

# PET Lab

Fatec  
Jundiaí  
Deputado Ary Fossen

Projetos Educacionais e de Tecnologia



# Caninos Loucos Labrador

# Caninos Loucos Labrador

20/06/2024

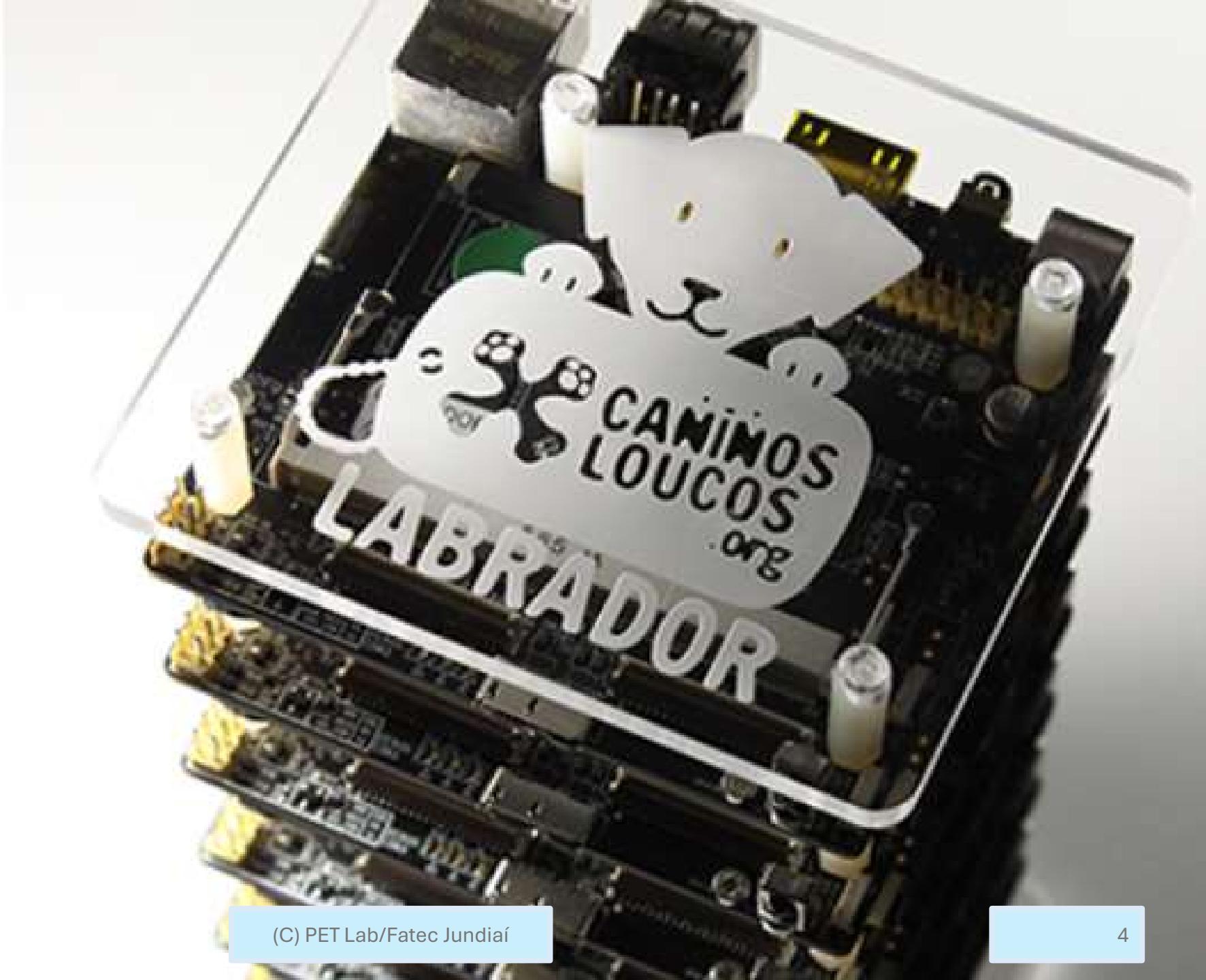
## IoT e MQTT

## Node RED

(C) PET Lab/Fatec Jundiaí

## Mãos na Massa

## Próximos Passos



# Caninos Loucos Labrador

- Single Board Computer brasileira, com projeto de hardware e software abertos, que pode ser utilizada como um microcomputador ou como plataforma para aplicações de Internet das Coisas, sistemas ciber-físicos, robótica ou monitoramento.

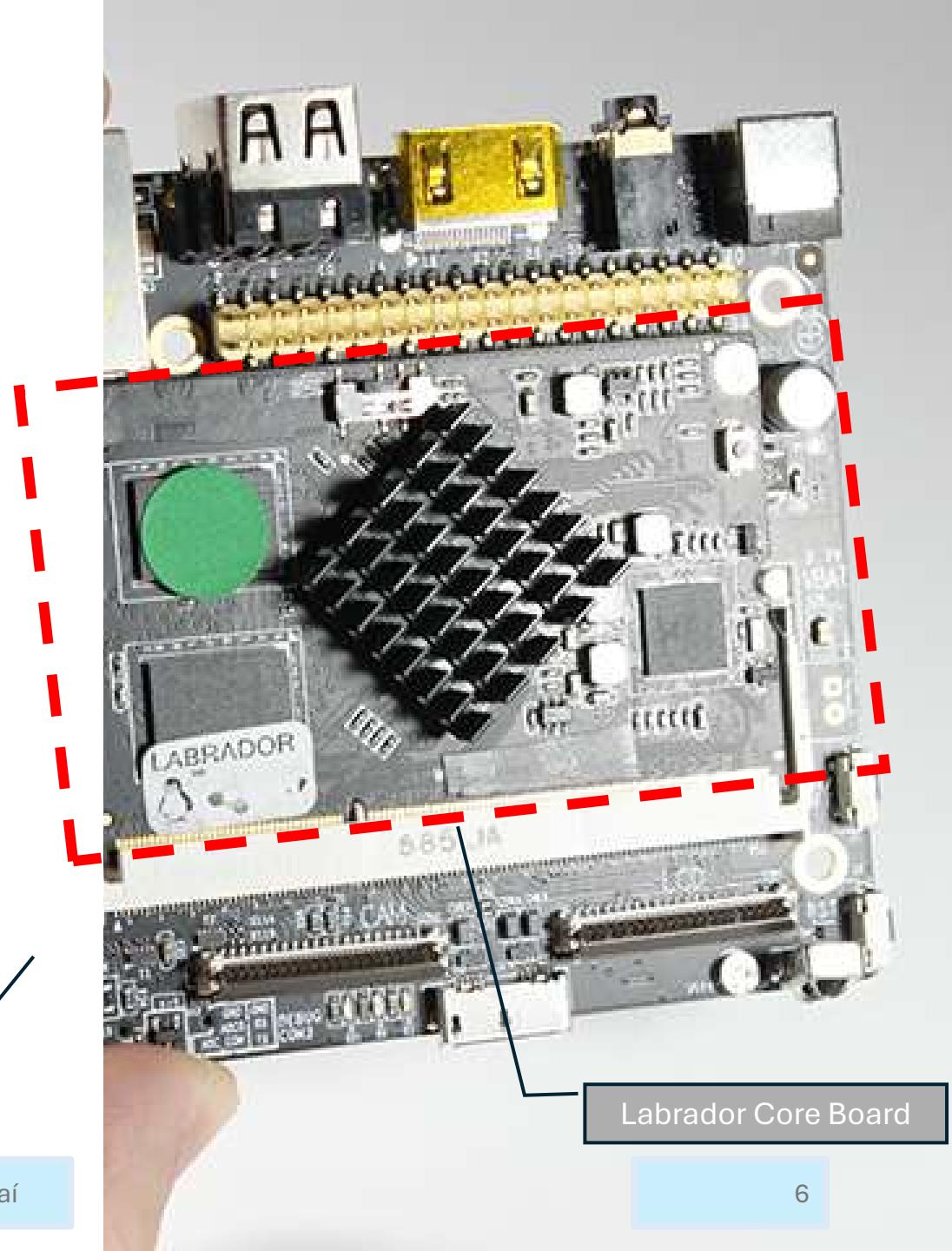


# Caninos Loucos Labrador

- Composta por duas placas:
  - Labrador Core Board, que fornece as funcionalidades de um computador
  - Labrador Base Board, que simplifica e expande as opções de comunicação

Labrador Base Board

(C) PET Lab/Fatec Jundiaí



Labrador Core Board

# Labrador Core Board v.2 [características]

- CPU Quad-core A9R4 ARM Cortex 1,3GHz (ARM v7 instruction set)
- GPU Imagination PowerVR SGX544 com suporte OpenGL (ES 1.1/2.0, OpenGL 1.2.1, OpenCL 1.1)
- Memória: 2GB DDR3 SDRAM
- Armazenamento: 16GB eMMC
- SOs: Android 5.0, Linux Kernel 4.14.13, Debian 10+
- Gerenciamento de energia e subsistema de áudio ATC2306C
- Vídeo 1080p@60fps com suporte a codificação H264, H263, MPEG-4
- Expansão via conector DDR3 SODIMM 204 pinos (macho)
- Dimensões 67.6 x 42.2 mm (13.6g)

# Labrador Base Board v.2 [características]

- Armazenamento: MicroSD Card Slot SD/SDHC/SDXC (até 32GB)
- Ethernet 10/100Mbps (RJ45)
- Wireless Wi-Fi 802.11 b/g/n 2.4GHz, Bluetooth 5.0 LE e receptor infravermelho (38kHz)
- USB: (2x) 2.0 HOST (tipo A), USB3.0 OTG (micro-B)
- Display:
  - HDMI 1.4 (tipo A), até 1920x1080@60Hz
  - LVDS-DSI para LCD até 1920x1080@60Hz
  - CVBS PAL/NTSC
- Áudio: saídas no HDMI e analógica estéreo; I2S entrada/saída; e microfone embutido
- Câmera: MIPI-CSI e interface paralela de 8-bits
- LED: ligar/desligar (vermelho), programável (verde e azul)
- Botões: ligar/desligar, reiniciar, programável
- Alimentação: 5~12V@2<sup>a</sup>
- Expansão:
  - conector SODIMM de 204 pinos (fêmea)
  - 40-pin header: 28 GPIOs (compatível com Raspberry Pi B+ com suporte UART, I2C, SPI, PWM, I2S e ADC input)
- Debug: UART
- Dimensões: 92.7mm x 88mm (59,4g)

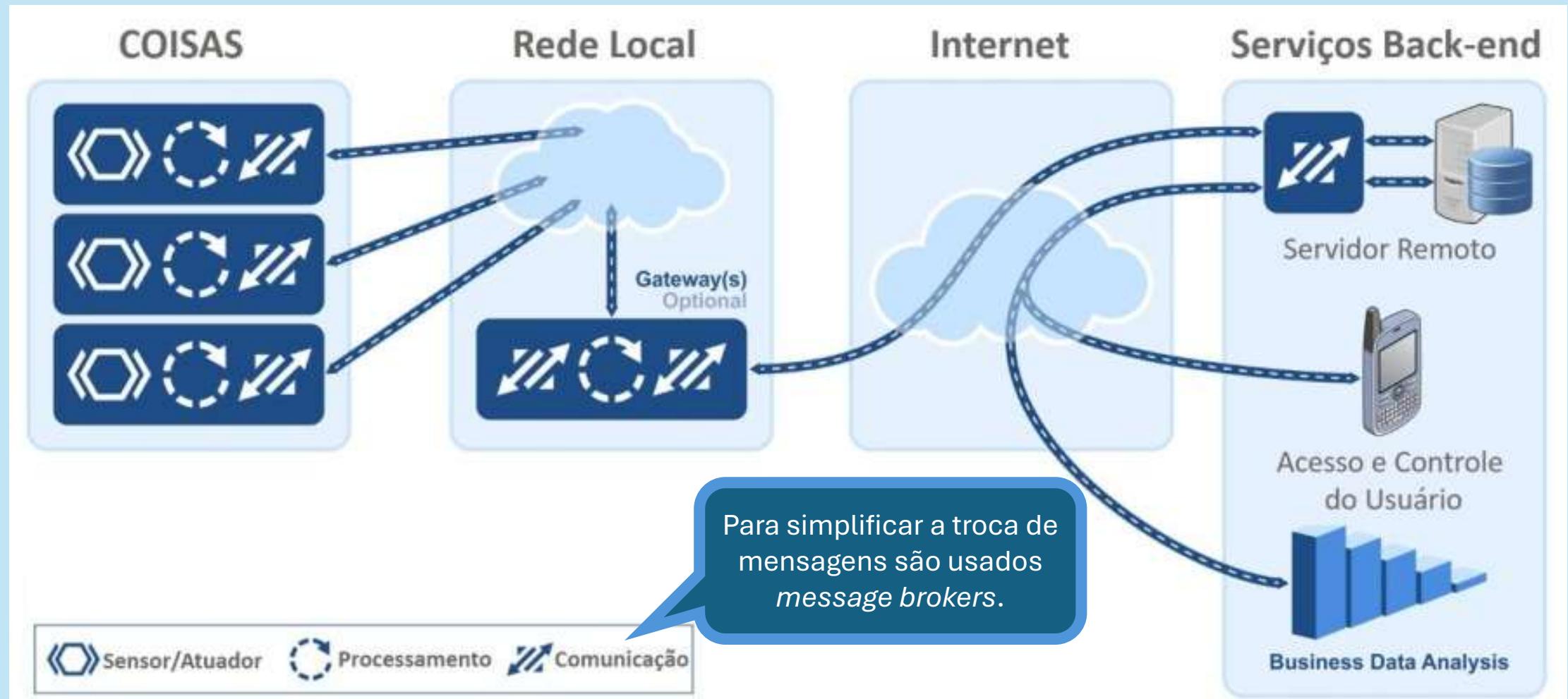
# Quadro Comparativo

Recurso	Placa Labrador v2 32 bits	Raspberry Pi 4
Processador	Quad-core A9R4 ARM Cortex (Arm v.7) 1,3 GHz	Quad-core A72 ARM Cortex (Arm v.8) 1,8 GHz
RAM	2 GB DDR3 SDRAM	1, 2, 4 ou 8 GB DDR4 SDRAM
Armazenamento	Interno 16GB eMMC micro SD Card (até 32 GB)	Micro SD Card (até 32 GB)
Conectividade	Wi-Fi IEEE 802.11 bgn, Bluetooth 5.0 LE, Ethernet, Infrared	Wi-Fi IEEE 802.11ac, Bluetooth 5.0 BLE, Ethernet
Conexões	1 x USB 3.0, 2 x USB 2.0	2x USB 3.0, 2 x USB 2.0
Vídeo	1 HDMI 1920p@60fps suporte H.264, H.263, OpenGL ES 2.1	2 HDMI 4k@60fps suporte H.265, H.265, OpenGL ES 3.1
SO	Linux, Debian, Android	Linux, Debian, Raspbian

---

# IoT e MQTT

# IoT::Internet das Coisas



# *message broker*

Um *message broker* ou **mom** (*message oriented middleware*) é um servidor de mensageria, ou seja, é um elemento da infraestrutura idealizado unicamente para processar e suportar o envio, recebimento, redirecionamento e também o monitoramento das mensagens compartilhadas entre os sistemas integrados por mensagem.



- É um protocolo leve, para a troca de informações máquina a máquina, que funciona no modo *publish-subscribe*.
- Projeto para conexão com dispositivos de capacidade ou comunicação limitada (como os usados em aplicações de Internet das Coisas – IoT).
- É um padrão OASIS e uma recomendação ISO (ISO/IEC 20922).
- Define dois tipos de entidades:
  - Um *message broker*, responsável por receber(, armazenar) e divulgar dados de clientes;
  - Qualquer número de clientes que fornecem ou consomem dados.

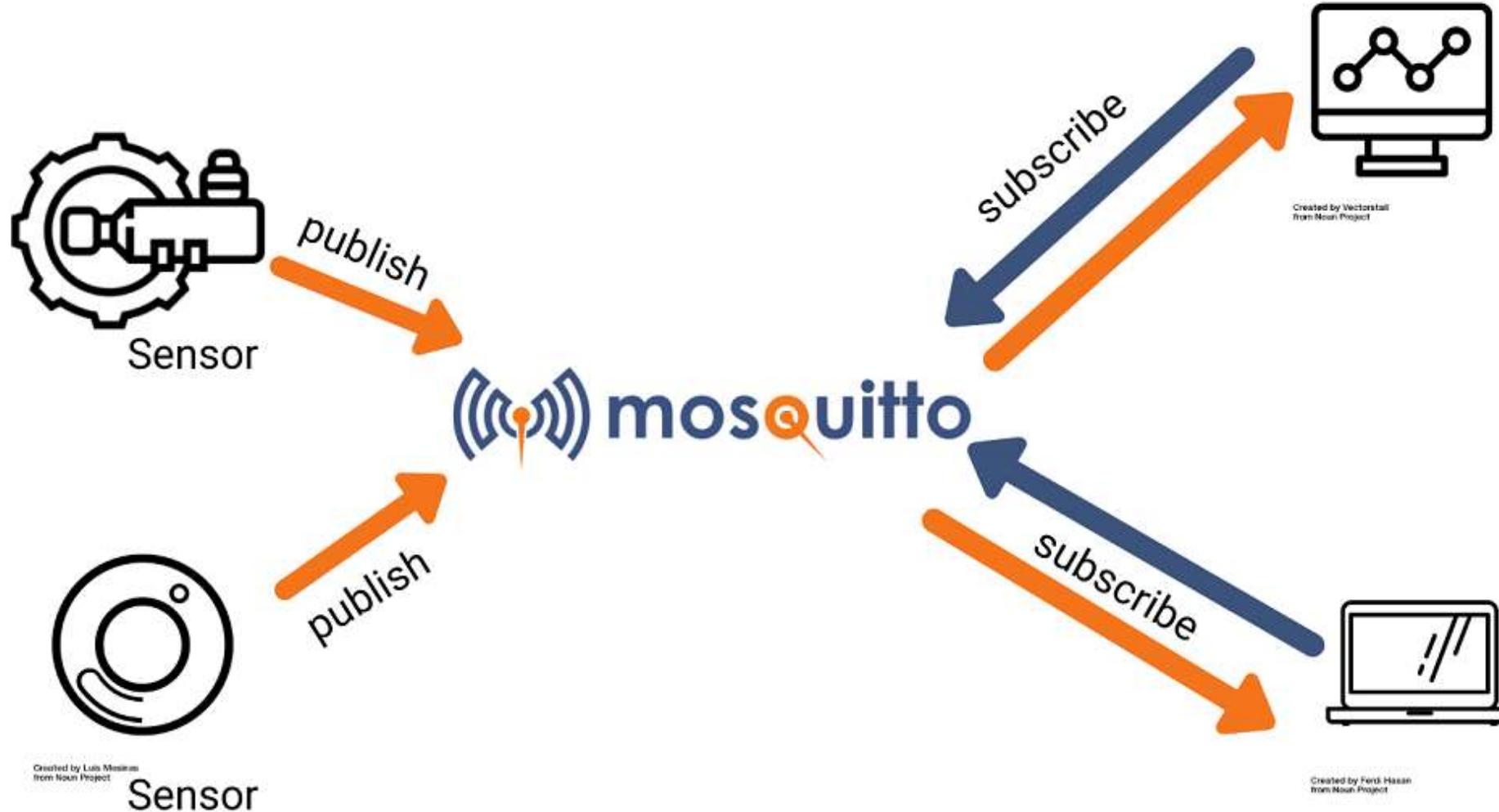


<https://mqtt.org/>

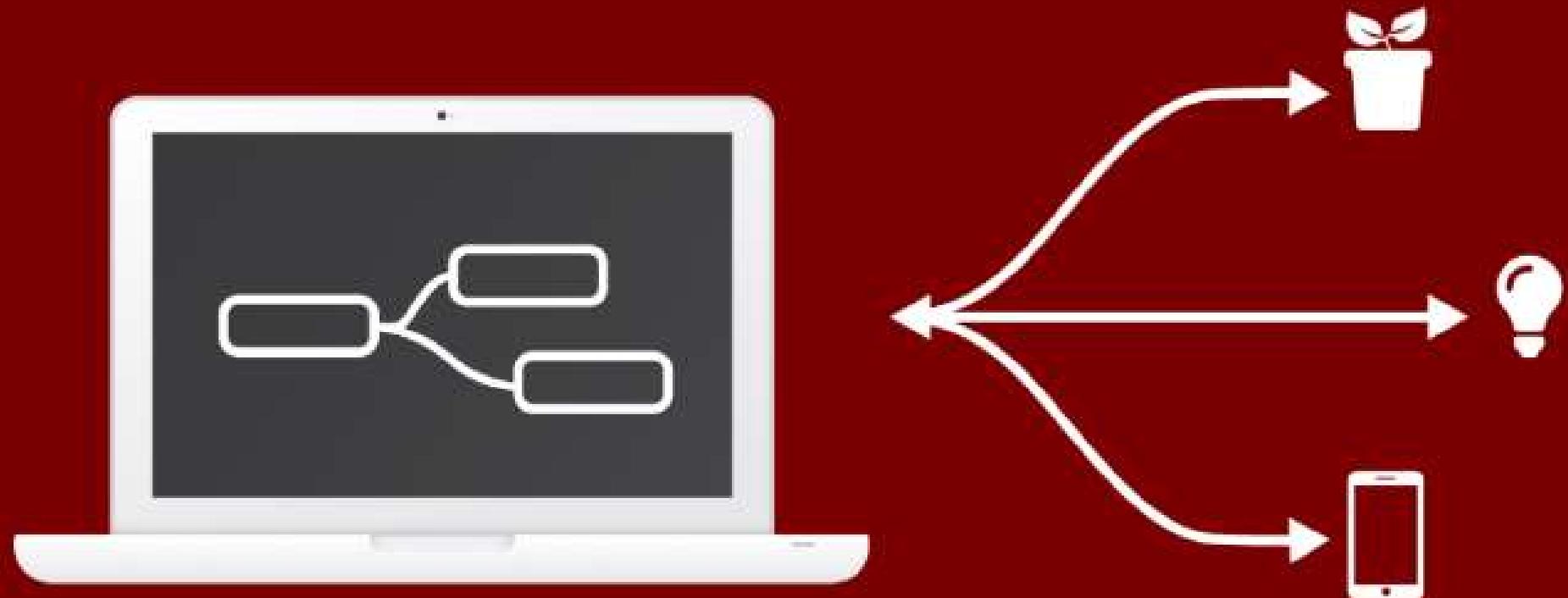
# Arquitetura



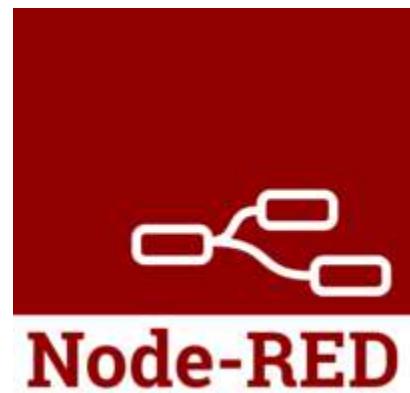
- É um *message broker* de código aberto, que implementa o protocolo MQTT (versões 5, 3.1.1 e 3.1).
- É bastante leve, adequado para computadores de placa única até servidores, o que o torna adequado para mensagens da Internet das Coisas, como sensores de baixa potência ou dispositivos móveis, como telefones, computadores incorporados ou microcontroladores.
- Inclui uma biblioteca C para construção de clientes MQTT; os clientes populares de linha de comando *mosquitto\_pub* e *mosquitto\_sub*; além de implementações em outras linguagens.



# Arquitetura



# Node-RED



- É uma ferramenta de desenvolvimento *low-code* baseada na programação visual de fluxos (de dados).
- Desenvolvida pela IBM (2013) para conectar dispositivos de hardware, APIs e serviços online como parte da Internet das Coisas.
- Em 2016 tornou-se um projeto de código aberto da OpenJS Foundation.
- Fornece um editor de fluxo baseado em navegador da web, usado para criar funções JavaScript, cuja execução é realizada pelo Node.js.
- Seus fluxos são armazenados usando formato JSON e seus elementos podem ser salvos ou compartilhados para reutilização.



- Formados pela conexão e sequenciamento de vários nós, onde cada nó de um fluxo executa uma tarefa específica dentro da plataforma.
- Os dados transmitidos para um nó são processados de acordo com a função designada e, depois, são passados para o nó subsequente no fluxo.

Tecnicamente, um fluxo consiste em uma lista de objetos JavaScript que descrevem os nós e suas configurações.

## Node-RED::fluxos

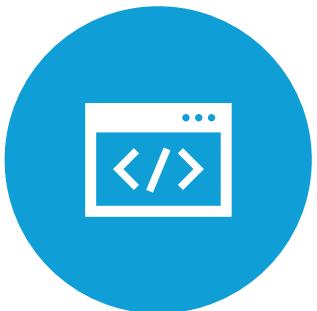
# Node-RED::nós



São os blocos construtivos do Node-RED.



Durante a execução dos fluxos, os nós geram, processam e consomem mensagens.



As mensagens transmitidas entre os nós são objetos JavaScript denominados **msg** que contém várias propriedades, entre elas, **msg.payload** (carga útil da mensagem).



Propriedades específicas podem ser adicionadas aos objetos **msg**, possibilitando o transporte de dados específicos da aplicação.

# Node-RED::nós

## Origem

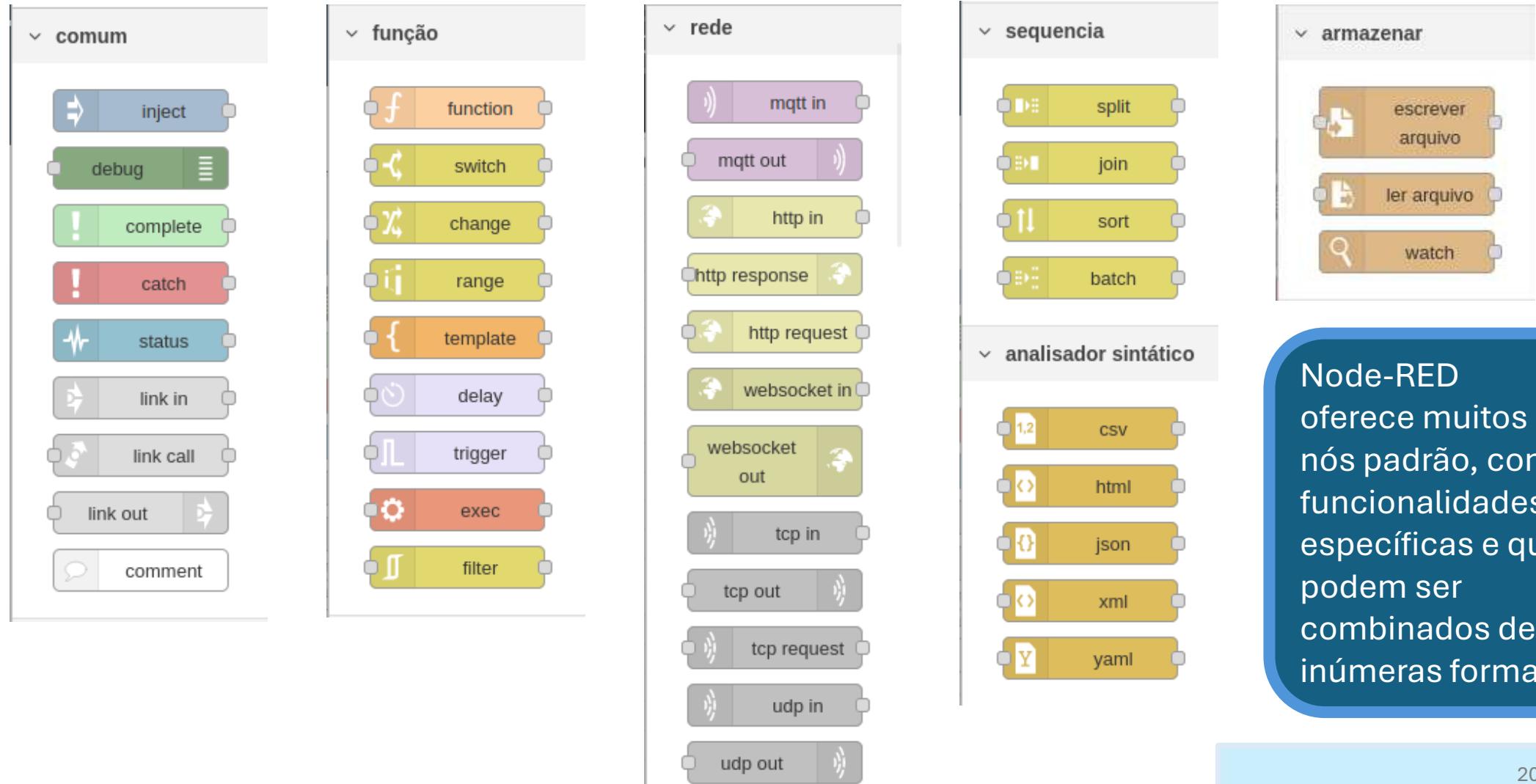
- Nós **padrão** (*default*) da plataforma e
- Nós de **contribuição**, criados pelo usuário/programador e que podem formar uma biblioteca compartilhada.

Isto permite a execução controlada de uma ampla gama de operações, além de oferecer muita flexibilidade na criação de aplicações em tempo real.

## Natureza

- Nós de **Entrada**
  - iniciam um fluxo, a partir de dados internos ou externos à plataforma;
- Nós de **Processamento**
  - realizam transformações específicas sobre os dados; e
- Nós de **Saída**
  - finalizam um fluxo com a exibição de dados ou o envio destes para plataformas externas.

# Node-RED:nós



Node-RED oferece muitos nós padrão, com funcionalidades específicas e que podem ser combinados de inúmeras formas.

# Node-RED::interface

The screenshot illustrates the Node-RED interface with several key components:

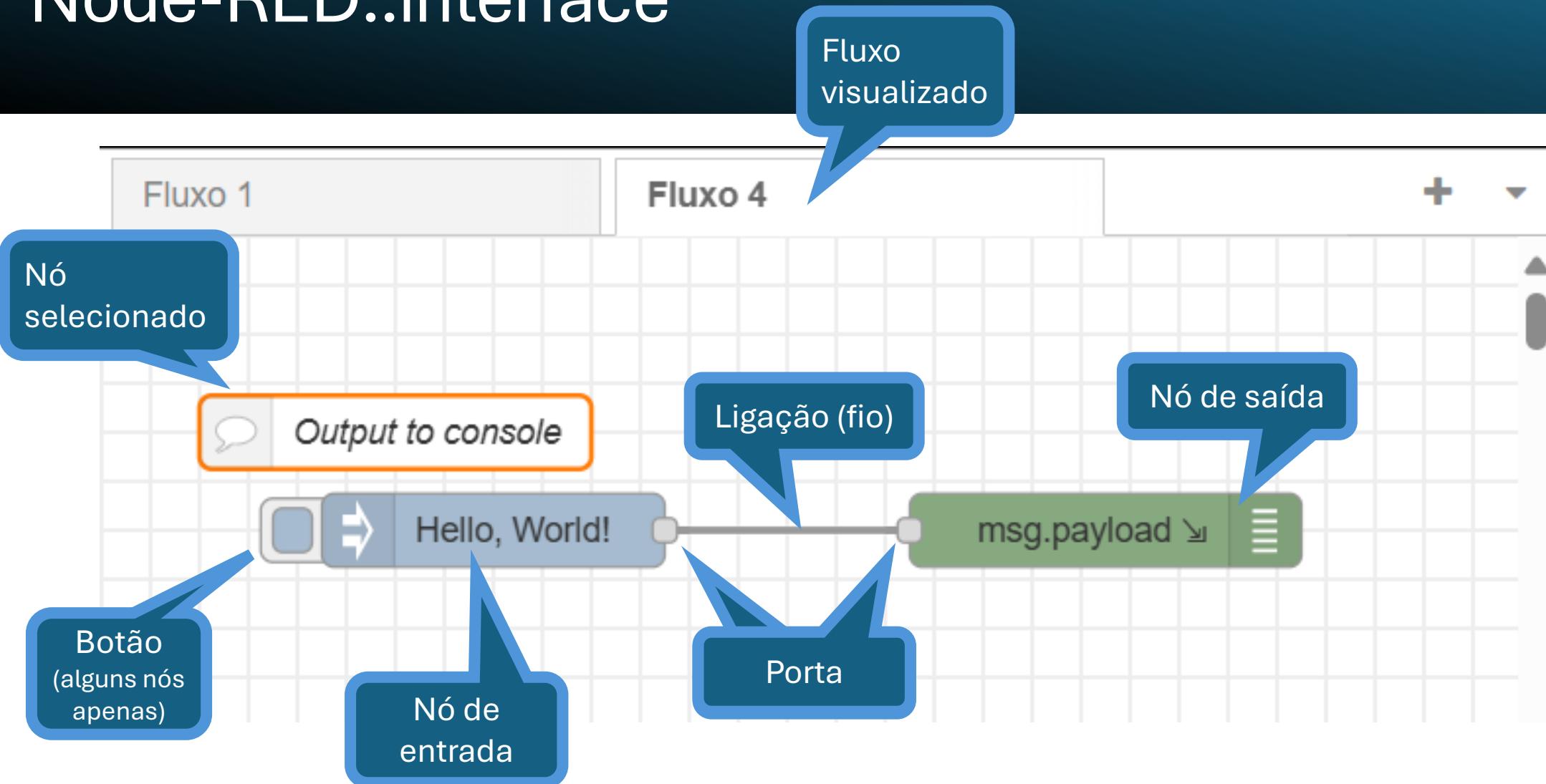
- Paleta de Nós (Palette):** Located on the left, it contains a search bar and two sections: "comum" and "função". The "comum" section includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The "função" section is currently collapsed.
- Editor de fluxos (Flow Editor):** The main workspace shows a flow titled "Fluxo 4". It starts with an "inject" node, followed by a "debug" node with the label "Hello, World!". A connection leads to a "msg.payload" node. An "Output to console" node is also present in the flow.
- Detalhes da interface (Details Panel):** This panel on the right provides information about the selected node ("Output to console"). It includes tabs for "informações" (Information), "Ajuda" (Help), "Depuração" (Debug), "Configurações" (Configurations), and "Contexto" (Context). The "informações" tab displays the node's ID ("fb1c3ce9.c29ee") and type ("comment"). The "Ajuda" tab contains a description: "Debug node can be used to output values to console."

Paleta de  
Nós

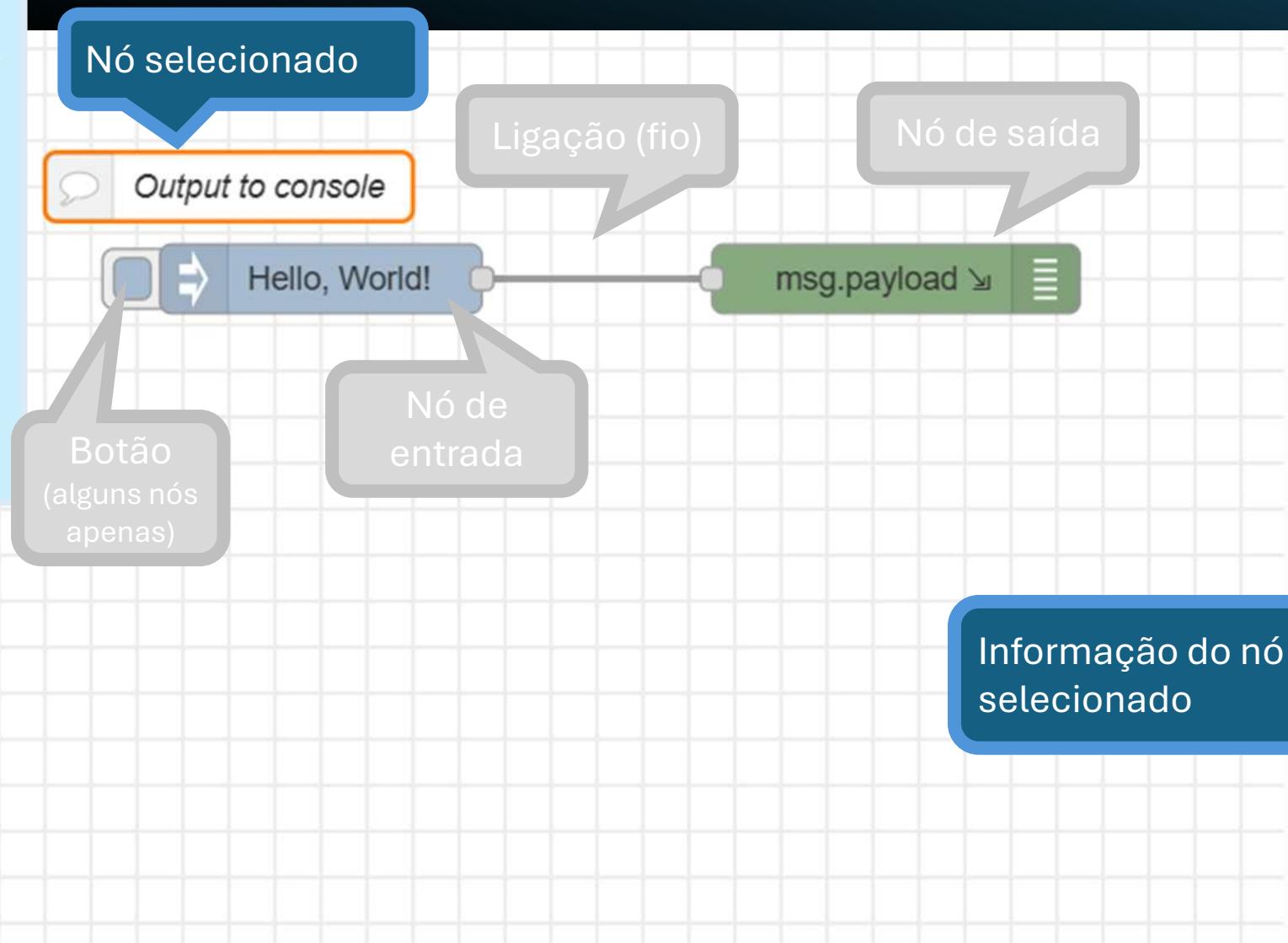
Editor de  
fluxos

- Abas de:
- Informações
  - Ajuda
  - Depuração
  - Configurações
  - Contexto

# Node-RED::interface



# Node-RED::interface



The sidebar shows the selected node configuration:

- Fluxos** (Flows):
  - Fluxo 1
  - Fluxo 4 (selected)
- Output to console** (Selected node):
  - Type: comment
  - Node ID: "fb1c3ce9.c29ee"
- Output to console** (Description):
 

Debug node can be used to output values to console.

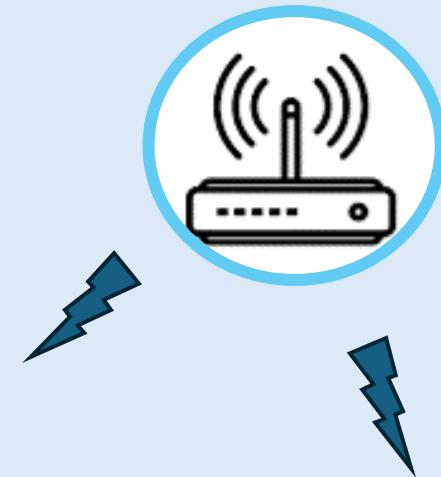


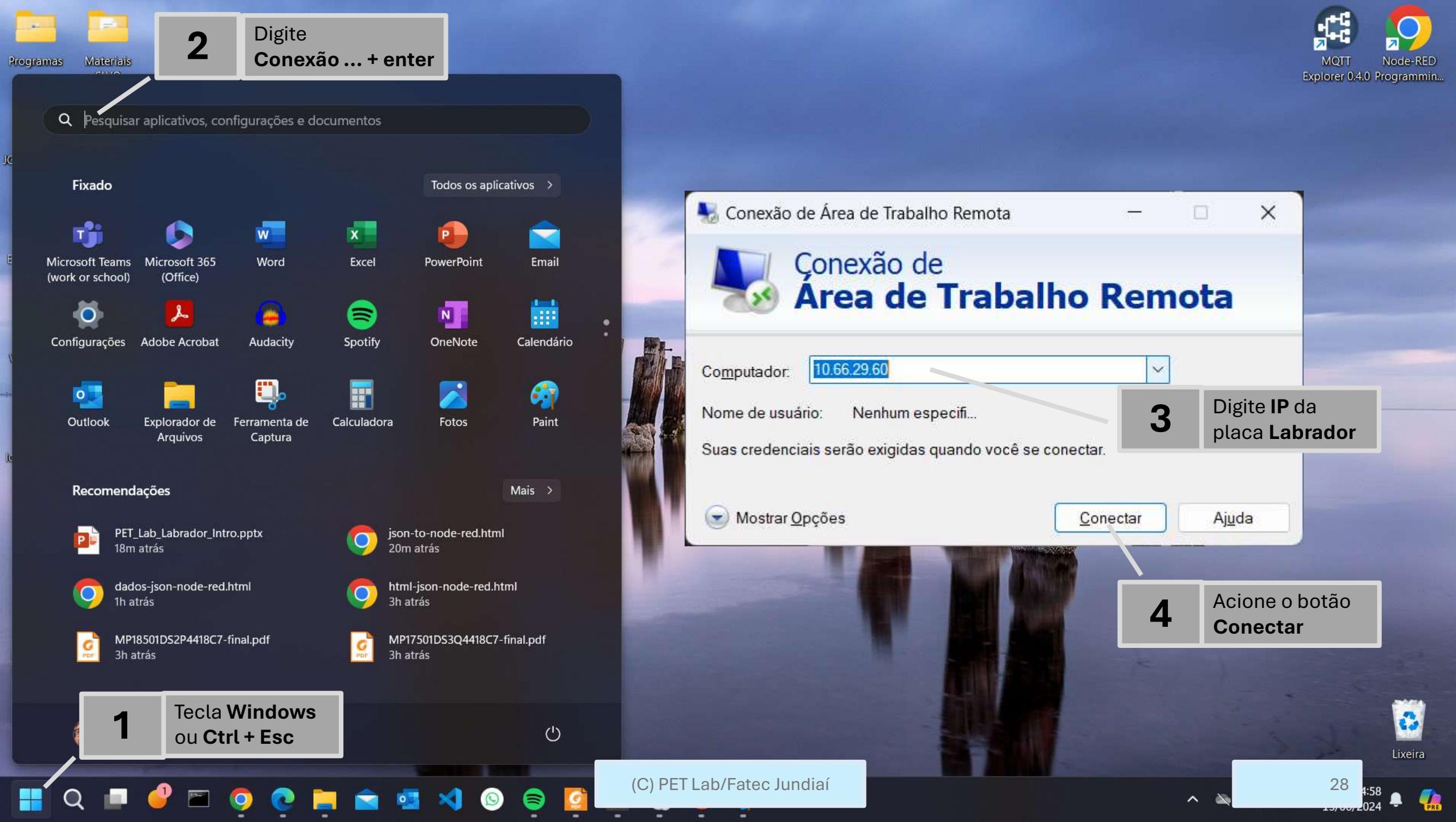
# Hands-On

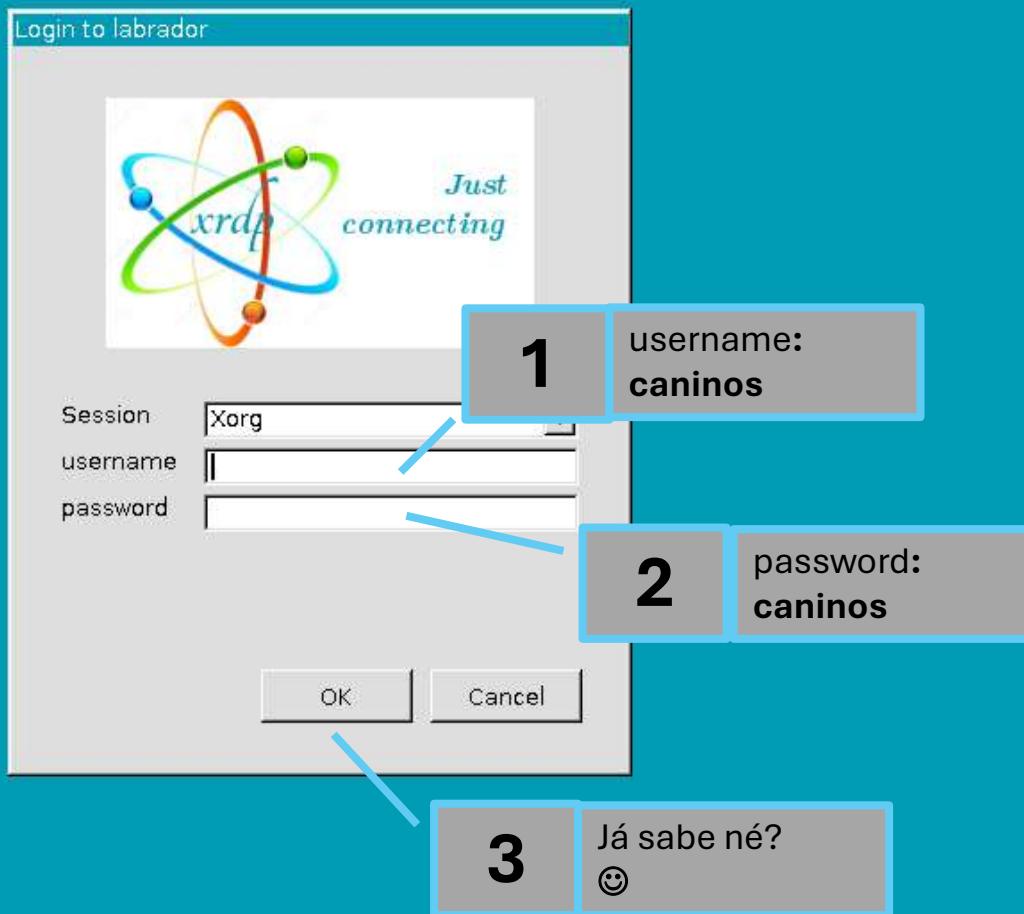
# Cenário & Roteiro



1. Setup inicial das placas Labrador:
  - SO Debian 11
  - Mosquitto
  - Node.js + Node-RED
  - xRDP
2. Conexão (via RDP) entre Notebook/Windows e Labrador/Debian
3. Utilização do Node-RED







Desktop  
semelhante ao  
Windows

1

Abra um  
LXTerminal

2

Digite comando  
**node-red**



Lixeira



caninos@labrador...

**1**

Minimize, mas  
não feche!

```
caninos@labrador: ~
Arquivo Editar Abas Ajuda
caninos@labrador: $ node-red
13 Jun 16:34:22 - [info]

Welcome to Node-RED
=====

13 Jun 16:34:22 - [info] Node-RED version: v3.1.9
13 Jun 16:34:22 - [info] Node.js version: v21.7.3
13 Jun 16:34:22 - [info] Linux 5.10.191 arm LE
13 Jun 16:34:26 - [info] Loading palette nodes
13 Jun 16:34:31 - [info] Settings file : /home/caninos/.node-red/settings.js
13 Jun 16:34:31 - [info] Context store : 'default' [module=memory]
13 Jun 16:34:31 - [info] User directory : /home/caninos/.node-red
13 Jun 16:34:31 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Jun 16:34:31 - [info] Flows file : /home/caninos/.node-red/flows.json
(node:1760) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
13 Jun 16:34:32 - [info] Server now running at http://127.0.0.1:1880/
13 Jun 16:34:32 - [info] Starting flows
13 Jun 16:34:32 - [info] Started flows
13 Jun 16:34:32 - [info] [mqtt-broker:Mosquitto] Connected to broker: mqtt://localhost:1883
```

Node-RED deve  
ser carregado,  
indicando conexão  
com broker  
Mosquitto

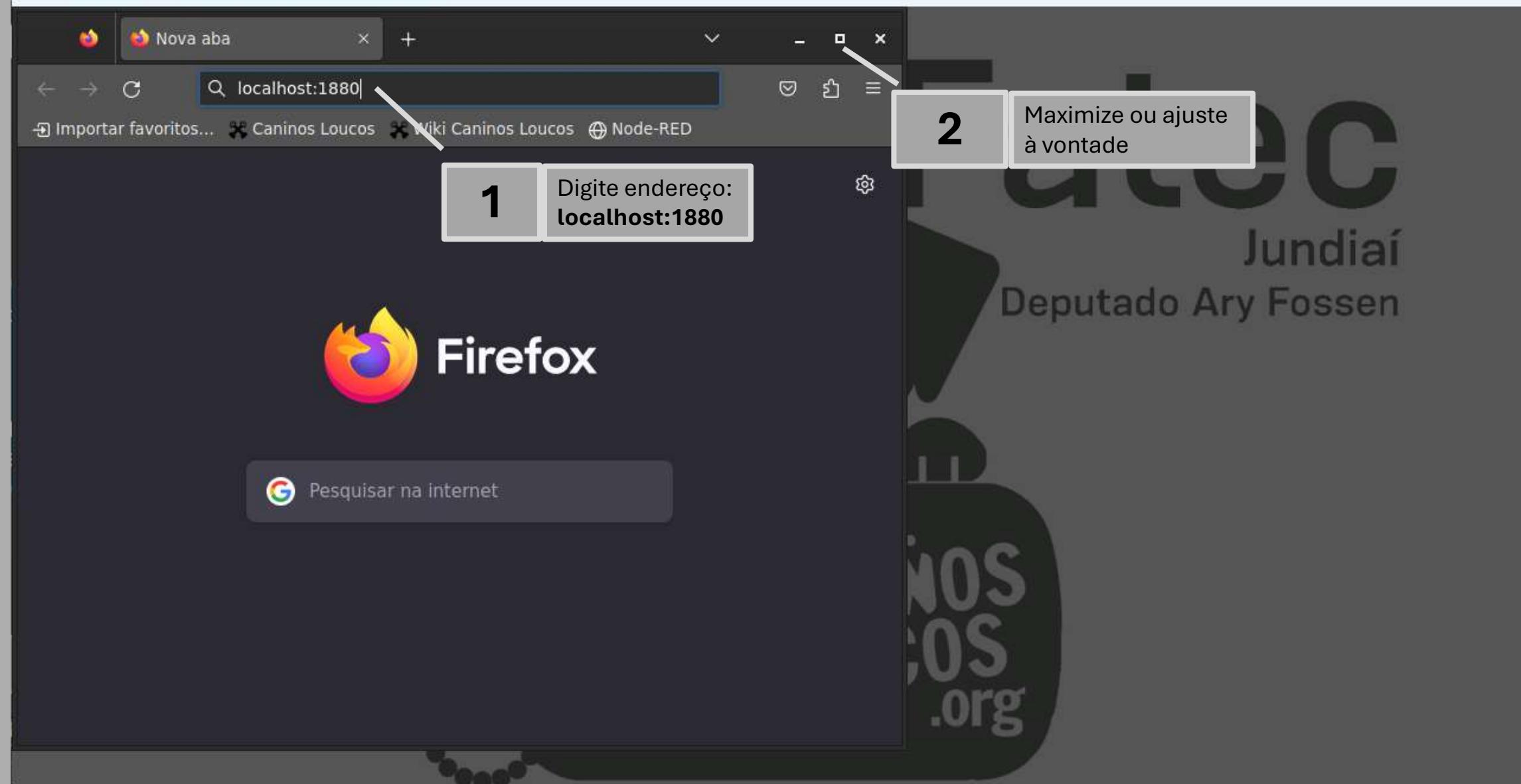
**2**

Abra o Firefox  
(navegador web)



Lixeira





20/06/2024



[caninos@labrad...

Mozilla Firefox

(C) PET Lab/Fatec Jundiaí

32

Node-RED

localhost:1880

Node-RED

Node-RED pode solicitar suas credenciais para completar o carregamento

The screenshot shows a Node-RED login interface. On the left is a red logo with the text "Node-RED". To its right are two input fields: "Nome do Usuário:" and "Senha:", each with a corresponding red box containing a number and text. Red arrows point from these boxes to the respective input fields. Below the inputs is a button labeled "Ingressar". A blue speech bubble on the left contains the text "Node-RED pode solicitar suas credenciais para completar o carregamento".

Nome do Usuário:  
caninos

Senha:  
caninos

Ingressar

1 Nome do Usuário:  
caninos

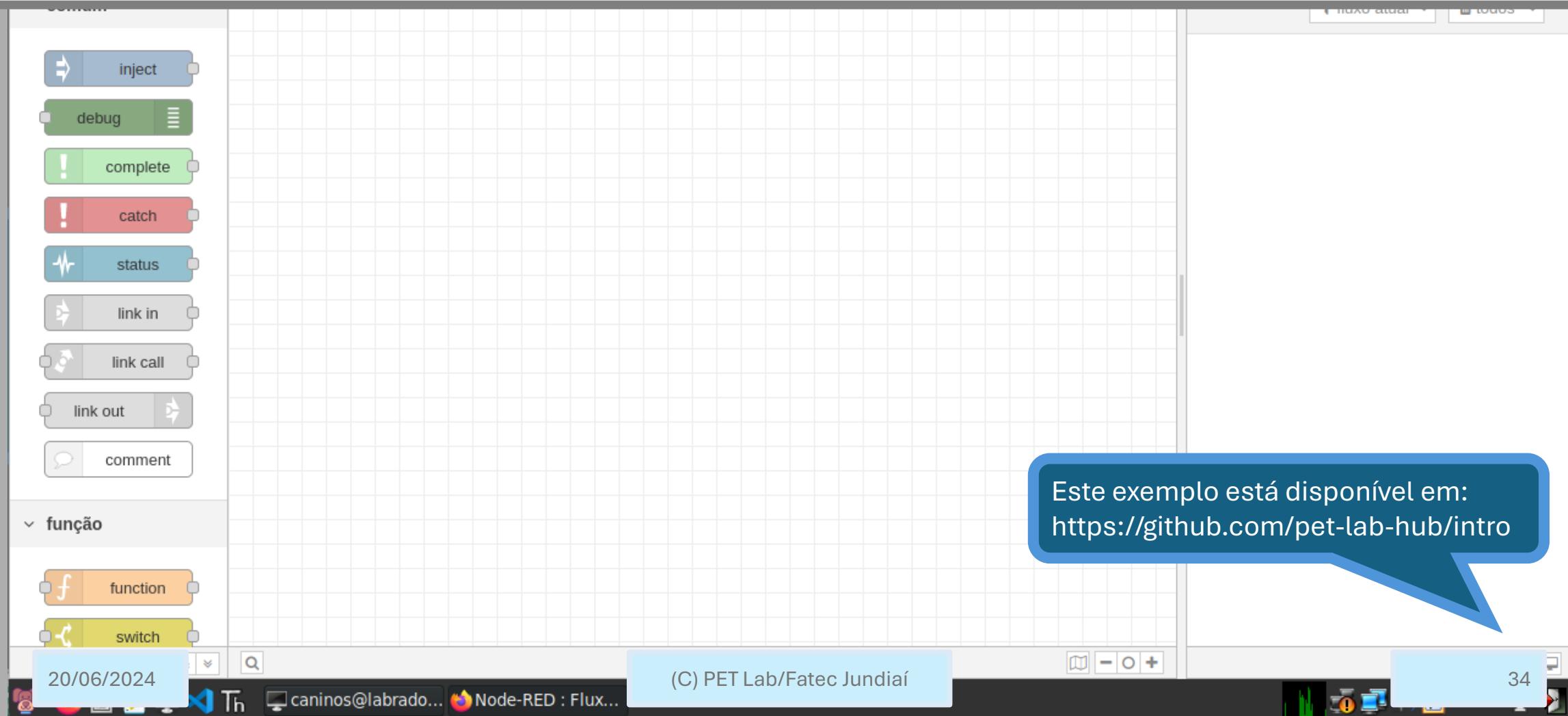
2 Senha:  
caninos

3 Já sabe né?  
😊

(C) PET Lab/Fatec Jundiaí

33

# Exemplo 1 [básico]



Node-RED : Fluxo 1 +

localhost:1880/#flow/a15badd50f929050

Node-RED

implementar depurar

filtrar nós

comum

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

função

- function
- switch

Fluxo 1

2 Clique duplo para editar o nome do fluxo

Um clique duplo permite a edição das propriedades de fluxos e de nós do Node-RED.

1 Acione + para acrescentar um novo fluxo

```
graph TD; subgraph F1 [Fluxo 1]; end
```

10.66.29.60 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/a15badd50f929050

Node-RED

Fluxo 1

Editar Fluxo: Fluxo 1

Deletar Cancelar Feito

Propriedades

Nome: Ola!

Descrição:

1 Um primeiro exemplo de fluxo no  
2 Node-RED para mostrar interacao  
3 simples com usuario.  
4  
5 Utiliza os nos:  
6 - inject  
7

depurar

fluxo atual todos

1 Altere o nome como desejado

2 Opcionalmente forneça uma descrição

As descrições podem ser feitas com texto puro ou usando formato *markdown* (um html simplificado).

3 Acione botão Feito para salvar e voltar ao fluxo

20/06/2024

(C) PET Lab/Fatec Jundiaí

Unlocked

36

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/78ea719049bf878e

90%

Node-RED

implementar

informações

Procurar fluxos

Fluxos

Ola!

subfluxos

Nós de configuração global

Árvore do fluxo

20/06/2024

(C) PET Lab/Fatec Jundiaí

37

Ponto azul indica alterações não salvas

1 Acrescente um nó inject

2 Acrescente um nó debug

Selezione o nó desejado na paleta e arraste-o para a posição desejada no fluxo.  
Repita o processo tantas vezes quanto necessário.  
O posicionamento dos nós é livre na área de edição.

Descrição do elemento selecionado aparece na aba de informações

192.168.0.138 - Conexão de Área de Trabalho Remota

### Node-RED : Fluxo 1

localhost:1880/#flow/78ea719049bf878e

90% implementar

Node-RED

filtrar nós

comum

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

função

- function
- switch
- change
- ...

Ola!

Carambo de data/hora

debug 1

informações

Fluxos

- Ola!
- subfluxos
- Nós de configuração global

Procurar fluxos

1 Conecte os nós

Posicione o mouse sobre a **porta de saída** de um nó, crie um fio/conexão arrastando o mouse até a **porta de entrada** de outro nó.

20/06/2024

(C) PET Lab/Fatec Jundiaí

caninos@labrado... Node-RED : Flux...

38

```
graph LR; inject[inject] --> timestamp[Carimbo de data/hora]; timestamp --> debug[debug 1]
```

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/78ea719049bf878e

90%

Node-RED

Ola!

comum

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

funcção

- function
- switch
- change

Ponto azul indica alterações não salvas

A configuração padrão do nó inject é a geração de um timestamp

1 Acione o botão implementar para salvar

2 Acione a seta para escolher modo

3 Opção menos onerosa

Cheio  
Implementar tudo no espaço de trabalho

Fluxos Modificados  
Implantar apenas fluxos que contêm nós alterados

Nós Modificados  
Implantar apenas nós que mudaram

Reiniciar Fluxos  
Reinicia os fluxos atuais implantados

Ola!

Fluxo "78ea719049bf878e"

Um primeiro exemplo de fluxo no Node-RED para mostra interacão simples com o usuário.

Utiliza os nos:

- inject
- debug

20/06/2024

(C) PET Lab/Fatec Jundiaí

39

```
graph LR; inject[inject] --> timestamp[Carimbo de data/hora]; timestamp --> debug[debug 1]
```

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! +

localhost:1880/#flow/78ea719049bf878e

Injetado com sucesso: Carimbo de data/hora

Node-RED

filtrar nós

comum

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

Ola!

1 Acione o nó **inject** pressionando o botão

2 Observe a notificação no console do Node-Red

Carimbo de data/hora

debug 1

3 Seleciona aba **debug** no painel lateral

4 Observe as mensagens recebidas pelo nó **debug**

Normalmente, os nós não retêm informação, ou seja, apenas reagem à ocorrência de uma entrada, produzindo uma saída correspondente.

(C) PET Lab/Fatec Jundiaí

20/06/2024

caninos@labrado... Node-RED : Ola! ...

40

depurar

fluxo atual

filtrado

13/06/2024, 17:37:56 nó: debug 1  
msg.payload : number  
**1718311076057**

13/06/2024, 17:38:03 nó: debug 1  
msg.payload : number  
**1718311083817**

Node-RED : Ola! localhost:1880/#flow/78ea719049bf878e

Node-RED

implementar depurar

fluxo atual filtrado

1 Acrescente um outro nó **inject** como segue

2 Com um clique duplo abra sua edição

3 Indique um **nome** para o nó

4 Selecione **cadeia de caracteres**

5 Digite a mensagem desejada

6 Acione **Feito** para concluir a edição

Existem outras possibilidades de uso deste nó, como atraso na injeção e repetição.

inserir agora

Injetar uma única vez depois 0.1 segundos, depois

C Repetir nenhum

adicionar

iniciar

habilitar

20/06/2024

(C) PET Lab/Fatec Jundiaí

caninos@labrado... Node-RED : Ola! ...

41

The screenshot shows the Node-RED interface with a flow titled 'Ola!' on the canvas. The flow consists of an 'inject' node, a 'Carimbo de data/hora' node, and another 'Carimbo de data/hora' node. A tooltip with six steps guides the user through editing the first 'inject' node:

- Acrescente um outro nó **inject** como segue
- Com um clique duplo abra sua edição
- Indique um **nome** para o nó
- Seleccione **cadeia de caracteres**
- Digite a mensagem desejada
- Acione **Feito** para concluir a edição

The tooltip also notes that there are other ways to use the inject node, such as delaying injection and repetition.

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! + localhost:1880/#flow/78ea719049bf878e

Node-RED

1 Implemente as alterações

2 Experimente a injeção de dados

3 Observe o resultado da injeção

Ola!

comum

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

funcção

- function
- switch

Carimbo de data/hora

Mensagem

Um nó pode receber entradas de um, dois ou mais nós.

depurar

fluxo atual

filtrado

13/06/2024, 18:13:23 nó: debug 1  
msg.payload : string[19]  
"Um ola do Node-RED!"

13/06/2024, 18:13:27 nó: debug 1  
msg.payload : number  
1718313207514

(C) PET Lab/Fatec Jundiaí

20/06/2024

caninos@labrado... Node-RED : Ola! ...

42

The screenshot shows the Node-RED interface with a flow titled 'Ola!'. The flow consists of an 'inject' node, a 'Carimbo de data/hora' (Timestamp) node, and a 'debug' node. A 'Mensagem' (Message) node is connected to the output of the timestamp node. Two hand icons point to the connection between the timestamp node and the message node, indicating that multiple nodes can feed into one. A blue callout box labeled '2 Experimente a injeção de dados' (Experiment with data injection) is positioned over this connection. Another blue callout box labeled '1 Implemente as alterações' (Implement the changes) is at the top right, pointing to the 'implementar' button. A third blue callout box labeled '3 Observe o resultado da injeção' (Observe the result of the injection) points to the 'debug' node's log output on the right, which shows the timestamp and a generated ID. The left sidebar lists common nodes like inject, debug, and status, while the bottom sidebar lists function and switch nodes. The bottom navigation bar includes the date '20/06/2024' and the user 'caninos@labrado...'. The title bar shows the IP address '192.168.0.138 - Conexão de Área de Trabalho Remota' and the URL 'localhost:1880/#flow/78ea719049bf878e'.

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! +

localhost:1880/#flow/78ea719049bf878e

Node-RED

implementar

depurar

fluxo atual

filtrado

Ola!

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

Carimbo de data/hora

Mensagem

function 1

debug 1

1 Acrescente um nó **function** ao fluxo

2 Refaça as conexões entre os nós

3 Clique duplo para editar o nó

```
graph LR; inject[inject] --> Mensagem[Mensagem]; Mensagem --> Carimbo[Carimbo de data/hora]; Carimbo --> function1(function 1); function1 --> debug1[debug 1];
```

(C) PET Lab/Fatec Jundiaí

caninos@labrado... Node-RED : Ola! ...

43

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! +

localhost:1880/#flow/78ea719049bf878e

Node-RED

Ola!

implementar

filtrar nós

comum

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

função

- function
- switch

Carimbo de data

Mensagem

Editar function nó

Deletar

Cancelar

Feito

Propriedades

Nome: Formatar Data/Hora

Configurar No início Na mensagem Na parada

1

Indique um nome apropriado

2

Digite o código JavaScript necessário

3

Acione Feito para concluir edição

024, 18:13:23 nº: debug 1  
load: strn(19)  
024, 18:  
/load: number  
313207514

Código do nó **function**:

- No início**: código executado *uma vez, no início* do funcionamento do fluxo.
- Na mensagem**: código executado *a cada mensagem recebida* (sem retenção de valor).
- Na parada**: código executado *uma vez, na finalização* do funcionamento do fluxo.

(C) PET Lab/Fatec Jundiaí

20/06/2024

caninos@labrado... Node-RED : Ola! ...

44

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! + localhost:1880/#flow/78ea719049bf878e

Node-RED

1 Implemente as alterações

2 Experimente a injeção de dados

3 Observe o resultado da injeção **após** função

Ola!

comum

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

funcção

- function
- switch

Formatar Data/Hora

Carimbo de data/hora

debug 1

Mensagem

depurar

fluxo atual

filtrado

13/06/2024, 20:46:08 nó: debug 1  
msg.payload : string[19]  
"Um ola do Node-RED!"

13/06/2024, 20:46:13 nó: debug 1  
msg.payload : string[15]  
"13/6/2024 20:46"

(C) PET Lab/Fatec Jundiaí

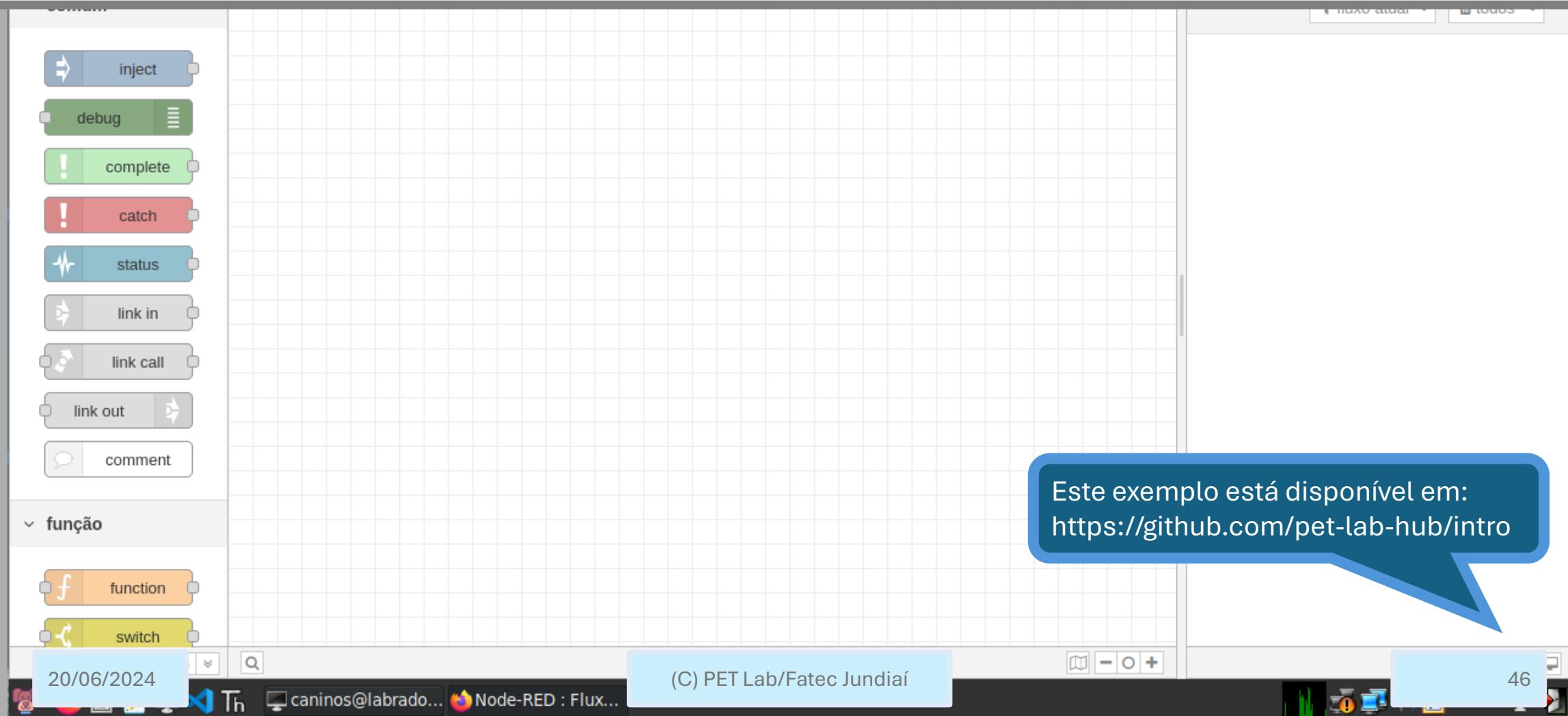
20/06/2024

caninos@labrado... Node-RED : Ola! ...

45

```
graph LR; inject[inject] --> timestamp[Carimbo de data/hora]; timestamp --> format[Formatar Data/Hora]; format --> debug[debug 1]; message[Mensagem] --> format;
```

# Exemplo 2 [MQTT]



192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Ola! +

localhost:1880/#flow/78ea719049bf878e

Node-RED

Ola!

filtrar nós

comum

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

função

- function
- switch

depurar

fluxo atual

filtrado

20:46:08 nó: debug 1  
: string[19]  
do Node-RED !

13/06/2024, 20:46:13 nó: debug 1  
msg.payload : string[15]  
"13/6/2024 20:46"

1 Clique no + para criar um novo fluxo

```
graph LR; inject[inject] --> Carimbo[Carimbo de data/hora]; Carimbo --> Mensagem[Mensagem]; Mensagem --> Formatar[Formatar Data/Hora]; Formatar --> debug1[debug 1];
```

(C) PET Lab/Fatec Jundiaí

20/06/2024

caninos@labrado... Node-RED : Ola! ...

47

192.168.0.138 - Conexão de Área de Trabalho Remota

### Node-RED : Fluxo 2

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

Fluxo 2

implementar

depurar

nós selecionados

filtrado

1 Adicione um nó **inject** e outro **function**

2 Acrescente um nó **mqtt out**

3 Faça as conexões indicadas

4 Clique duplo, altere o nome para **MQTT Count**

Triângulo vermelho indica erro na configuração ou código

Carimbo de data/hora → function 1 → mqtt

20/06/2024

(C) PET Lab/Fatec Jundiaí

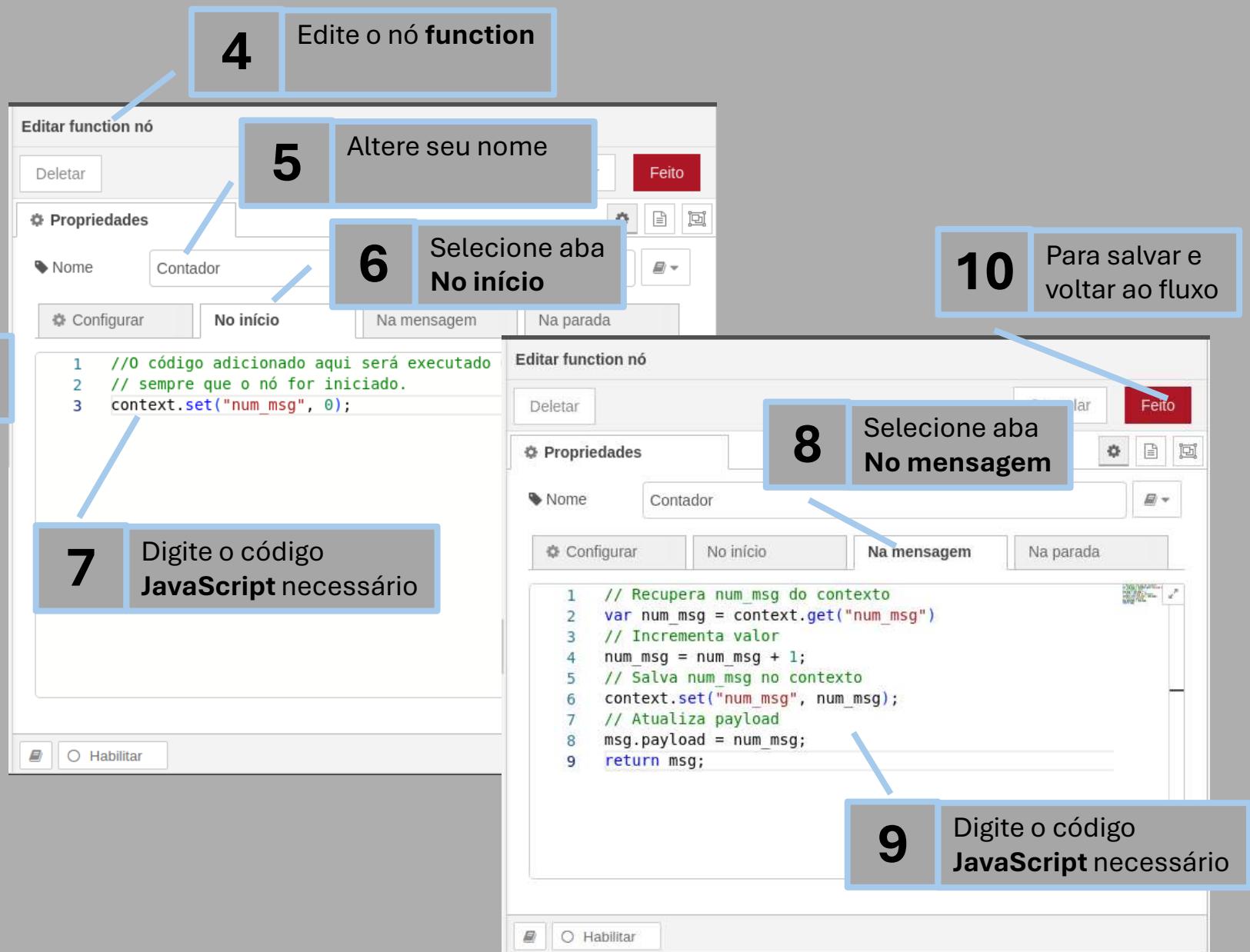
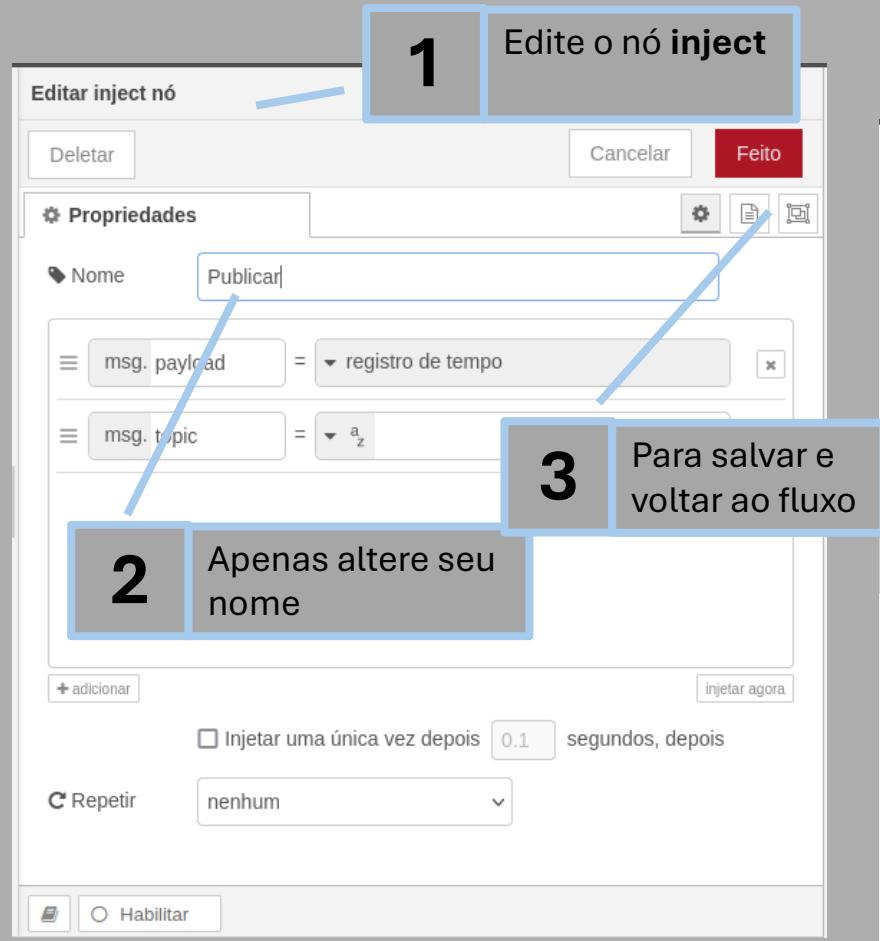
caninos@labrado... Node-RED : Flux...

48

```
graph LR; Carimbo[Carimbo de data/hora] --> Function1(function 1); Function1 --> MQTT[mqtt]
```

# Nós e contexto

- Como padrão, as mensagens (**msg.payload**) constituem o único meio para que um nó obtenha seus dados.
- Os dados assim obtidos não ficam retidos nos nós, ou seja, tanto as mensagens, como os resultados do seu processamento, são descartados após o encaminhamento da mensagem para o próximo nó e, portanto, não podem ser reutilizados.
- Para preservar dados entre processamentos diferentes, no mesmo nó, no mesmo fluxo ou na mesma instância do Node-RED, podemos utilizar o **contexto**.
- Existem três níveis de contexto:
  - Local (objeto **context**), acessível apenas pelo nó que o criou.
  - Fluxo (objeto **flow**), acessível por todos os nós de um fluxo.
  - Global (objeto **global**), acessível por todos os nós uma instância do Node-RED.



**1** Edite o nó mqtt out

**2** Verifique a seleção do Mosquitto (pré-instalação)

**3** Especifique o tópico MQTT desejado

**4** Use QoS = 2

**5** Para salvar e voltar ao fluxo

**Propriedades**

- Servidor: Mosquitto
- Tópico: contagem
- QoS: 2
- Nome: Nome

**Propriedades**

- Servidor: Mosquitto
- Tópico: Mosquitto

**Propriedades**

- Nome: Nome
- Conexão: por exemplo, localhost
- Protocolo: MQTT V3.1.1
- ID do cliente: Deixe em branco para geração automática
- Mantenha-se vivo: 60
- Sessão: Usar sessão limpa

**IP de host (p.e., Labrador) onde serviço MQTT (p.e., Mosquitto) está disponível.**

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 2

localhost:1880/#flow/ba926e435d2f64b5

Injetado com sucesso: Publicar

implementar

depurar

Quadrado verde indica conexão OK

1 Implemente as alterações

2 Experimente a injeção de dados

3 Observe o resultado da injeção **após** função

A cada injeção deve ser exibida uma mensagem no console, mas nada fica registrado na aba de debug, pois não temos nós debug no fluxo.

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 2

Injetado com sucesso: Publicar

implementar

depurar

fluxo atual

filtrado

13/06/2024, 22:00:01 nó: debug 2  
msg.payload : number

2

3

4

5

2

Implemente as alterações

1 Acrescente um nó debug como indicado

3 Experimente a injeção de dados

4 Observe a contagem das mensagens

20/06/2024

caninos@labrado... Node-RED : Flux...

(C) PET Lab/Fatec Jundiaí

53

192.168.0.138 - Conexão de Área de Trabalho Remota

### Node-RED : Fluxo 2

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

implementar

depurar

fluxo atual

filtrado

Ola! MQTT Count

inject

debug

complete

catch

status

link in

link call

link out

comment

funcção

rede

mqtt in

1 Adicione um nó mqtt in e outro debug

2 Clique duplo para editar o nó

3 Clique duplo para editar o nó

Publicar

Contador

contagem

conectado

debug 2

mqtt

debug 3

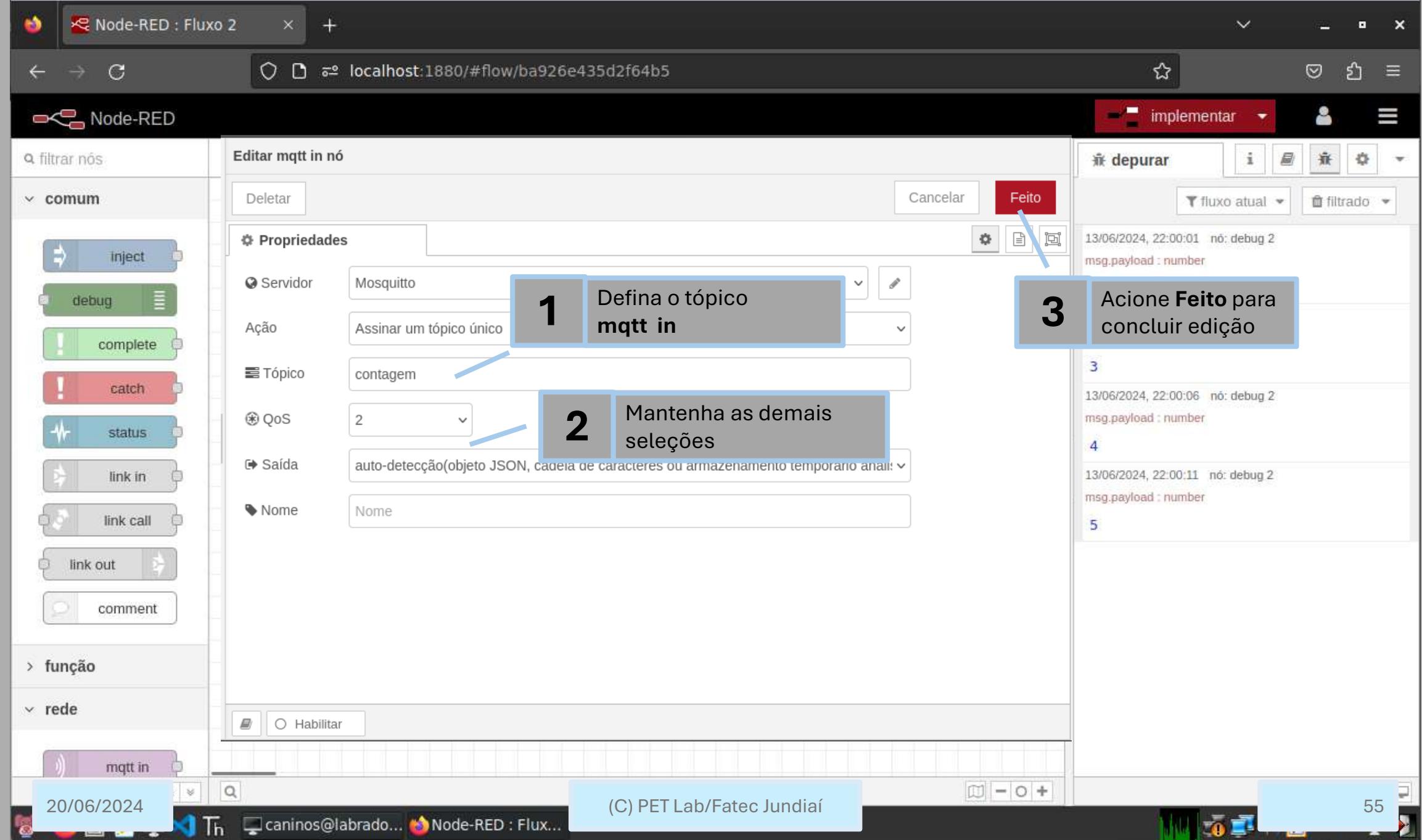
13/06/2024, 22:00:01 nó: debug 2  
msg.payload : number  
2  
13/06/2024, 22:00:03 nó: debug 2  
msg.payload : number  
3  
13/06/2024, 22:00:06 nó: debug 2  
msg.payload : number  
4  
13/06/2024, 22:00:11 nó: debug 2  
msg.payload : number  
5

20/06/2024

caninos@labrado... Node-RED : Flux...

(C) PET Lab/Fatec Jundiaí

54



Node-RED : MQTT Count

Injetado com sucesso: Publicar

implementar

depurar

fluxo atual

filtrado

17/06/2024, 11:20:55 nó: debug 2  
msg.payload : number  
1  
17/06/2024, 11:20:56 nó: debug 3  
contagem : msg.payload : number  
1

2 Experimente a injeção de dados

3 Observe o resultado da injeção **após** função

A injeção da contagem no tópico **contagem** no nó **MQTT out** é exibida pelo nó **debug 2**. Em seguida capturada pelo nó **MQTT in** e exibida pelo nó **debug 3**.

```
graph LR; inject[inject] --> Publicar[Publicar]; Publicar --> Contador[Contador]; Contador --> contagem1[contagem]; contagem1 --> debug2[debug 2]; contagem2[contagem] --> debug3[debug 3]
```

10.66.29.210 - Conexão de Área de Trabalho Remota

Node-RED : MQTT Count

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

implementar

depurar

fluxo atual

filtrado

17/06/2024, 11:20:56 - nó: debug 3  
contagem : msg.payload : number  
1

Olá! MQTT Count

inject

debug

complete

catch

status

link in

link call

link out

comment

funcion

switch

Publicar

Contador

contagem

debug 2

contagem

debug 3

interruptor

debug 4

debug 5

**1** Acrescente um nó switch

**2** Adicione outros dois nós debug

**3** Clique duplo para editar os nós adicionados

20/06/2024

(C) PET Lab/Fatec Jundiaí

57

**4**

Para salvar e  
voltar ao fluxo

**1**

Seleciona expressão  
e digite a indicada

**2**

Seleciona expressão  
e digite a indicada

**3**

Acione adicionar e  
repita o passo 2

**6**

Para salvar e  
voltar ao fluxo

Deletar  
Cancelar Feito

Propriedades

Nome

Saída J: payload

Para

janela de depuração

console do sistema

estado do nó (32 caracteres)

Nome

Par

**5**

Altere no nome

Deletar  
Cancelar Feito

Propriedades

Nome

Saída J: msg.payload

Para

janela de depuração

console do sistema

estado do nó (32 caracteres)

Nome

Impar

**7**

Altere no nome

Node-RED : MQTT Count

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

implementar

depurar

fluxo atual

filtrado

3 Implemente as alterações

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

10010

10011

10012

10013

10014

10015

10016

10017

10018

10019

10020

10021

10022

10023

10024

10025

10026

10027

10028

10029

10030

10031

10032

10033

10034

10035

10036

10037

10038

10039

10040

10041

10042

10043

10044

10045

10046

10047

10048

10049

10050

10051

10052

10053

10054

10055

10056

10057

10058

10059

10060

10061

10062

10063

10064

10065

10066

10067

10068

10069

10070

10071

10072

10073

10074

10075

10076

10077

10078

10079

10080

10081

10082

10083

10084

10085

10086

10087

10088

10089

10090

10091

10092

10093

10094

10095

10096

10097

10098

10099

100100

100101

100102

100103

100104

100105

100106

100107

100108

100109

100110

100111

100112

100113

100114

100115

100116

100117

100118

100119

100120

100121

100122

100123

100124

100125

100126

100127

100128

100129

100130

100131

100132

100133

100134

100135

100136

100137

100138

100139

100140

100141

100142

100143

100144

100145

100146

100147

100148

100149

100150

100151

100152

100153

100154

100155

100156

100157

100158

100159

100160

100161

100162

100163

100164

100165

100166

100167

100168

100169

100170

100171

100172

100173

100174

100175

100176

100177

100178

100179

100180

100181

100182

100183

100184

100185

100186

100187

100188

100189

100190

100191

100192

100193

100194

100195

100196

100197

100198

100199

100200

100201

100202

100203

100204

100205

100206

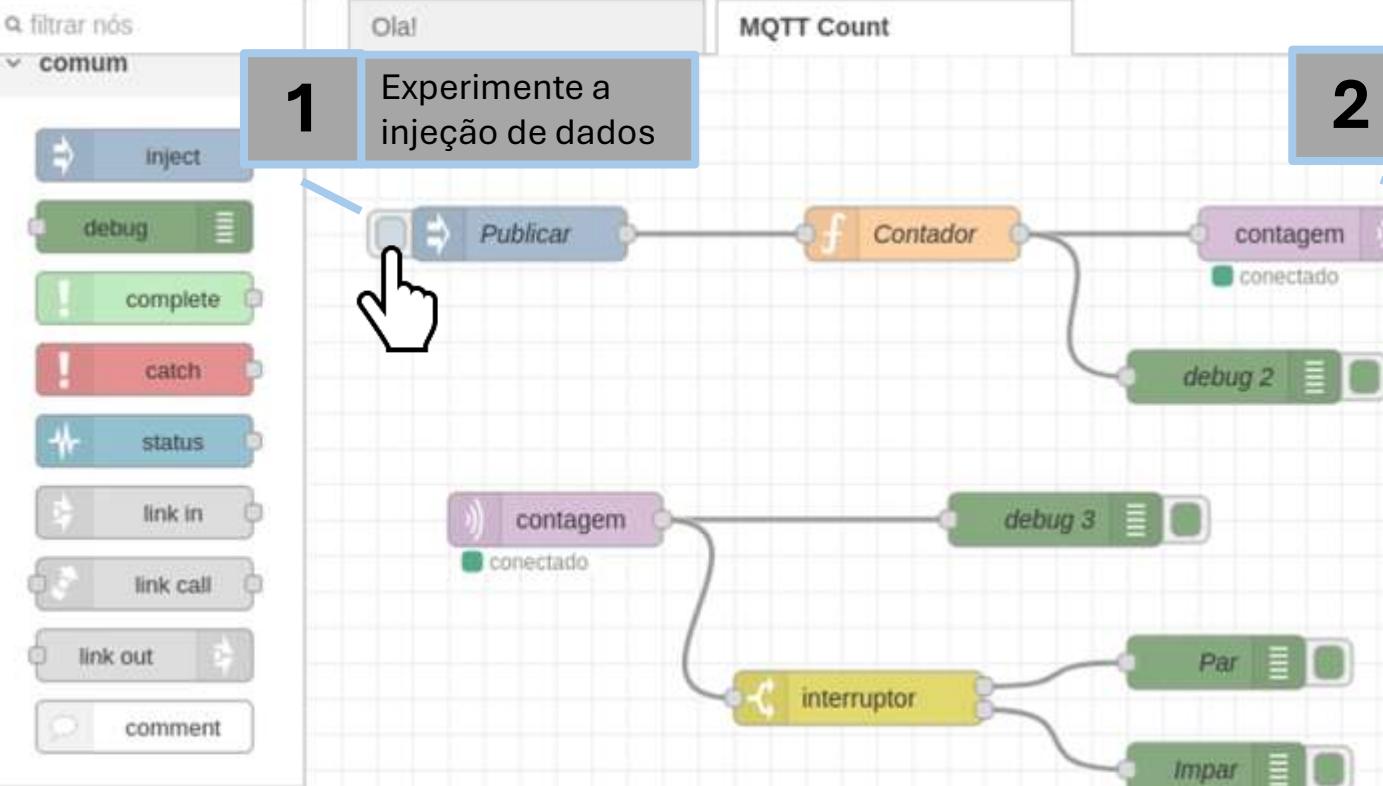
100

**Q. filtrar nós**

## 1 Experimente a injeção de dados



## 2 Observe as mensagens da aba de debug



A mensagem publicada no tópico **contagem** desviada para caminhos distintos com uso do nó **switch**.

11:38:13 nō: Par

17/06/2024, 11:38:20 nô: debug 2  
msg.payload : number

3  
17/06/2024, 11:38:20 nō: debug 3  
contagem : msg.payload : number

3  
17/06/2024, 11:38:20 nº: Imper  
contagem msg payload: number

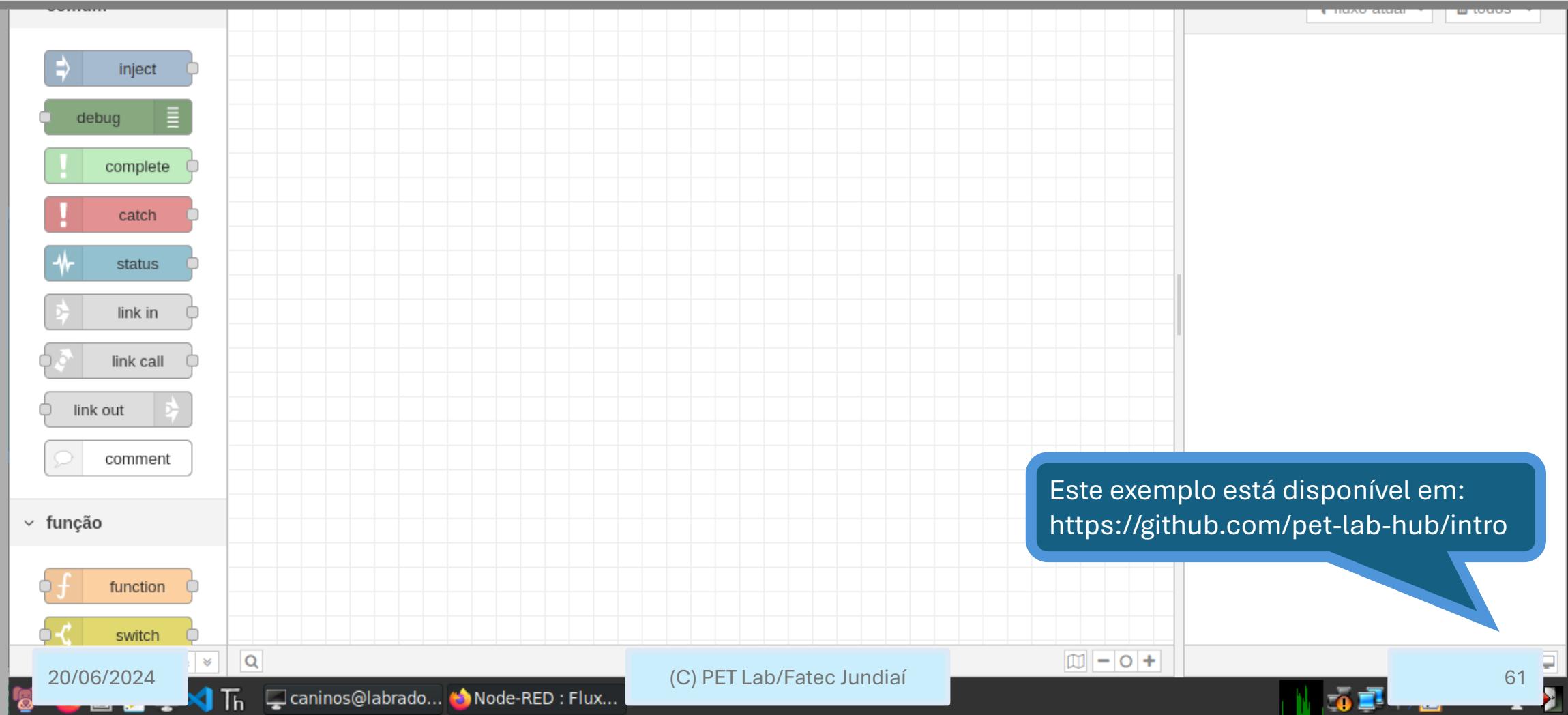
3  
17/06/2024, 12:05:47 no: debug 2  
msg.payload: number

4  
17/06/2024, 12:05:47 nó: debug 3  
contagem : msg.payload : number

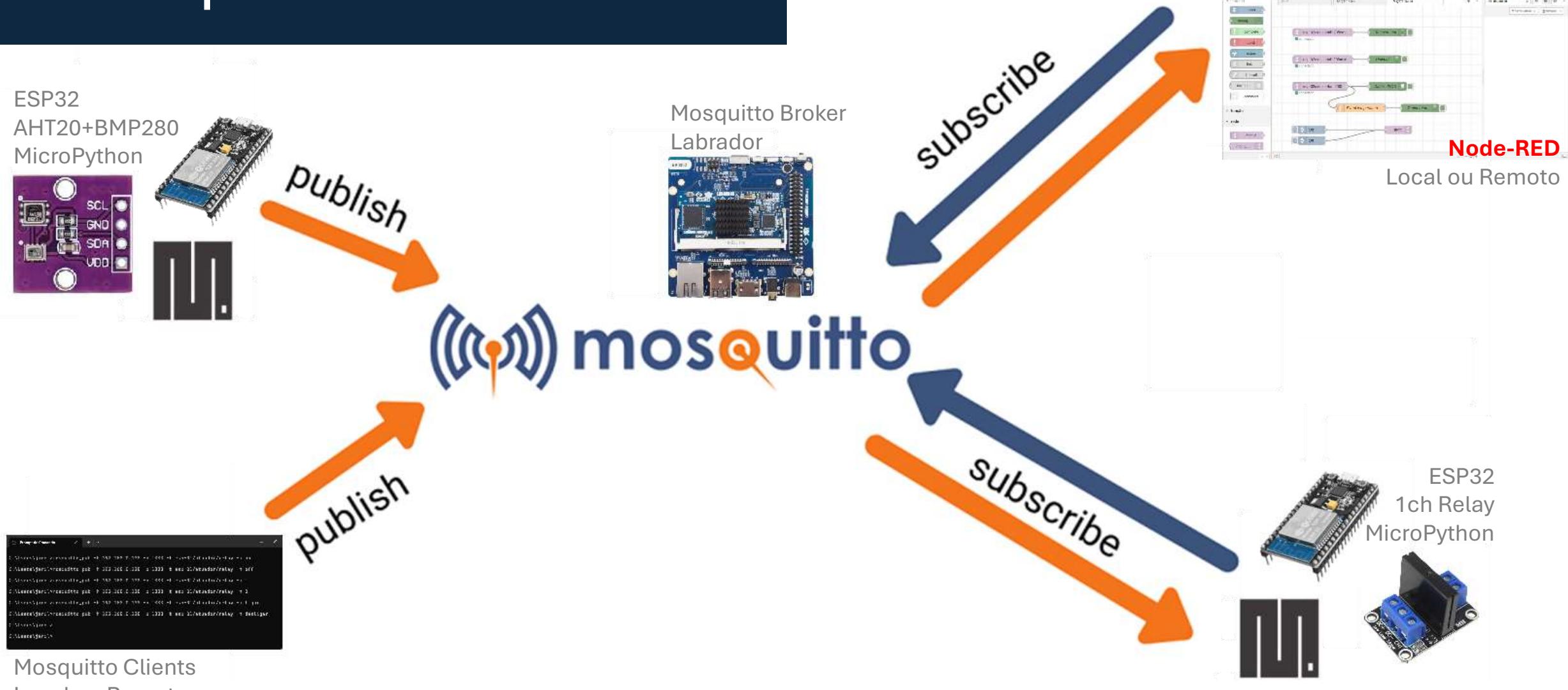
17/06/2024, 12:05:47 nº: Par  
contagem : msg : number



# Exemplo 3 [MQTT & IoT]



# Exemplo 3::Cenário



10.66.29.210 - Conexão de Área de Trabalho Remota

### Node-RED : MQTT Count

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

Ola! MQTT Count

implementar depurar

fluxo atual filtrado

filtrar nós

comum

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

função

- function
- switch

1 Acione o + para adicionar um novo fluxo

```
graph LR; inject[inject] --> Publicar[Publicar]; Publicar --> Contador[Contador]; Contador --> contagem1[contagem conectado]; contagem1 -- feedback --> Contador; contagem1 --> debug2[debug 2]; contagem2[contagem conectado] --> debug3[debug 3]; contagem2 --> interruptor[interruptor]; interruptor --> Par[Par]; interruptor --> Impar[Impar];
```

20/06/2024

(C) PET Lab/Fatec Jundiaí

caninos@labrado... Node-RED : MQT...

63

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

filtrar nós

comum

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

função

- function
- switch

editar Fluxo: MQTT Data

1 Altere o nome como desejado

2 Opcionalmente forneça uma descrição

3 Para salvar e voltar ao fluxo

deletar

Cancelar

Feito

depurar

fluxo atual

filtrado

implementar

filtrar

modo

Unlocked

(C) PET Lab/Fatec Jundiaí

20/06/2024

caninos@labrado... Node-RED : Fluxo 1

64

10.66.29.210 - Conexão de Área de Trabalho Remota

### Node-RED : Fluxo 1

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

implementar

depurar

fluxo atual

filtrado

Ola!

MQTT Count

MQTT Data

inserir

debug

complete

catch

status

link in

link call

link out

comment

funcção

rede

mqtt in

3 Dê um clique duplo para editar cada nó

2 Conecte os nós como indicado

1 Adicione dois nós **mqtt in** e dois nós **debug**

```
graph LR; mqtt_in1[mqtt in] --> mqtt1[mqtt]; mqtt1 --> debug4[debug 4]; mqtt_in2[mqtt in] --> mqtt2[mqtt]; mqtt2 --> debug5[debug 5];
```

## Editar mqtt in nó

1 Selezione opção **Adicionar novo mqtt broker...**

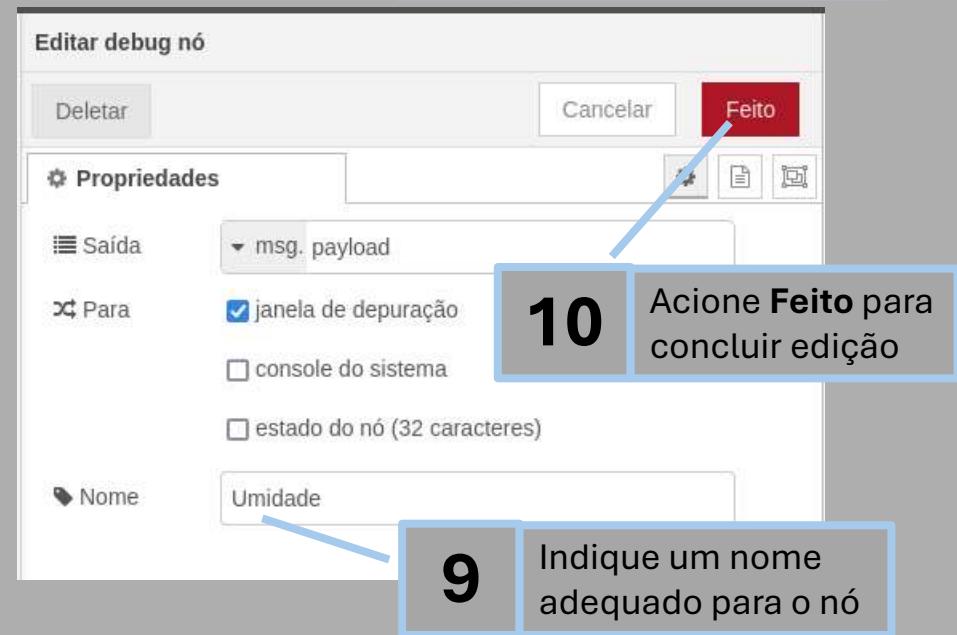
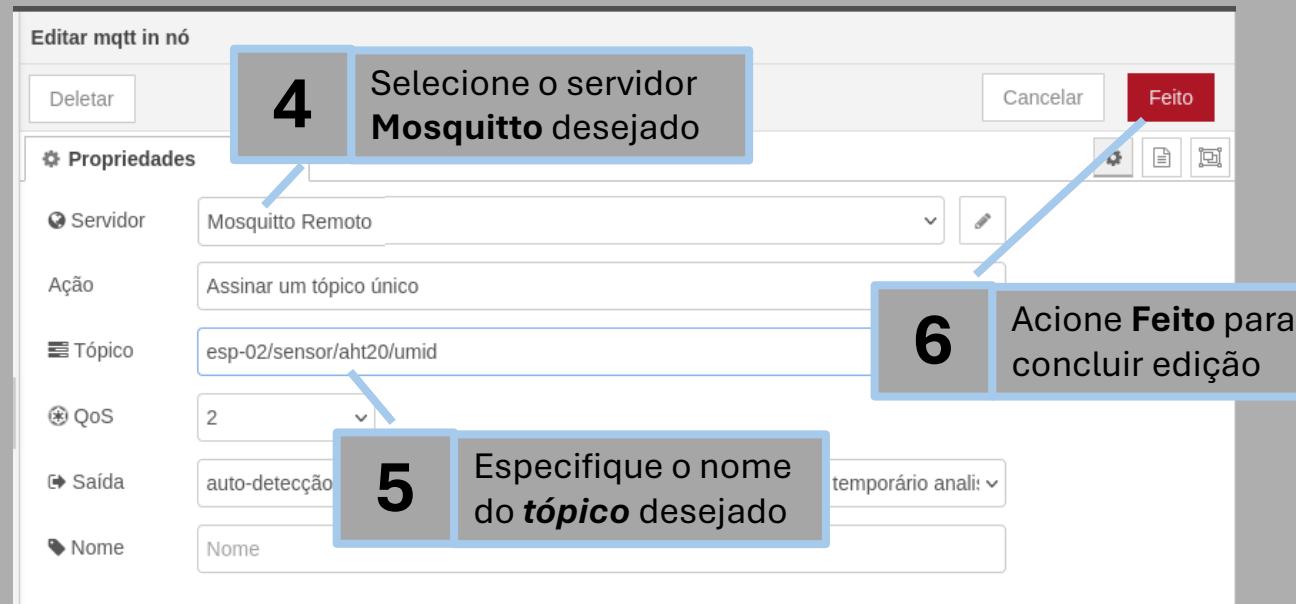
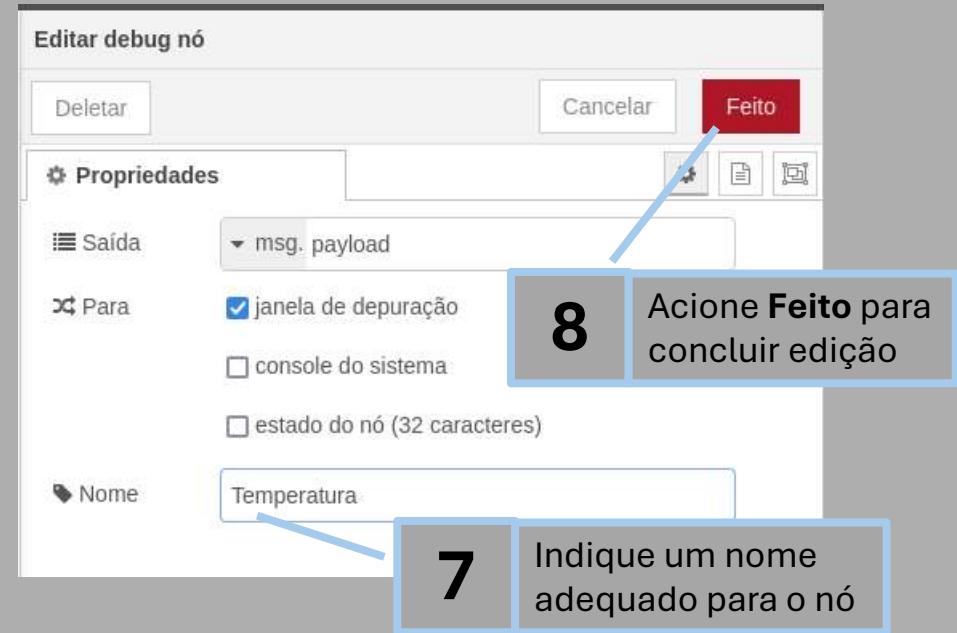
2 Acione o botão de edição

3 Forneça o IP do broker

4 Acione **Adicionar**

5 Acione **Feito**

The screenshot shows the 'Editar mqtt in nó' (Edit MQTT node) configuration screen. On the left, there are fields for 'Propriedades' (Properties), 'Servidor' (Broker), 'Ação' (Action), 'Tópico' (Topic), 'QoS', 'Saída' (Output), and 'Nome' (Name). The 'Servidor' dropdown is set to 'Mosquitto'. The 'Ação' dropdown has 'Adicionar novo mqtt-broker...' selected. The 'Tópico' field contains 'esp-02/sensor/aht20/temp'. The 'Nome' field is empty. At the bottom are buttons for 'Habilitar' (Enable) and 'Cancelar' (Cancel). A red 'Feito' (Done) button is highlighted. A blue box labeled '1' points to the 'Adicionar novo mqtt broker...' option. A blue box labeled '2' points to the edit icon next to the 'Ação' dropdown. A blue box labeled '3' points to the 'Servidor' input field where '10.66.0.15' is entered. A blue box labeled '4' points to the 'Adicionar' (Add) button in the sub-modal window. A blue box labeled '5' points to the 'Feito' button in the sub-modal window.



10.66.29.210 - Conexão de Área de Trabalho Remota

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

Node-RED : Fluxo 1 +

localhost:1880/#flow/884d430cb0401c89

Node-RED

filtrar nós

comum

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

Ola! MQTT Count MQTT Data

depurar fluxo atual filtrado

18/06/2024, 17:17:40 nó: Temperatura  
esp-02/sensor/aht20/temp : msg.payload : number  
**24.29008** ←

18/06/2024, 17:17:40 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
**41.69788** ←

1 Implemente as modificações para ativar o fluxo

2 Observe a configuração e estado nos nós

Aqui, utilizamos um nó **mqtt in** para cada **dado simples** publicado em tópico individual.

3 Aguarde e observe a **recepção de dados do sensor**

20/06/2024 caninos@labrado... Node-RED : Flux... (C) PET Lab/Fatec Jundiaí

68

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

implementar

depurar

fluxo atual

filtrado

18/06/2024, 17:24:53 nó: Umidade  
esp-02/sensor/ah20/umid : msg.payload : number  
**42.71154**

18/06/2024, 17:25:55 nó: Temperatura  
esp-02/sensor/ah20/temp : msg.payload : number  
**24.27731**

18/06/2024, 17:25:55 nó: Umidade  
esp-02/sensor/ah20/umid : msg.payload : number  
**42.50174**

18/06/2024, 17:26:05 nó: Temperatura  
esp-02/sensor/ah20/temp : msg.payload : number  
**24.25461**

18/06/2024, 17:26:06 nó: Umidade  
esp-02/sensor/ah20/umid : msg.payload : number  
**42.29422**

18/06/2024, 17:27:07 nó: Temperatura  
esp-02/sensor/ah20/temp : msg.payload : number  
**24.24641**

18/06/2024, 17:27:07 nó: Umidade  
esp-02/sensor/ah20/umid : msg.payload : number  
**42.08221**

inserir

filtrar nós

comum

funcção

rede

Olá!

MQTT Count

MQTT Data

+

Temperatura

Umidade

esp-02/sensor/ah20/temp

esp-02/sensor/ah20/umid

mqtt

debug

complete

catch

status

link in

link call

link out

comment

1 Adicione outro nó **mqtt in** e outro nó **debug**

2 Conecte os nós como indicado

3 Dê um clique duplo para editar cada nó

```
graph LR; mqtt_in1[ mqtt in ] --> debug1[ debug ]; debug1 --> fun1[ Temperatura ]; debug1 --> fun2[ Umidade ]; mqtt_in2[ mqtt in ] --- mqtt_in1;
```

**Editar mqtt in nó**

**1** Selecione o servidor **Mosquitto** desejado

**2** Especifique o nome do **tópico** desejado

**3** Acione **Feito** para concluir edição

**4** Indique um nome adequado para o nó

**5** Acione **Feito** para concluir edição

**Editar debug nó**

**1** Selecione o servidor **Mosquitto** desejado

**2** Especifique o nome do **tópico** desejado

**3** Acione **Feito** para concluir edição

**4** Indique um nome adequado para o nó

**5** Acione **Feito** para concluir edição

# JSON

## JavaScript Object Notation

- Formato comum de APIs da web usado para representar objetos do JavaScript como texto.
- Elementos:
  - { } delimitador de objetos
  - :
 separador do par k:v  
(chave:valor)
- ,
 separador de pares k:v  
ou elementos de array- [] delimitador de array

Semelhante aos  
dicionários Python

- As chaves são geralmente texto (string).
- Valores podem ser strings, números, booleanos, nulo, objetos JSON ou arrays.
- Exemplo:

```
{  
    'temperatura': 25.3 ,  
    'pressao': 1234.567 ,  
    'umidade': 63.8  
}
```

10.66.29.210 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

filtrar nós

comum

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

funcção

rede

- mqtt in

Ola!

MQTT Count

MQTT Data

depurar

fluxo atual

filtrado

24.10622

18/06/2024, 17:36:22 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
42.44251

18/06/2024, 17:37:23 nó: Temperatura  
esp-02/sensor/aht20/temp : msg.payload : number  
24.09229

18/06/2024, 17:37:24 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
24.09153

18/06/2024, 17:38:25 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
42.63973

18/06/2024, 17:38:56 nó: Dados JSON  
esp-02/sensor/bmp280 : msg.payload : Object  
object  
pressao: "936.1512"  
temperatura: "24.75"

1

Implemente as modificações para ativar o fluxo

2

Observe a configuração e estado nos nós

Aqui, utilizamos um nó **mqtt in** para **dados múltiplos** recebidos como um objeto **JSON**.

3

Aguarde e observe a **recepção de dados do sensor**

```
graph LR; esp1[esp-02/sensor/aht20/temp] --> T1[Temperatura]; esp2[esp-02/sensor/aht20/umid] --> U1[Umidade]; esp3[esp-02/sensor/bmp280] --> D1[Dados JSON]
```

10.66.29.210 - Conexão de Área de Trabalho Remota

### Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

implementar

depurar

fluxo atual

filtrado

Ola! MQTT Count MQTT Data

comum

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

funcção

- function
- switch

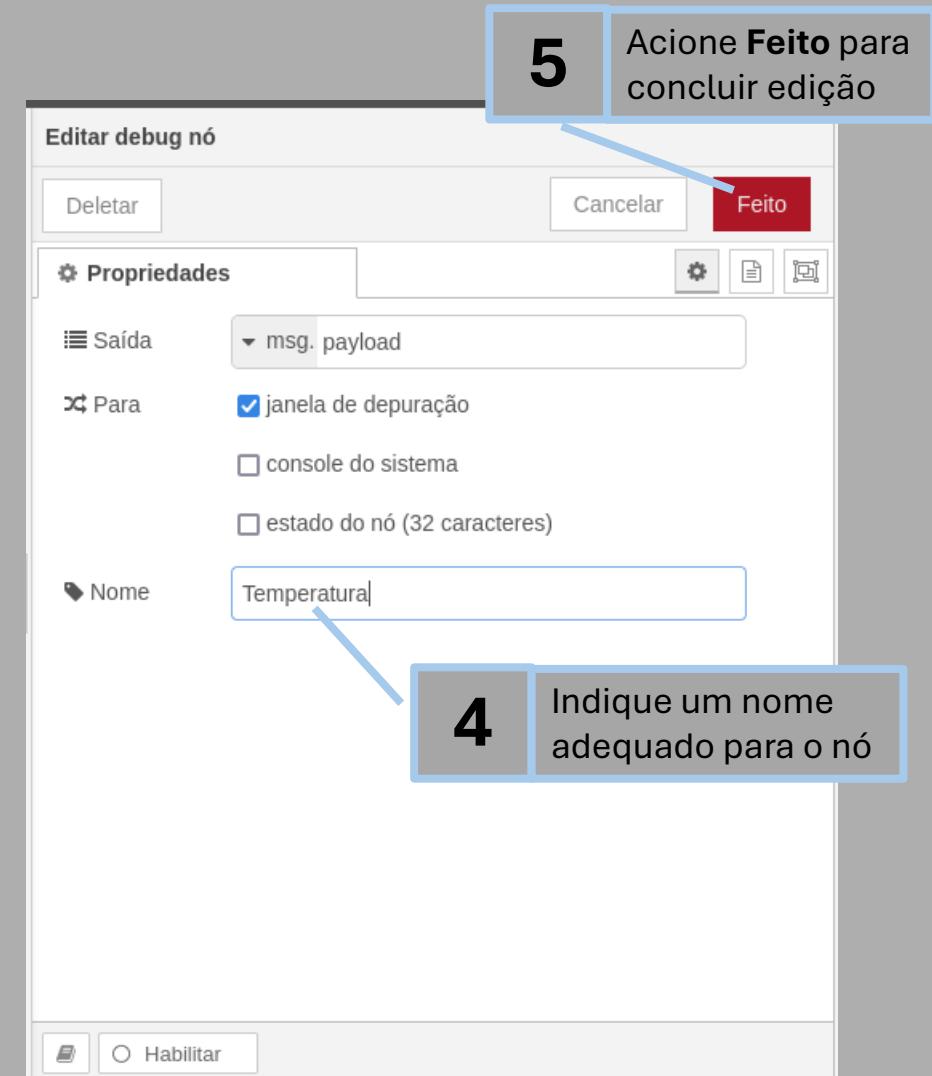
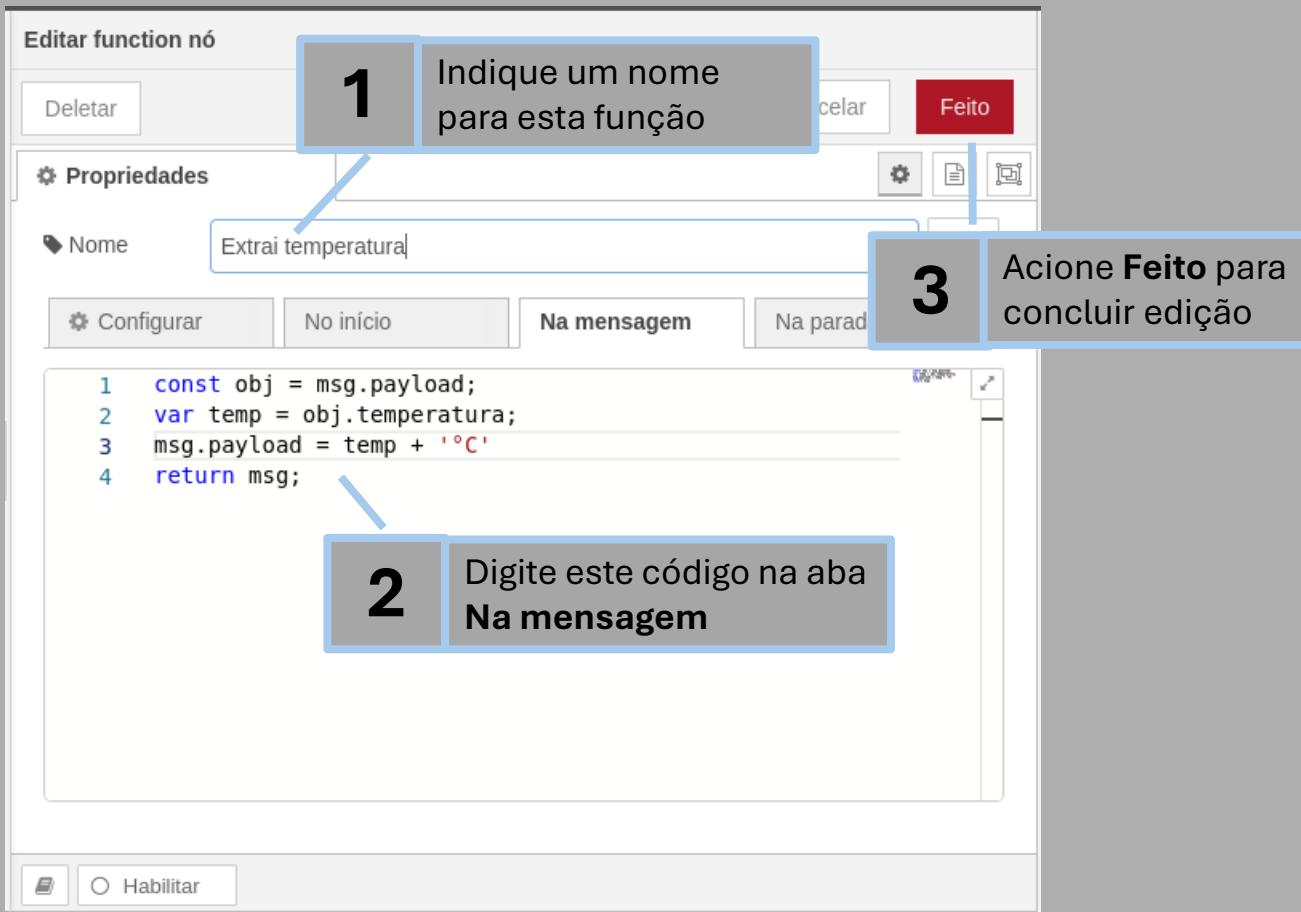
2 Coplete os nós como indicado

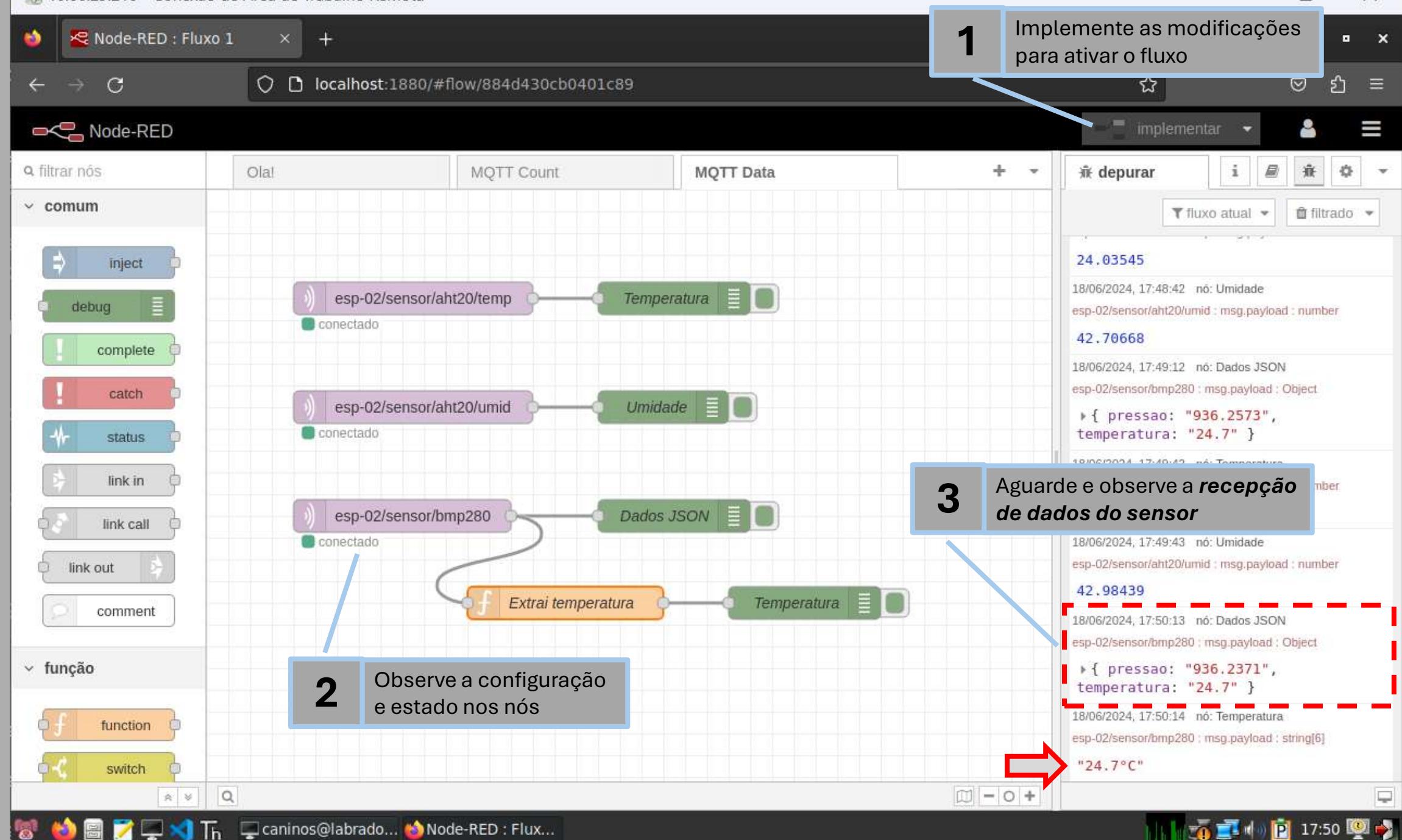
3 Adicione um nó **function** e outro nó **debug**

3 Dê um clique duplo para editar cada nó

```
graph TD; inject[inject] --> debug1[debug]; debug1 --> complete1[complete]; complete1 --> catch1[catch]; catch1 --> status1[status]; status1 --> linkIn1[link in]; linkIn1 --> linkCall1[link call]; linkCall1 --> linkOut1[link out]; linkOut1 --> comment1[comment]; linkIn1 --> function1(function); linkCall1 --> function1; linkOut1 --> debug4[debug 4]; function1 --> debug4; debug4 --> MQTTData[MQTT Data]; espTemp[esp-02/sensor/aht20/temp] --> Temperatura[Temperatura]; espUmid[esp-02/sensor/aht20/umid] --> Umidade[Umidade]; espBmp[bmp280] --> DadosJSON[Dados JSON]; espBmp --> function1;
```

24.10622  
18/06/2024, 17:36:22 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
42.44251  
18/06/2024, 17:37:23 nó: Temperatura  
esp-02/sensor/aht20/temp : msg.payload : number  
24.09229  
18/06/2024, 17:37:24 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
42.88177  
18/06/2024, 17:38:25 nó: Temperatura  
esp-02/sensor/aht20/temp : msg.payload : number  
24.09153  
18/06/2024, 17:38:25 nó: Umidade  
esp-02/sensor/aht20/umid : msg.payload : number  
42.63973  
18/06/2024, 17:38:56 nó: Dados JSON  
esp-02/sensor/bmp280 : msg.payload : Object  
object  
pressao: "936.1512"  
temperatura: "24.75"





192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : MQTT Data

localhost:1880/#flow/884d430cb0401c89

Node-RED

implementar

depurar

fluxo atual

filtrado

filtrar nós

Ola!

MQTT Count

MQTT Data

inject

debug

complete

catch

status

link in

link call

link out

comment

função

rede

mqtt.in

mqtt out

esp-02/sensor/aht20/temp

Temperatura

esp-02/sensor/aht20/umid

Umidade

esp-02/sensor/bmp280

Dados JSON

Carimbo de data/hora

Carimbo de data/hora

Temperatura

2 Cione os nós como indicado

1 Adicione um nó mqtt out e dois nós inject

3 Dê um clique duplo para editar cada nó

20/06/2024

(C) PET Lab/Fatec Jundiaí

caninos@labrado... Node-RED : MQT...

76

```
graph TD; mqttIn((mqtt.in)) --> inject1[inject]; mqttIn --> inject2[inject]; inject1 --> ahtTemp[esp-02/sensor/aht20/temp]; inject2 --> ahtUmid[esp-02/sensor/aht20/umid]; ahtTemp --> temperatura1[Temperatura]; ahtUmid --> umidade[Umidade]; ahtUmid --> json[Dados JSON]; json --> temperatura2[Temperatura]; ahtTemp --> timestamp1[Carimbo de data/hora]; ahtUmid --> timestamp2[Carimbo de data/hora]; timestamp1 --> mqttOut((mqtt.out)); timestamp2 --> mqttOut;
```

**1** Indique o nome **On** (ou **Ligar**) neste **inject**

**2** Selecione cadeia de caracteres e digite **ligar**

**3** Indique o nome do tópico MQTT

O Node-RED adicione esta propriedade automaticamente quando um **inject** está conectado a um **mqtt out**.

**4** Acione **Feito** para concluir edição

**5** Indique o nome **Off** (ou **Desligar**) neste **inject**

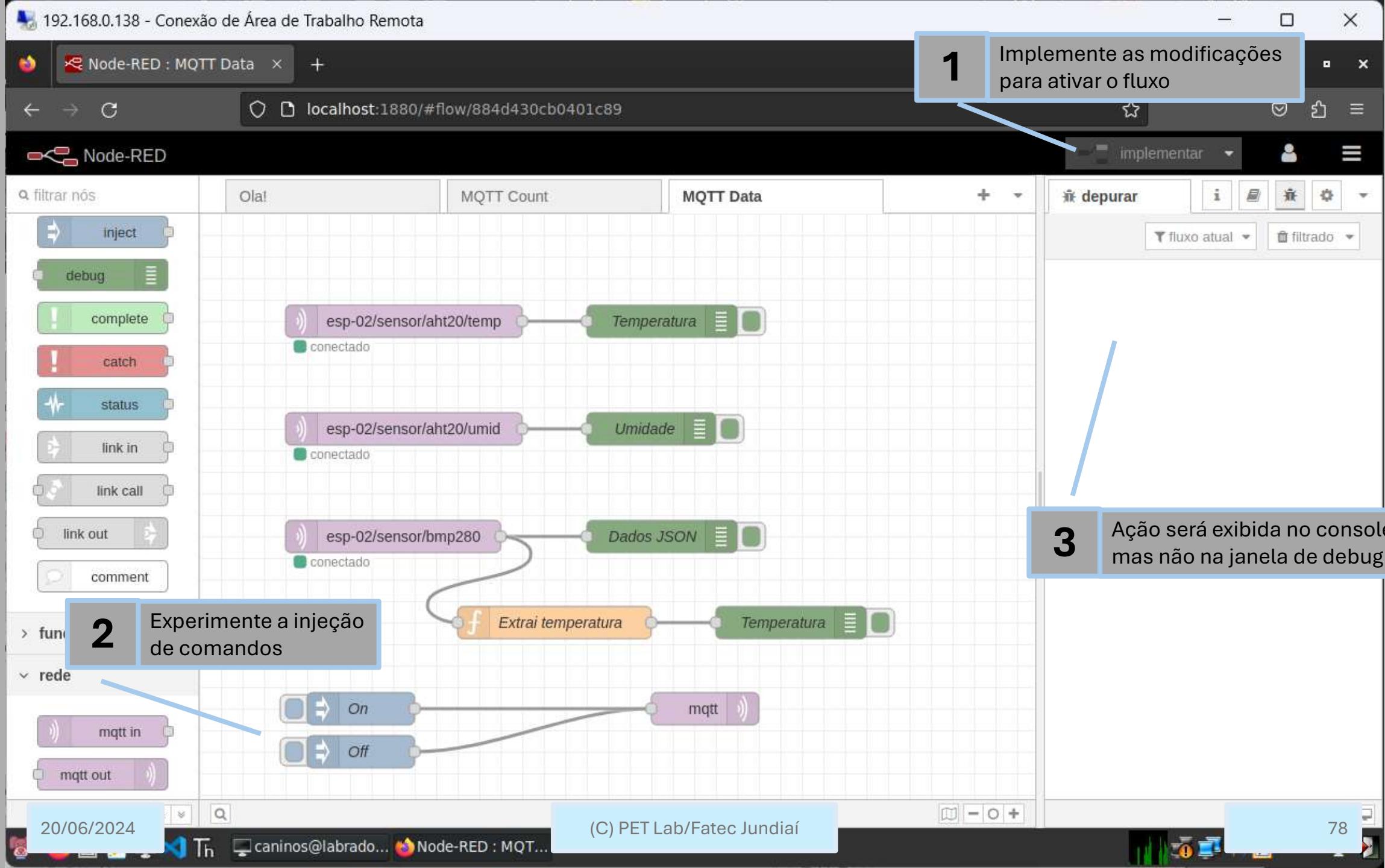
**6** Selecione cadeia de caracteres e digite **desligar**

**7** Indique o mesmo nome do tópico MQTT

**8** Acione **Feito** para concluir edição

**9** Deixe as propriedades em branco

**10** Acione **Feito** para concluir edição



# Próximos Passos

# Próximos Passos

## Node-RED

- Aprofundamento e outras aplicações da ferramenta, por exemplo:
  - Integração com banco de dados, páginas HTML etc.
  - Criação de dashboards com componentes do Node-RED.

## Labrador

- Aprofundamento e outras aplicações, por exemplo:
  - Programação C/C++ com uso de GPIO.
  - Integração com **FastIoT** (plataforma de prototipagem rápida da Fatec Jundiaí).

# obrigado

- PET Lab  
**Projetos Educacionais e Tecnologia**  
<https://github.com/pet-lab-hub>
- Peter Jndl Jr  
[peter.jndl@fatec.sp.gov.br](mailto:peter.jndl@fatec.sp.gov.br)
- Cláudio Luis V. Oliveira  
[claudio.oliveira@fatec.sp.gov.br](mailto:claudio.oliveira@fatec.sp.gov.br)
- Humberto A. P. Zanetti  
[humberto.zanetti@fatec.sp.gov.br](mailto:humberto.zanetti@fatec.sp.gov.br)