de bigode

→ Saída como JSON analisado



{ payload: "Jan" } → switch → { payload: "Jan" }

{ payload: "Zoë" } → switch (não envia nada)



... Propriedade ▾ msg. payload

<  $0_9$  5

==  $0_9$  5

>  $0_9$  5

+ adicionar

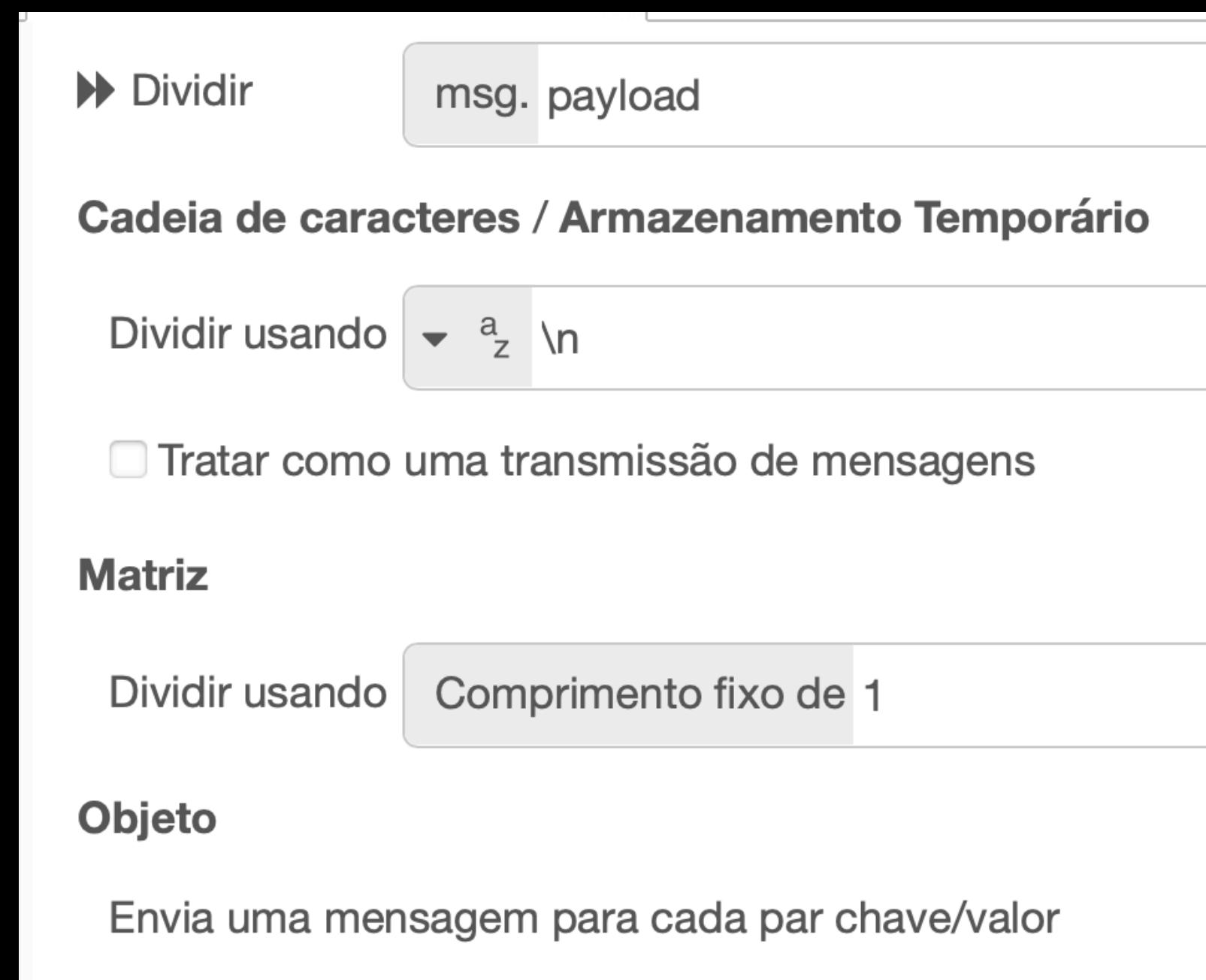
✓ checando todas as regras  
parando após a primeira correspondência

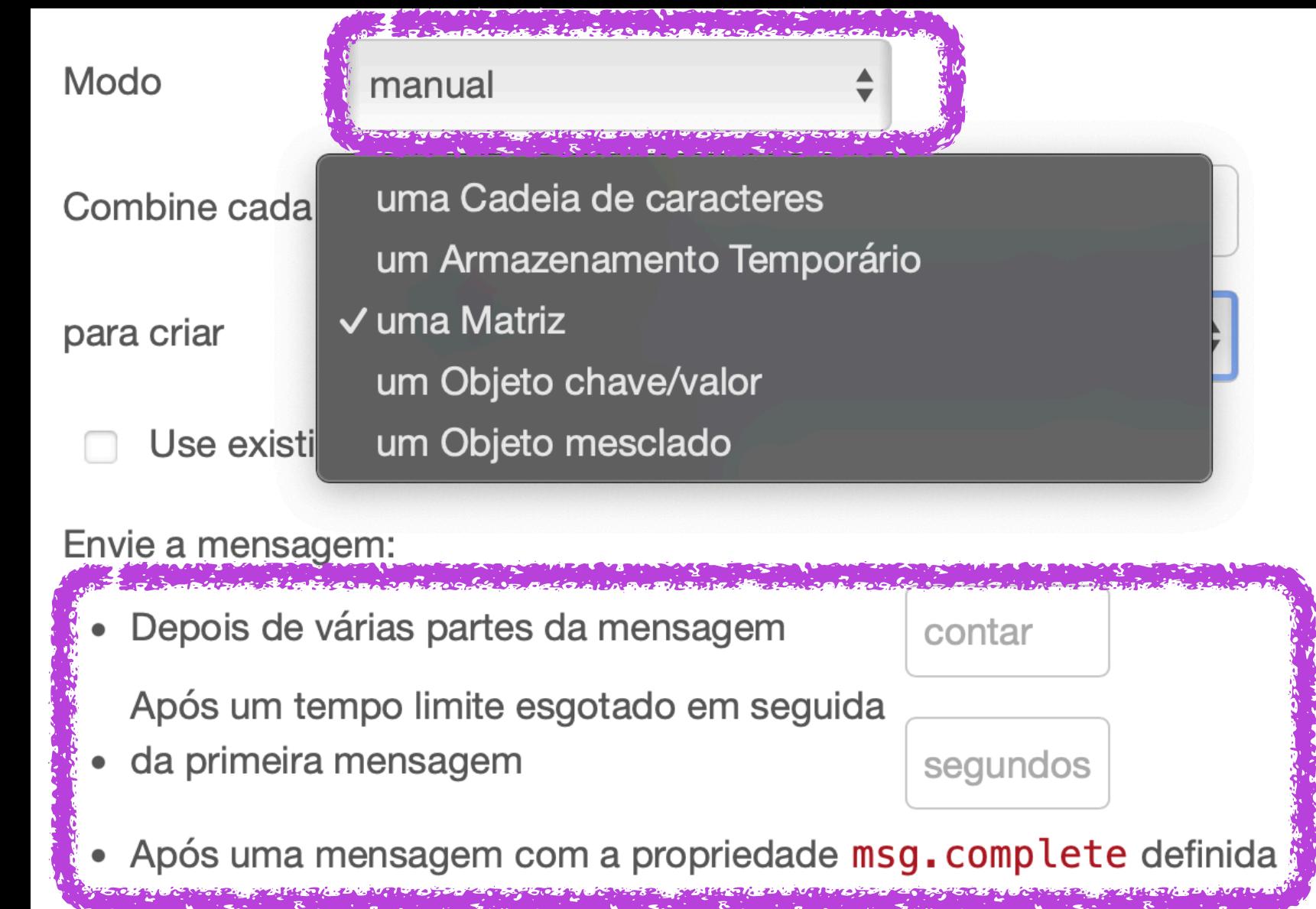
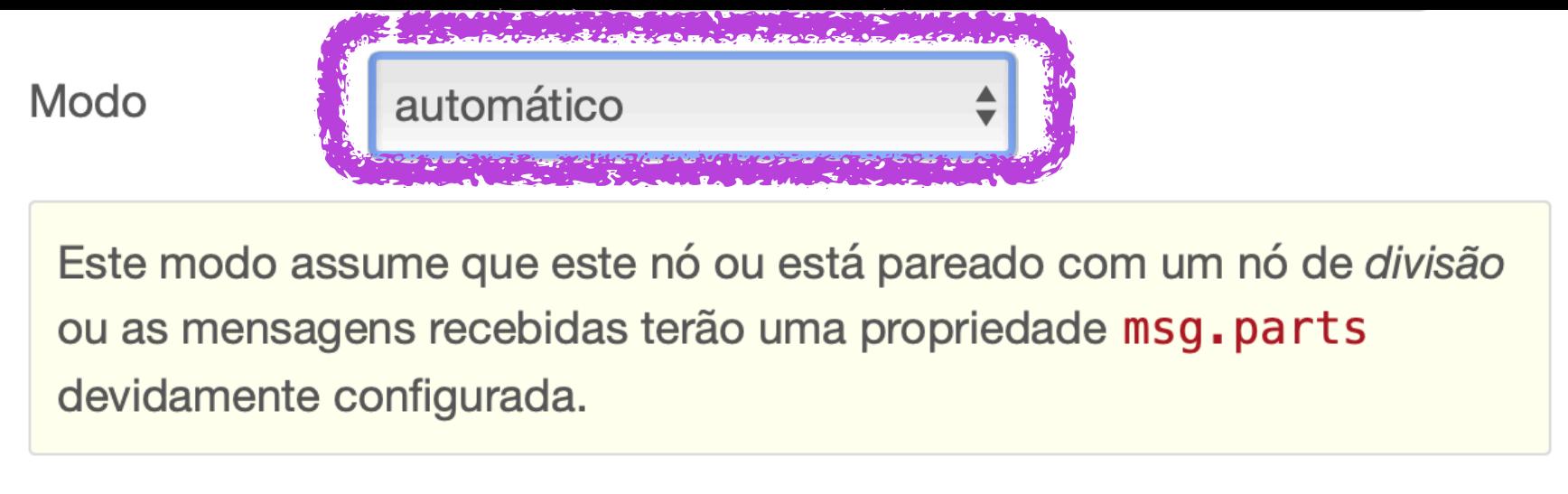
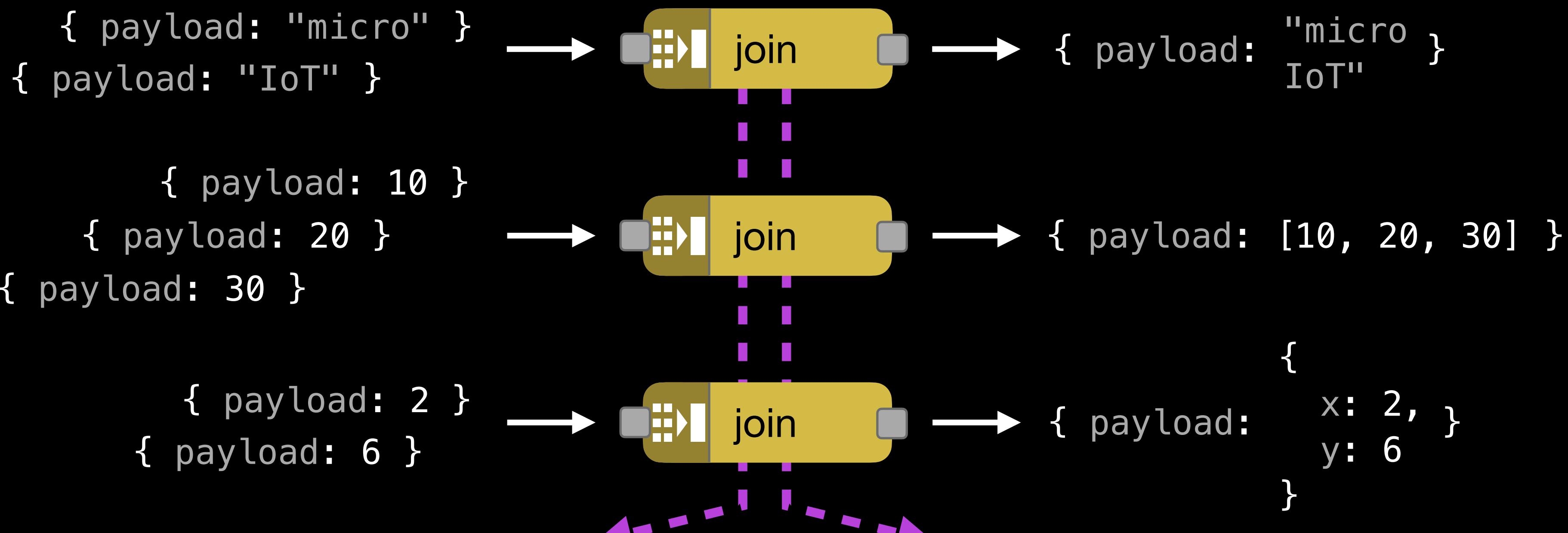
{ payload: 8 } → switch (não envia nada)  
(não envia nada) → { payload: 8 }

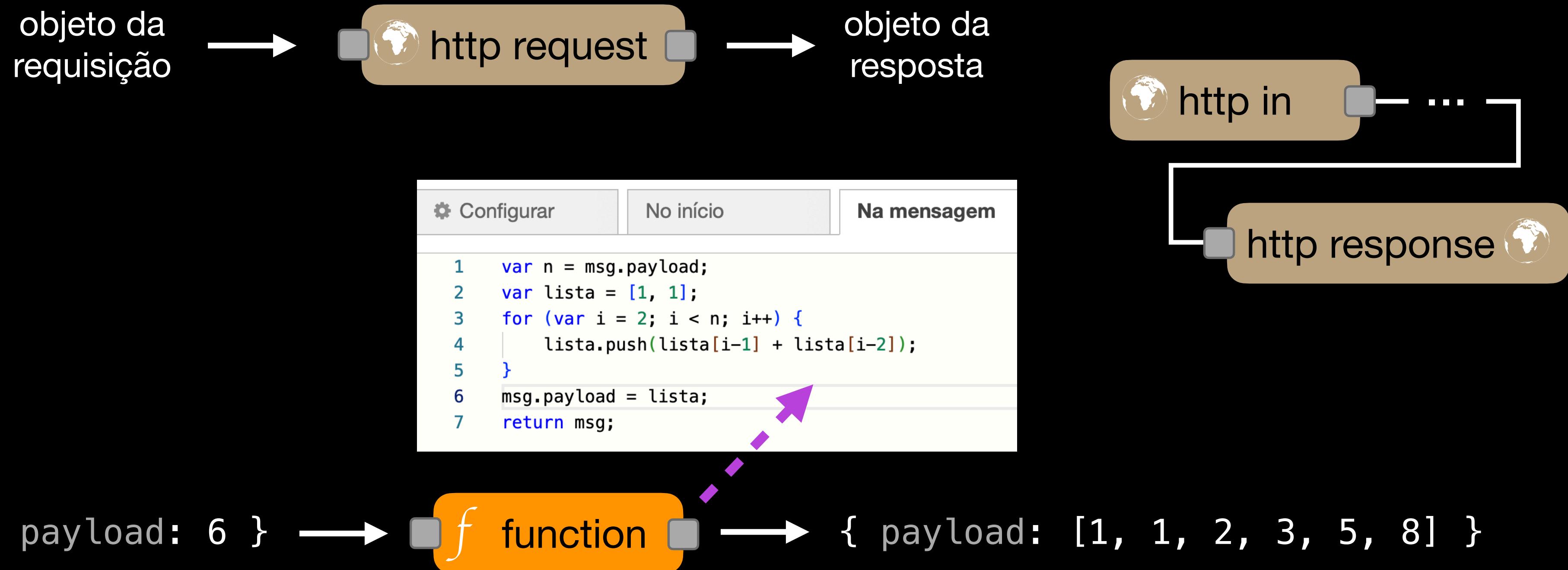
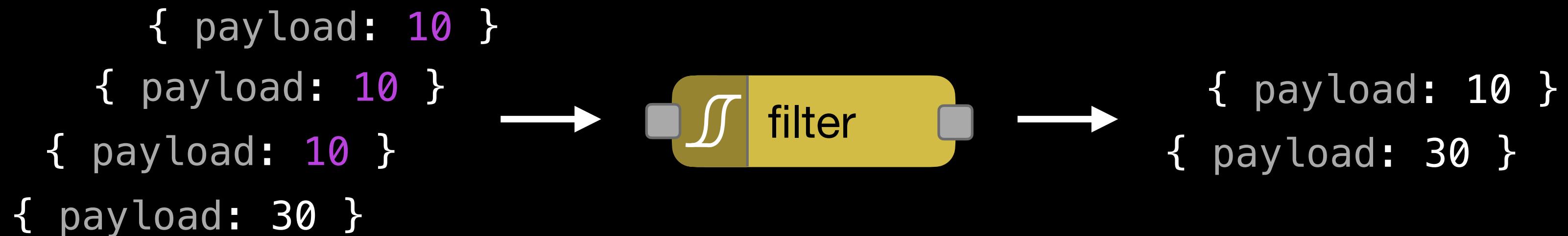
{ payload: "micro" } → split → { payload: "micro" }  
IoT

{ payload: [10, 20, 30] } → split → { payload: 10 }  
{ payload: 20 }  
{ payload: 30 }

{  
{ payload: { x: 2, } } → split → { payload: 2 }  
y: 6 } → { payload: 6 }  
}







Configurações do usuário

Fechar

Visão Nós Instalar

Node-RED Community catalogue ordenar: JF A Z

Paleta

Teclado

Environment

search: telegram 15 / 5044

node-red-contrib-telegrambot node-red-contrib-telegrambot-home

instalar instalado

Procurar fluxos

Configuração dos nós

Fluxos

subfluxos

Grupos

Gerenciar paleta

Configurações

Atalhos do teclado

sítio do Node-RED

v3.1.0

Configurações do usuário

Visão Nós Instalar

Node-RED Community catalogue ordenar: JF A Z

Paleta

Teclado

Environment

search: telegram 15 / 5044

node-red-contrib-telegrambot node-red-contrib-telegrambot-home

instalar instalado

Procurar fluxos

Configuração dos nós

Fluxos

subfluxos

Grupos

Gerenciar paleta

Configurações

Atalhos do teclado

sítio do Node-RED

v3.1.0

telegram

- receiver
- command
- event
- sender
- reply
- control



Olá, Node-RED

receiver

{  
chatId: 123456789,  
messageId: 1234,  
type: "message",  
content: "Olá, Node-RED!",  
date: 1725765672

**Propriedades**

**Bot**

Adicionar novo telegram bot...

**Bot-Name** Meu Bot do Telegram

**Token** CHAVE SECRETA QUE O BOT FATHER TE DEU

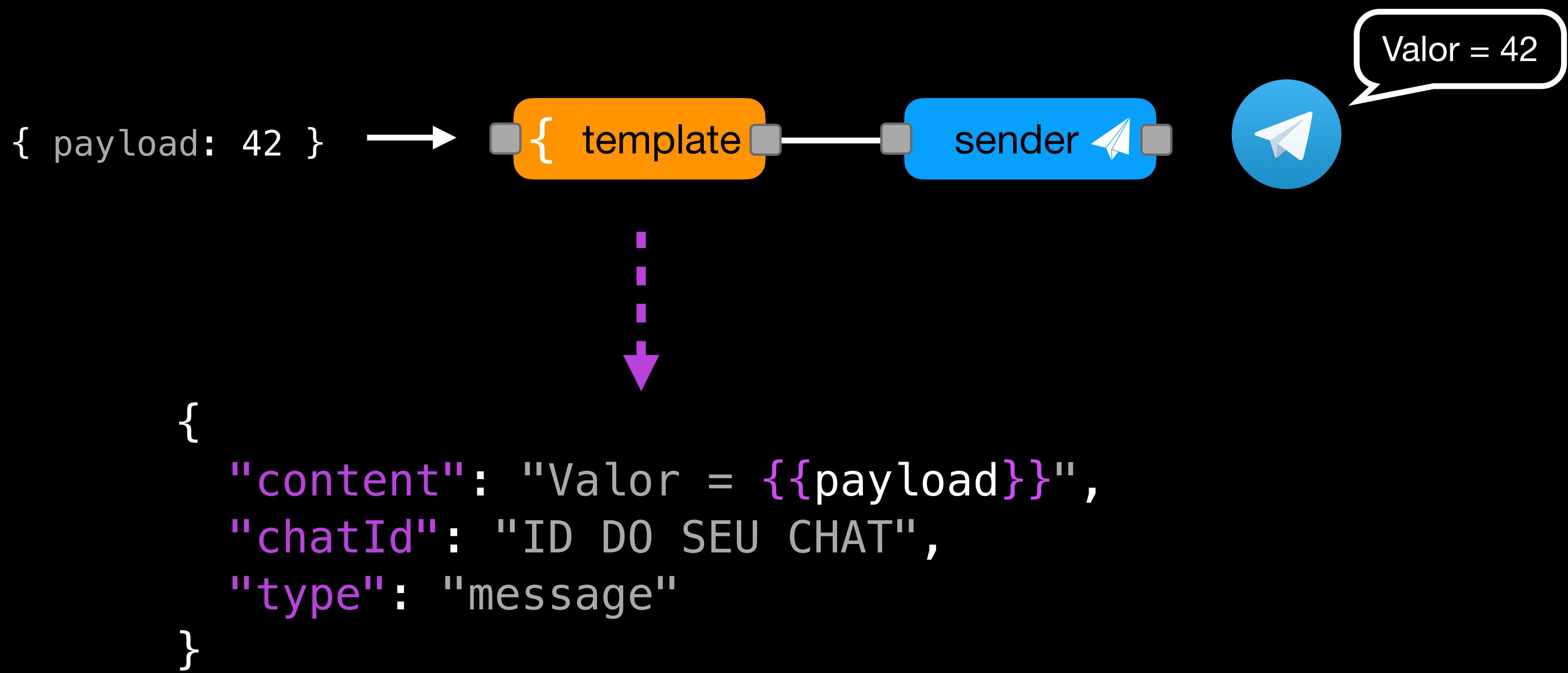
**Tip:** If you don't have a token yet, you can create a new one here:  
[@BotFather.](#)

**Users** (Optional list of authorized user names e.g.: hugo,sepp,egon)

**ChatIds** (Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)

A segunda saída do bloco é para mensagens de usuários não-autorizados.





{ payload: <dados binários de imagem> }



habilita/desabilita  
visualização



... Property msg. payload

↔ Width 160

Resize images on server side

Allow image passthrough

### dashboard 2

- form
- text input
- file input
- button
- button group
- dropdown
- radio group
- slider
- switch
- text A
- table
- chart
- gauge
- notification
- markdown
- template
- event
- ui control

### depurar

- Informação
- Ajuda
- mensagens de depuração
- Configuração dos nós
- Contexto dos Dados
- Dashboard 2.0

**Table Cell Types**

My Column	My Other Column ↓	URL	Bar	Row	Trend
Hello	World	<a href="https://news.bbc.co.uk">https://news.bbc.co.uk</a>		1	
Hey	Globe	<a href="https://flowfuse.com">https://flowfuse.com</a>		2	

**chart**

**Built-in Form Example**

Name: Firstname Surname

Checkbox

**SUBMIT** **CLEAR**