

PET Lab

Fatec
Jundiaí
Deputado Ary Fossen

Projetos Educacionais e de Tecnologia



Introdução aos Single Board Computers (SBC) e ao desenvolvimento de aplicações IoT usando Labrador e plataforma Low Code (Node-RED)

Prof. Ms. Peter Jandl Jr

Prof. Ms. Cláudio Luís V. Oliveira

Prof. Dr. Humberto A. P. Zanetti

Caninos Loucos Labrador

Caninos Loucos Labrador

27/06/2024

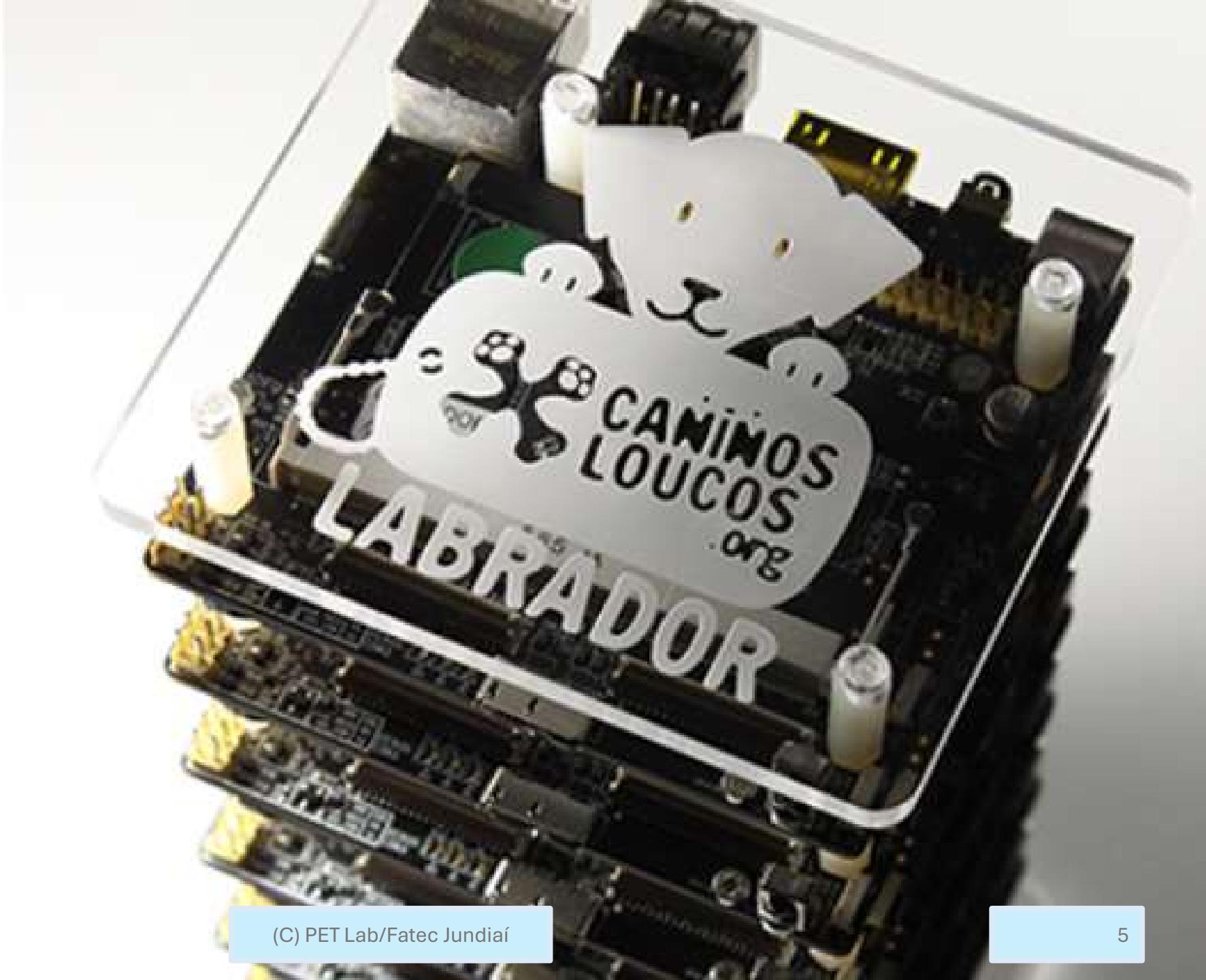
IoT e MQTT

Node RED

(C) PET Lab/Fatec Jundiaí

Mãos na Massa

Próximos Passos



Caninos Loucos Labrador

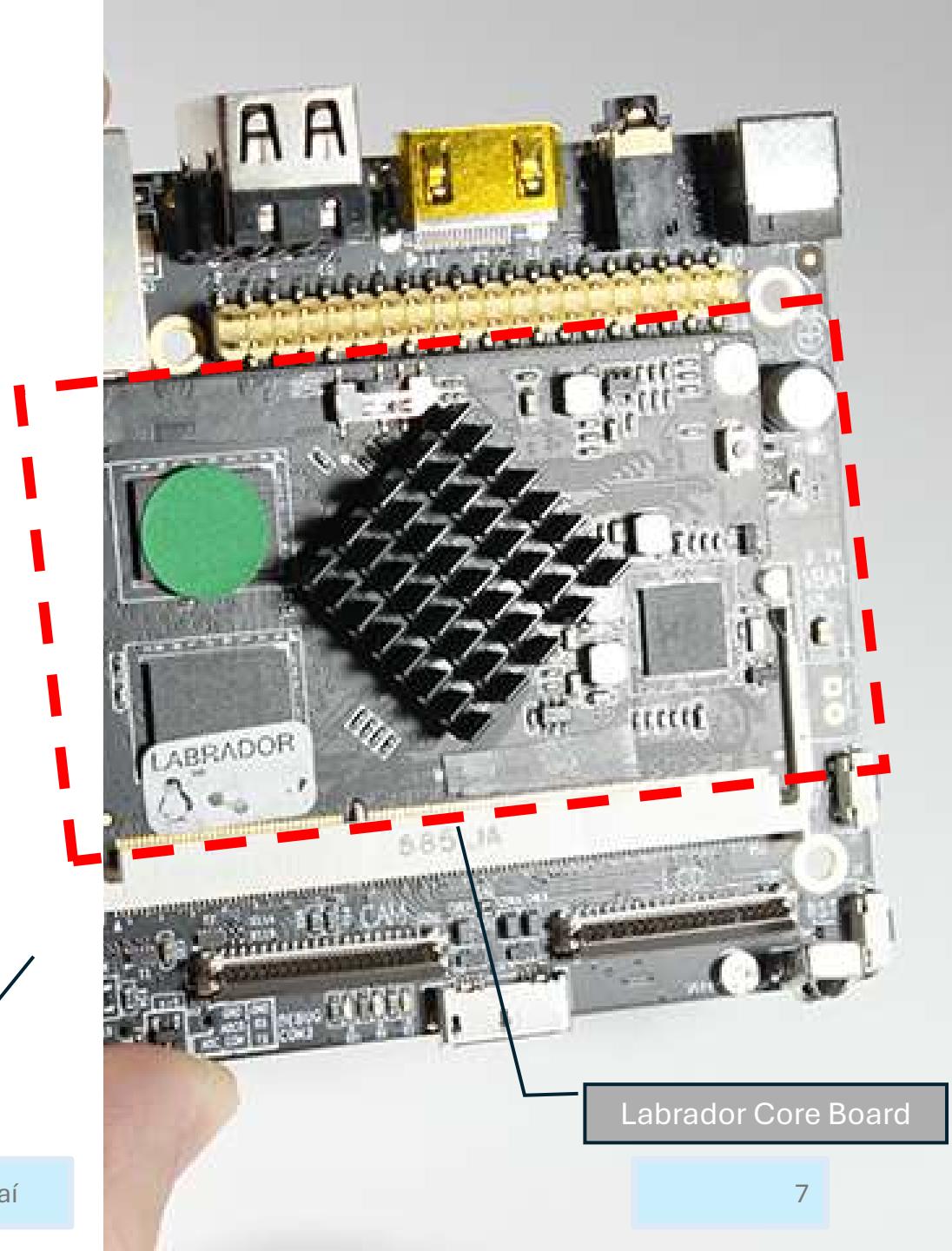
- Single Board Computer brasileira, com projeto de hardware e software abertos, que pode ser utilizada como um microcomputador ou como plataforma para aplicações de Internet das Coisas, sistemas ciber-físicos, robótica ou monitoramento.



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



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]

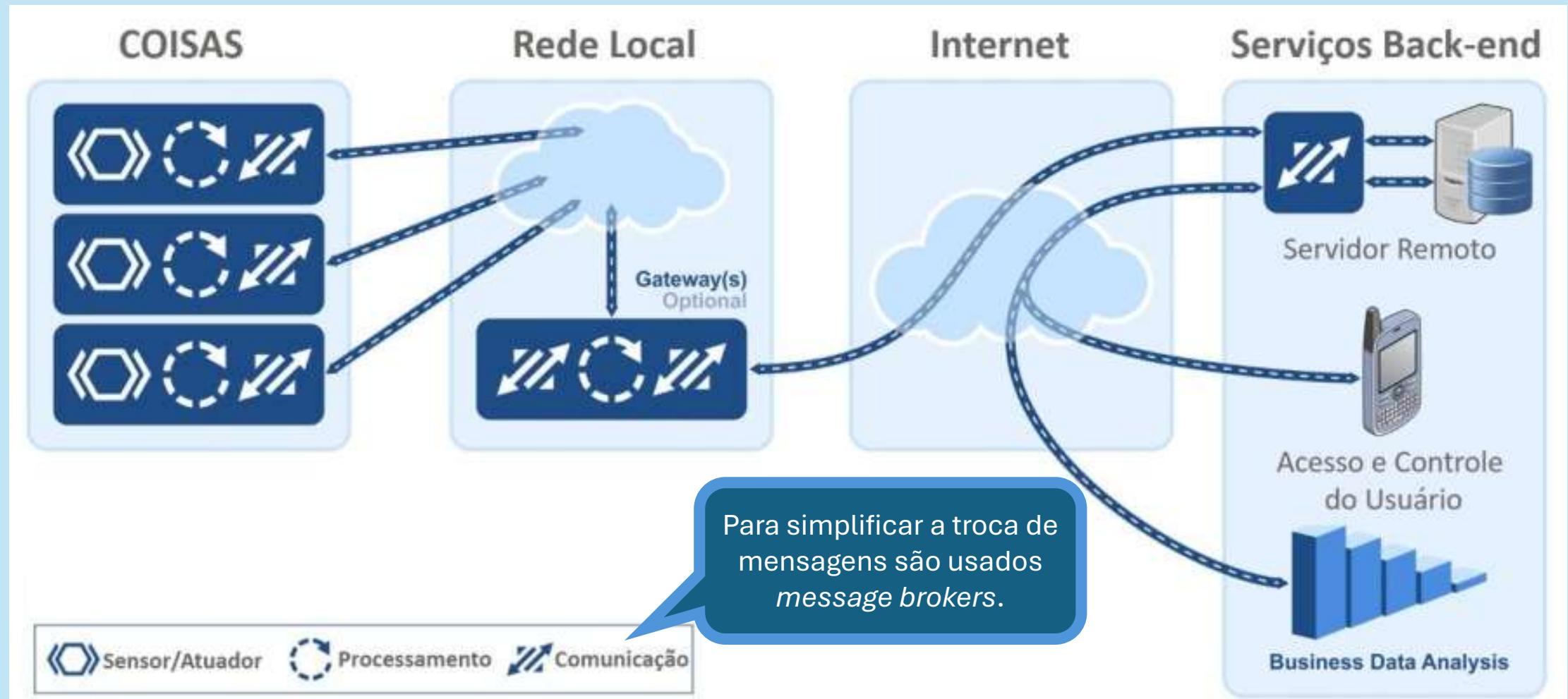
- Armazenamento: MicroSD Card Slot SD/SDHC/SDXC (até 32GB)
- Ethernet 10/100Mbps (RJ45)
- Wireless Wi-Fi 802.11 b/g/n 2.4GHz, Bluetooth 5.0 LE e receptor infravermelho (38kHz)
- USB: (2x) 2.0 HOST (tipo A), USB3.0 OTG (micro-B)
- Display:
 - HDMI 1.4 (tipo A), até 1920x1080@60Hz
 - LVDS-DSI para LCD 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@2A
- 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) 1,3GHz	Quad-core A72 ARM Cortex (Arm v.8) 1,8GHz
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 32GB)
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

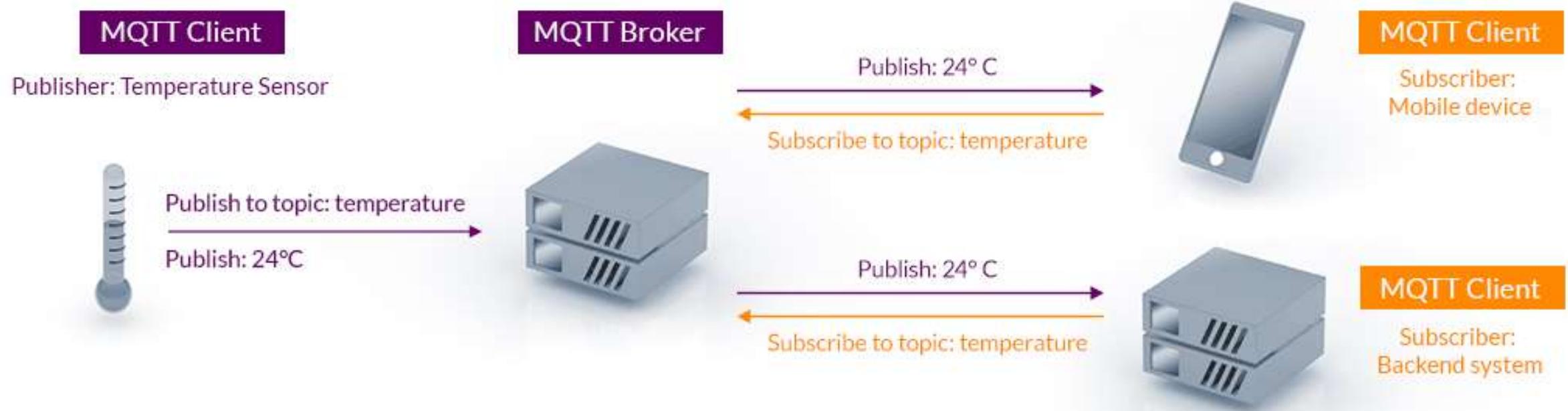


message broker

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.

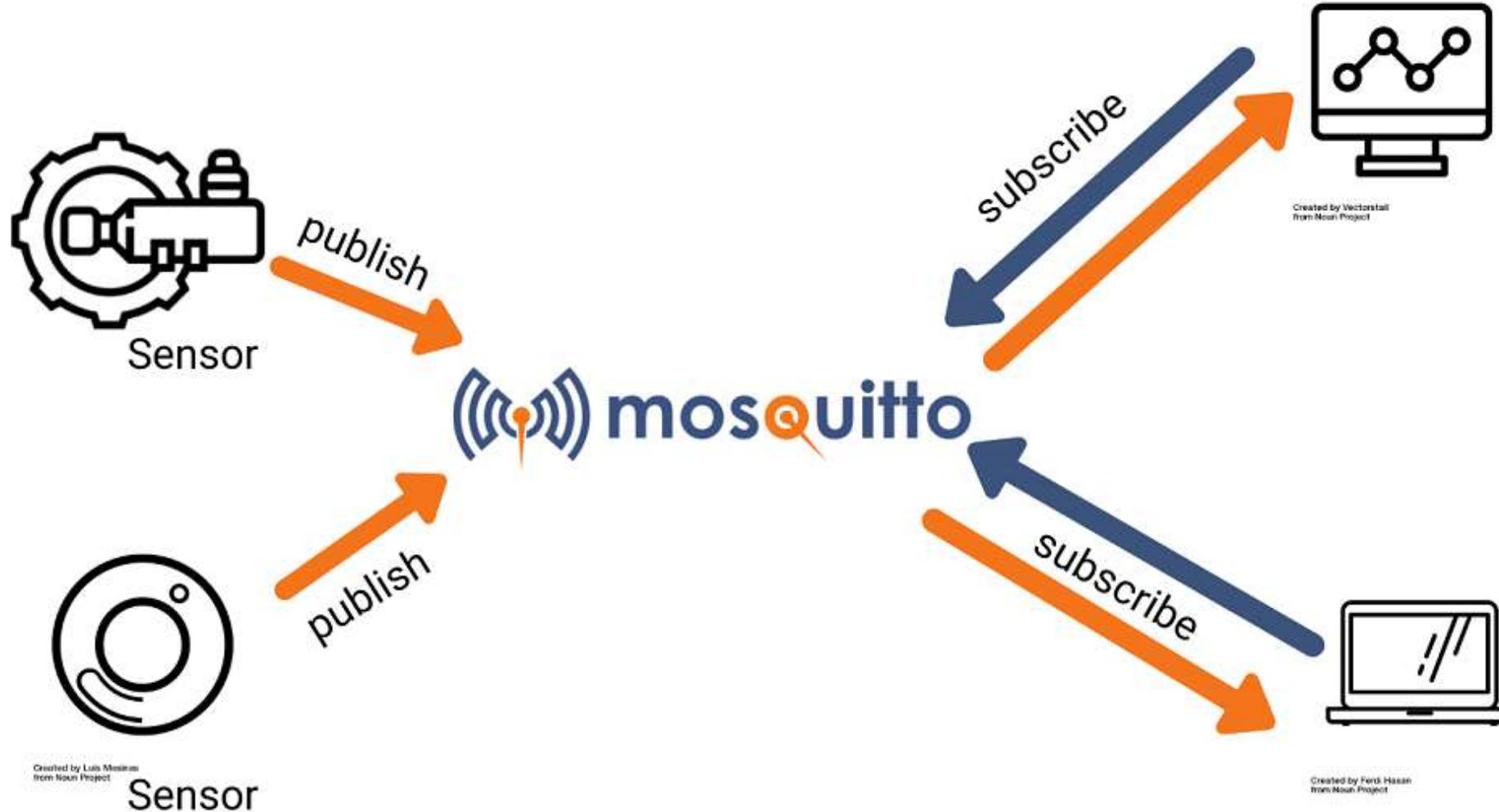


<https://mqtt.org/>

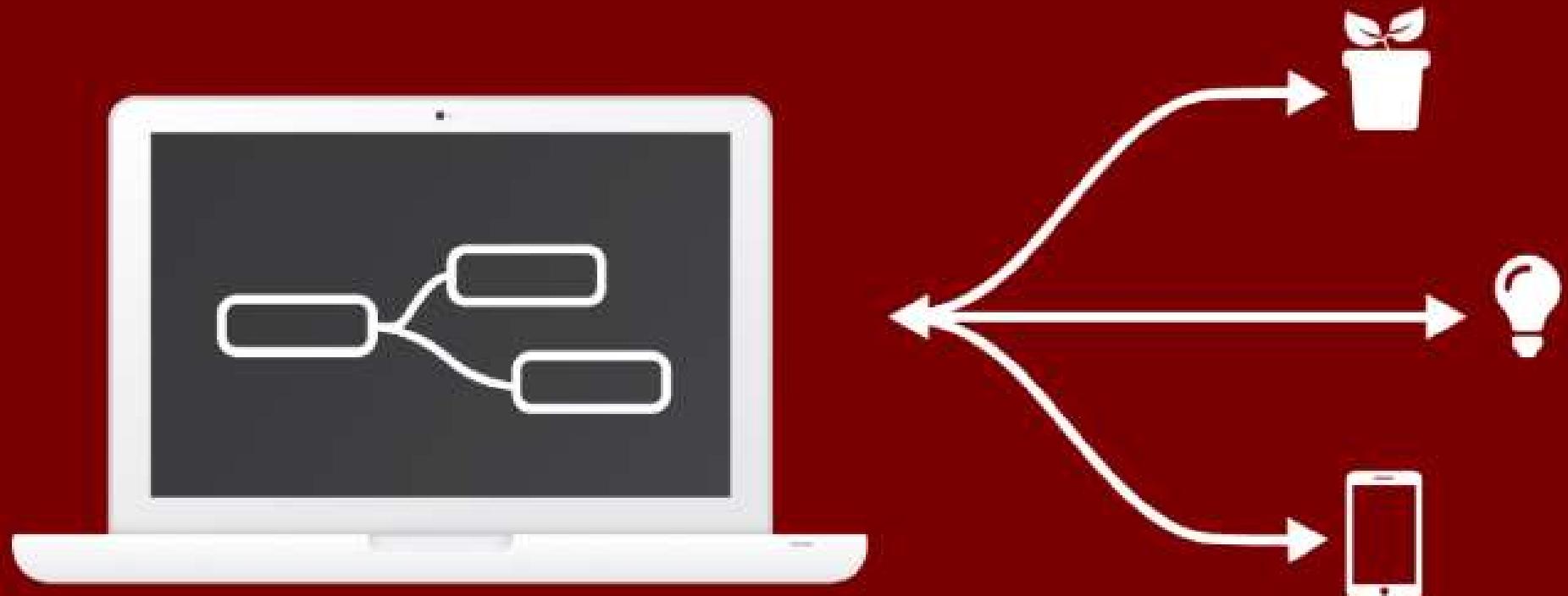
Arquitetura



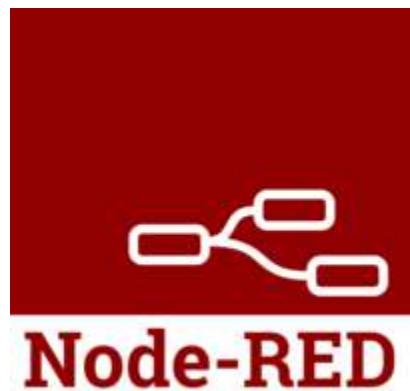
- É um *message broker* de código aberto, que implementa o protocolo MQTT (versões 5, 3.1.1 e 3.1).
- É bastante leve, adequado para computadores de placa única até servidores, o que o torna adequado para mensagens da Internet das Coisas, como sensores de baixa potência ou dispositivos móveis, como telefones, 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

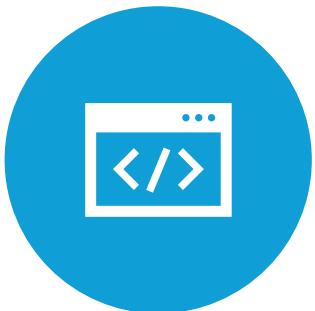
Node-RED::nós



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

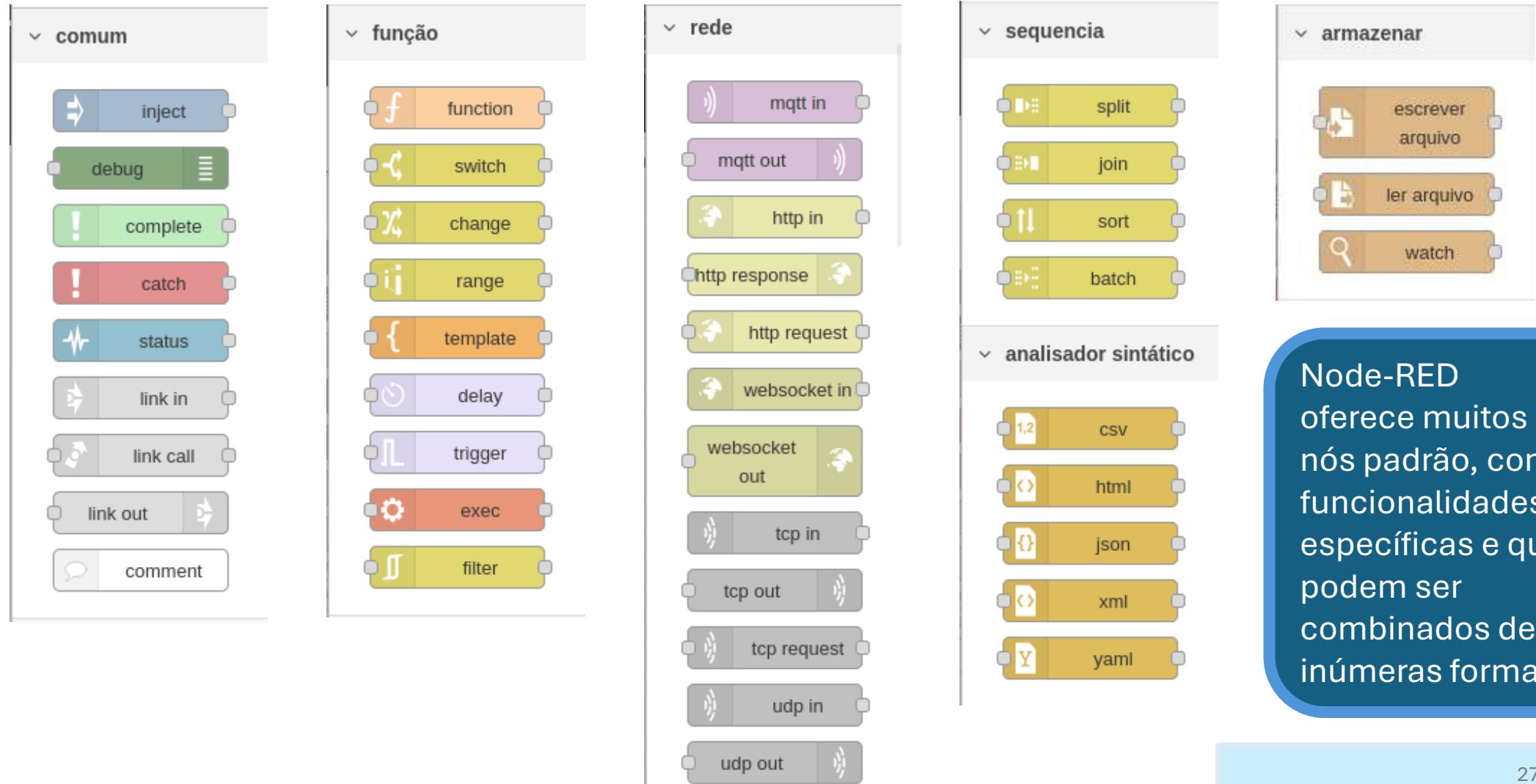
- Nós **padrão** (*default*) da plataforma e
- Nós de **contribuição**, criados pelo usuário/programador e que podem formar uma biblioteca compartilhada.

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

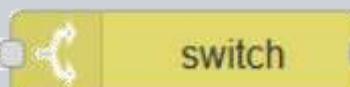
Natureza

- 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.

Nó	Funcionalidade
 inject	Injeta uma mensagem em um fluxo, manualmente ou em intervalos regulares, permitindo iniciar um fluxo com um texto, carimbo de tempo ou objeto.
 debug	Exibe a mensagem, alguma de suas propriedades ou uma expressão JSON, podendo ser habilitado ou desabilitado à critério do usuário.
 function	Bloco que permite a definição de uma função JavaScript que pode atuar sobre a mensagem recebida, transformando-a ou produzindo outras ações. Pode não retornar um resultando, parando um fluxo.
 template	Possibilita a definição de um padrão que pode ser preenchido com propriedades da mensagem ou o resultado de expressões utilizando a notação <i>Mustache</i> : “Esta é a mensagem: {{payload}}”, onde <i>payload</i> é substituído pela propriedade msg.payload .
 switch	Permite desviar o fluxo de mensagem para um caminho/rota específico, baseado na avaliação das propriedades da mensagem ou do contexto.
 change	Possibilita modificar uma propriedade da mensagem sem recorrer a uma função, aplicando suas operações (<i>Set</i> , <i>Change</i> , <i>Move</i> , <i>Delete</i>) na ordem desejada.

Nó	Funcionalidade
 json	Realiza a conversão de uma string JSON (<i>JavaScript Object Notation</i>) para um objeto JavaScript e vice-versa.
 xml	Realiza a conversão de uma string XML (<i>eXtensible Markup Language</i>) para um objeto JavaScript e vice-versa.
 csv	Realiza a conversão de uma string CSV (<i>Comma Separated Values</i>) para um objeto JavaScript e vice-versa.
 escrever arquivo	Efetua a gravação da mensagem em um arquivo, adicionando ao final ou substituindo o conteúdo existente. Alternativamente, pode excluir o arquivo.
 ler arquivo	Lê o conteúdo de um arquivo, como texto ou binário, armazenando-o temporariamente.
 exec	Executa um comando do sistema, e retorna sua saída.

Nó	Funcionalidade
 mqtt in	Efetua a recepção de mensagens de um tópico MQTT específico (<i>subscribe</i>) em um MQTT broker.
mqtt out 	Efetua a publicação de mensagens em um tópico MQTT específico (<i>publish</i>) em um MQTT broker.
 http in	Cria um <i>endpoint</i> HTTP que permite definir um serviço web recepção de mensagens.
http response 	Envia solicitações HTTP, permitindo o uso de métodos diferentes, retornando a resposta recebida.
 split	Divide uma mensagem em uma série de mensagens.
 join	Reúne uma série de mensagens em uma mensagem única.

Node-RED::interface

The screenshot displays the Node-RED interface with the following components:

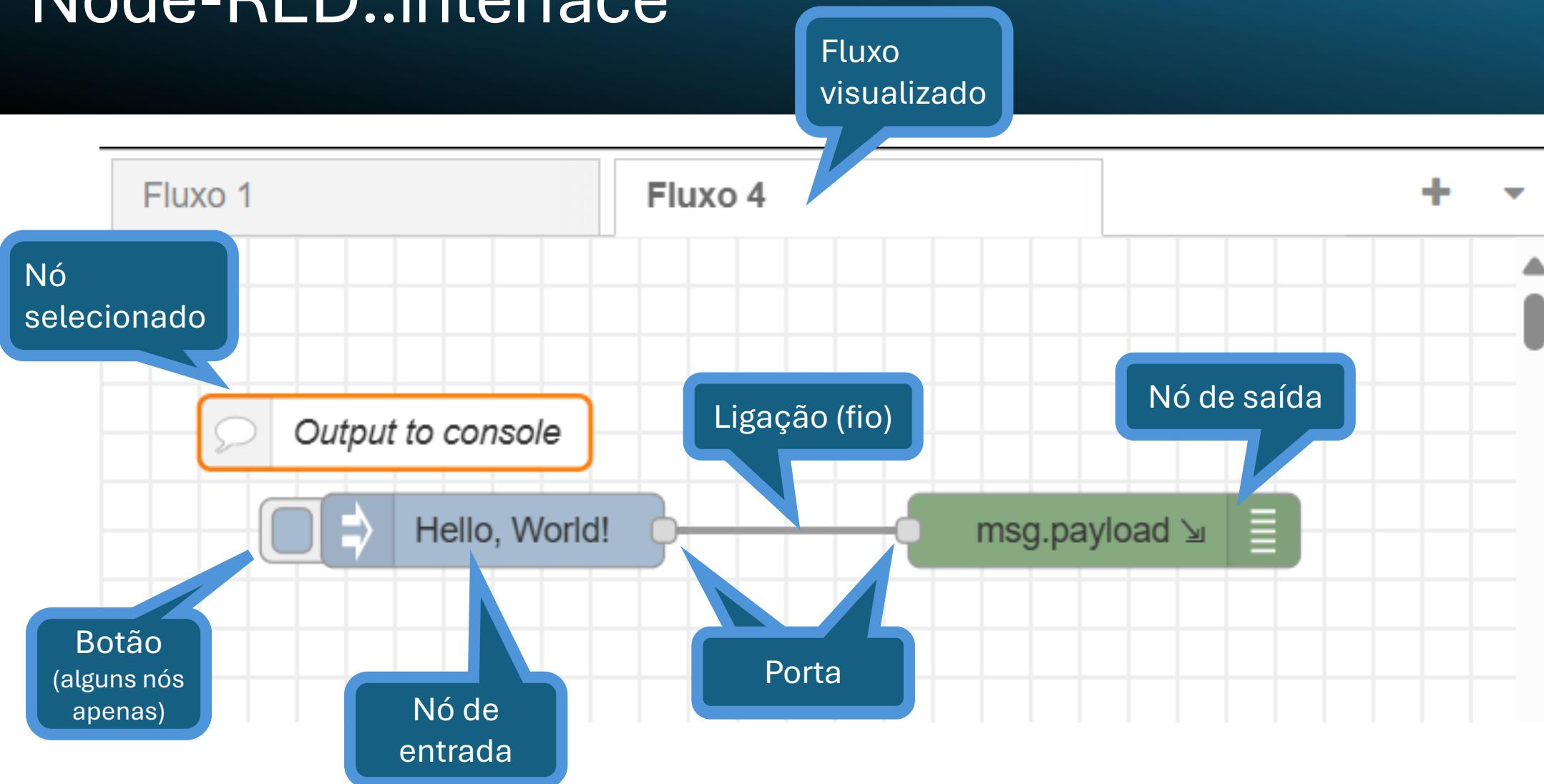
- Paleta de Nós (Palette):** Located on the left, it contains a search bar and two sections: "comum" and "função". The "comum" section includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The "função" section is currently collapsed.
- Editor de fluxos (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, which then connects to an "Output to console" node. A tooltip for this node reads: "Output to console
Debug node can be used to output values to console."
- Informações (Information Sidebar):** On the right, it has tabs for "informações", "Ajuda", "Depuração", "Configurações", and "Contexto". The "informações" tab is active, showing details for the selected "Output to console" node: "Nó: fb1c3ce9.c29ee", "Tipo: comment". Below this, there is a description of the node's function.

Paleta de
Nós

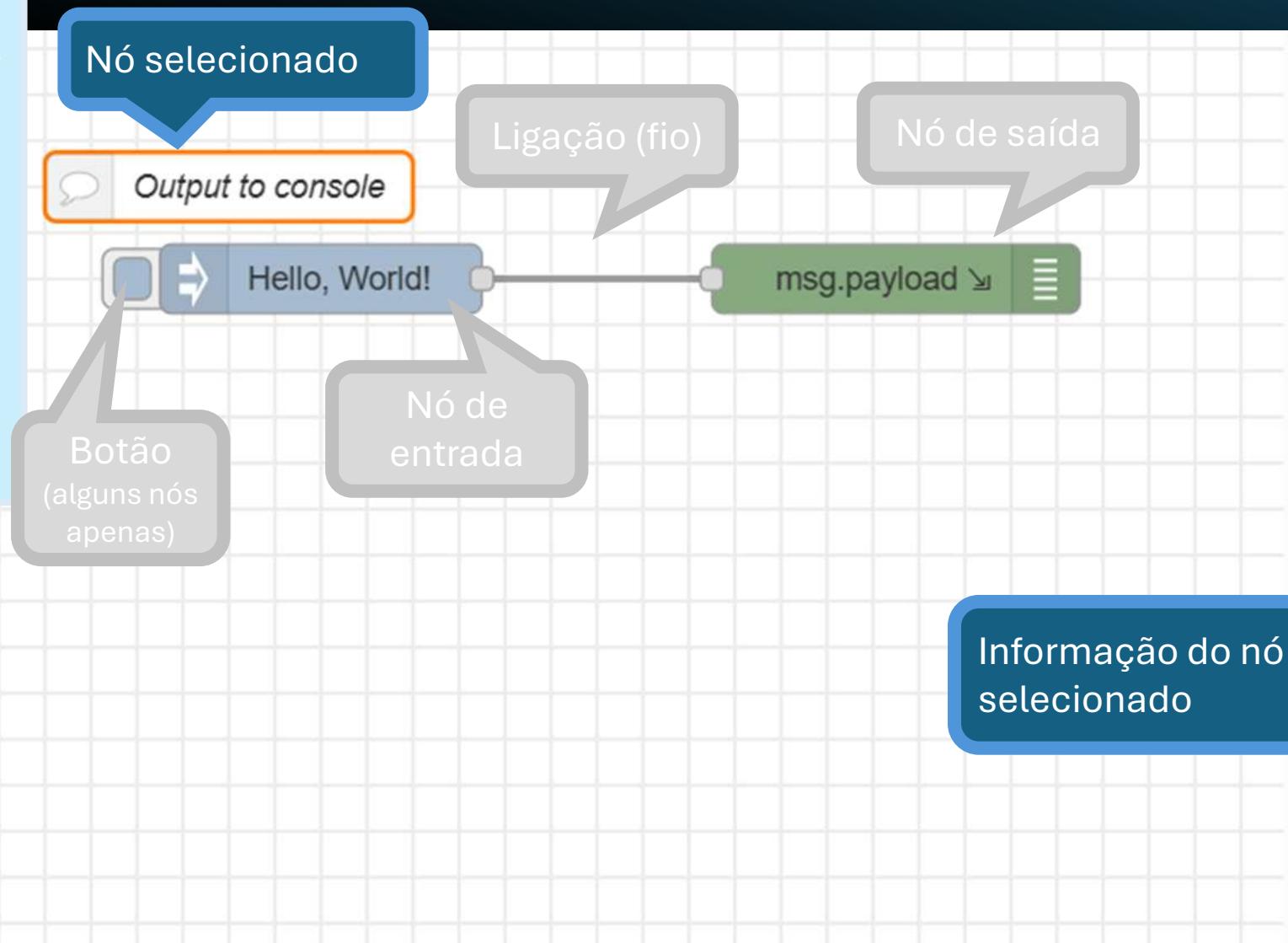
Editor de
fluxos

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

Node-RED::interface



Node-RED::interface



The sidebar shows the following structure:

- Fluxos** (Flows):
 - > Fluxo 1
 - < Fluxo 4
 - Fluxo selecionado** (Selected Flow): Points to the highlighted "Output to console" node in the list.
 - Output to console
 - Hello, World!
 - msg.payload ↴

Output to console

Nó	"fb1c3ce9.c29ee"
Tipo	comment

mostrar mais ▾

Output to console

Debug node can be used to output values to console.

27/06/2024 29

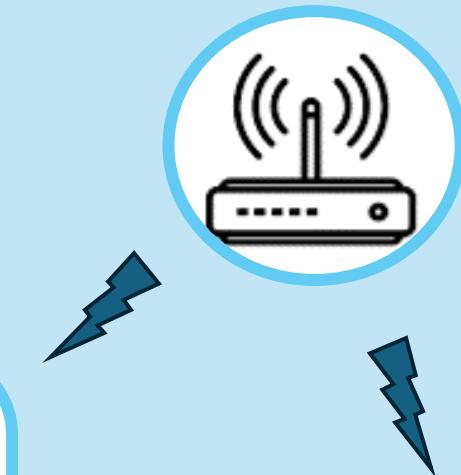


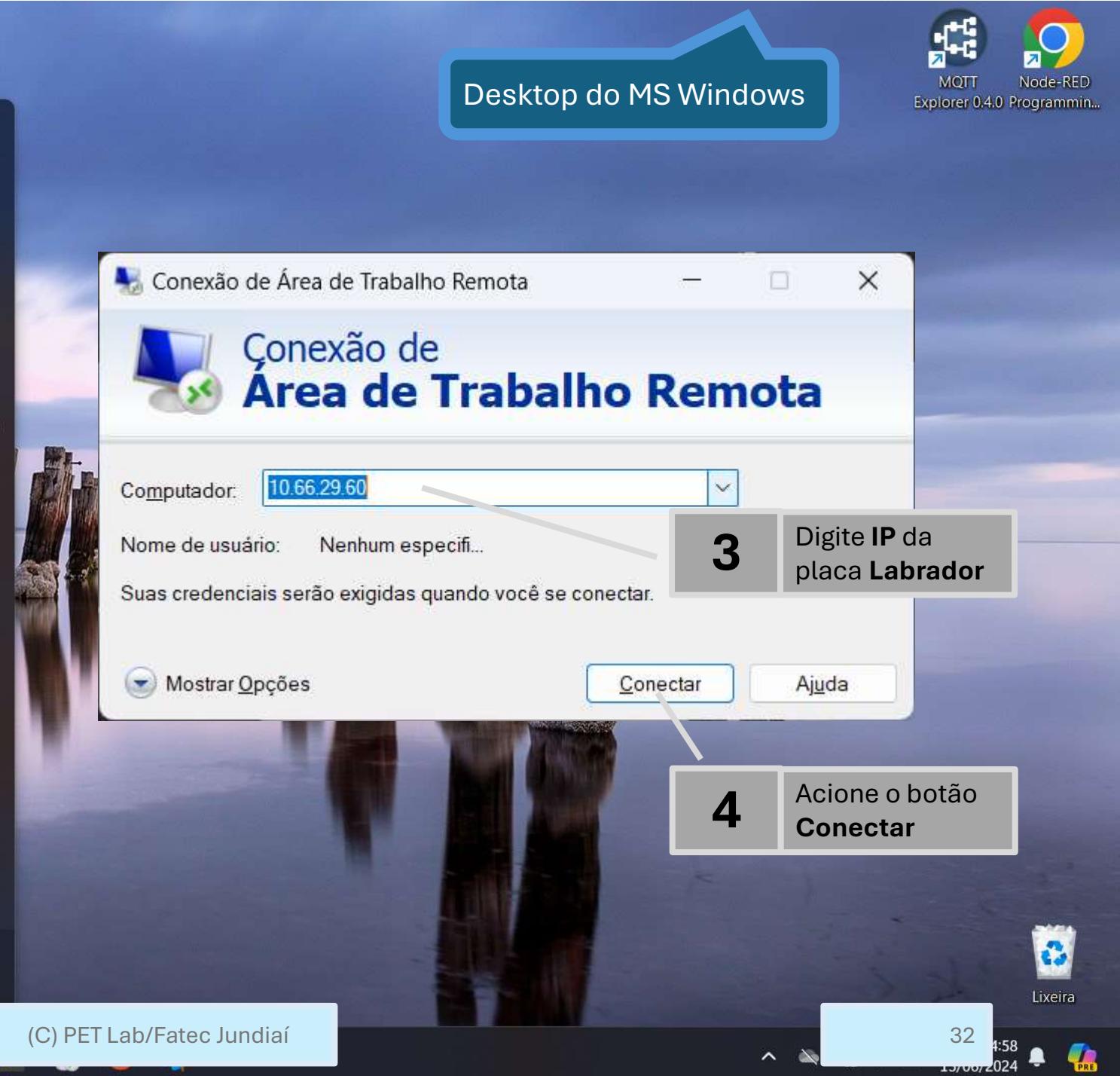
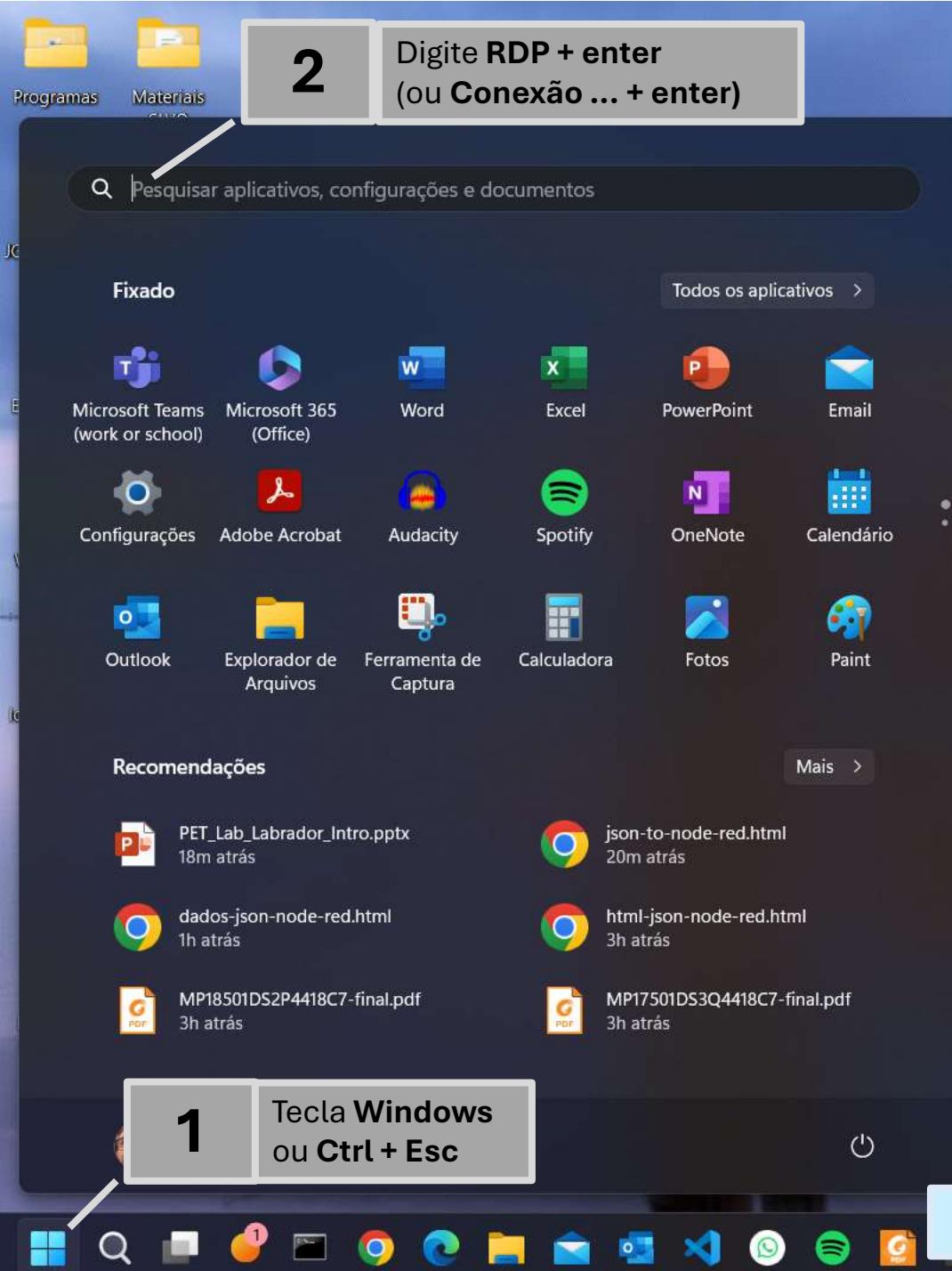
Hands-On

Cenário & Roteiro

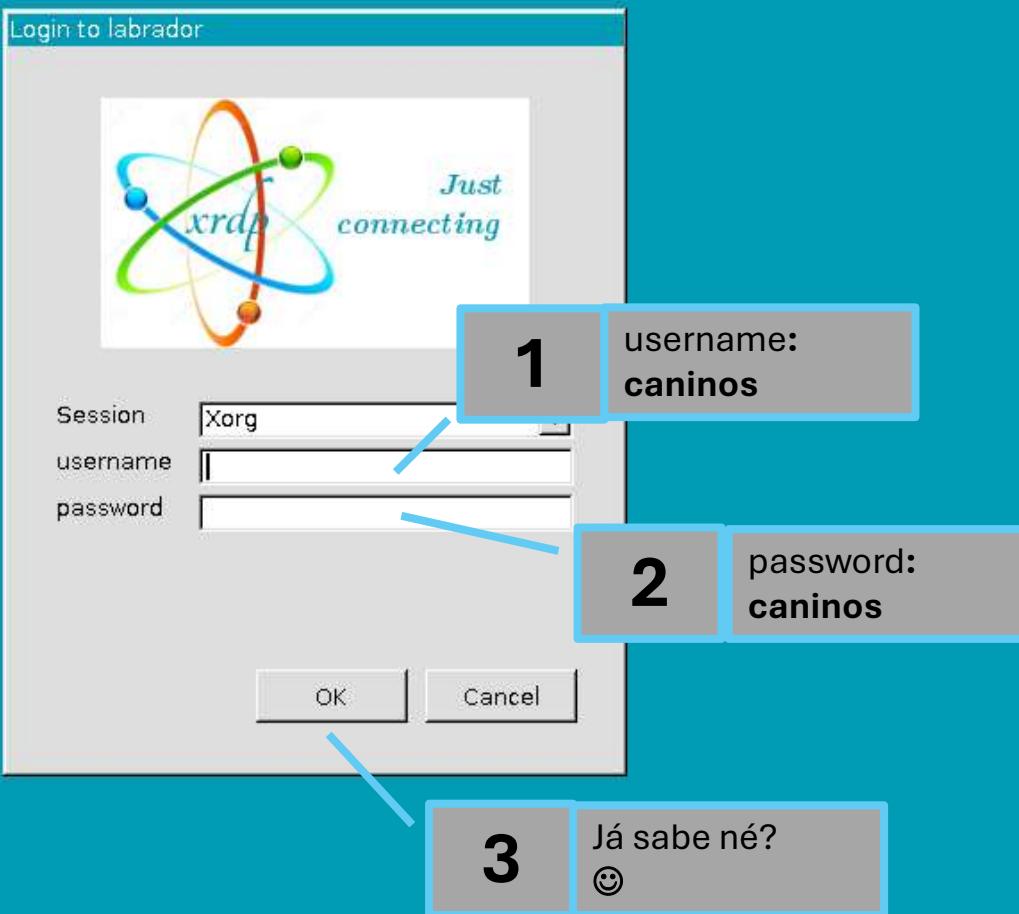


1. Setup inicial das placas Labrador:
 - SO Debian 11
 - Mosquitto
 - Node.js + Node-RED
 - xRDP
2. Conexão (via RDP) entre Notebook/Windows e Labrador/Debian
3. Utilização do Node-RED





Tela de login do xRDP
em execução no
Labrador e acessada
remotamente



Desktop
semelhante ao
MS Windows

1

Abra um
LXTerminal

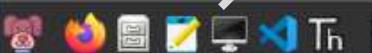
```
caninos@labrador: ~
Arquivo Editar Abas Ajuda
caninos@labrador: $ node-red
```

2

Digite comando
node-red



Lixeira



caninos@labrado...

(C) PET Lab/Fatec Jundiaí

34

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

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
```

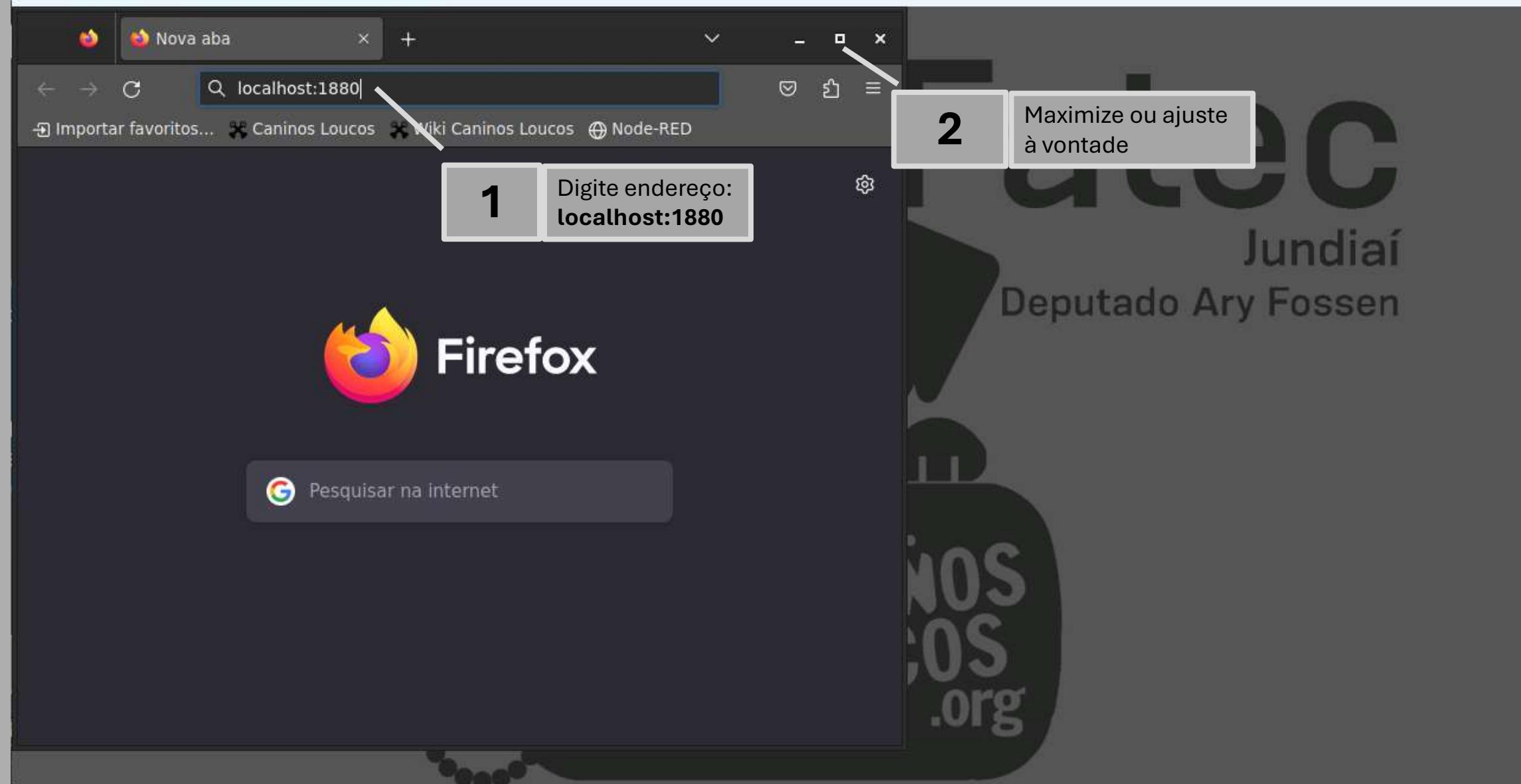


Lixeira

2

Abra o Firefox
(navegador web)





27/06/2024

[caninos@labrad...

Mozilla Firefox

(C) PET Lab/Fatec Jundiaí

36

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 callout box containing a number and text. Below the inputs is a "Ingressar" button. A blue speech bubble on the left contains the explanatory text.

Nome do Usuário:
caninos

Senha:
caninos

Ingressar

1 Nome do Usuário:
caninos

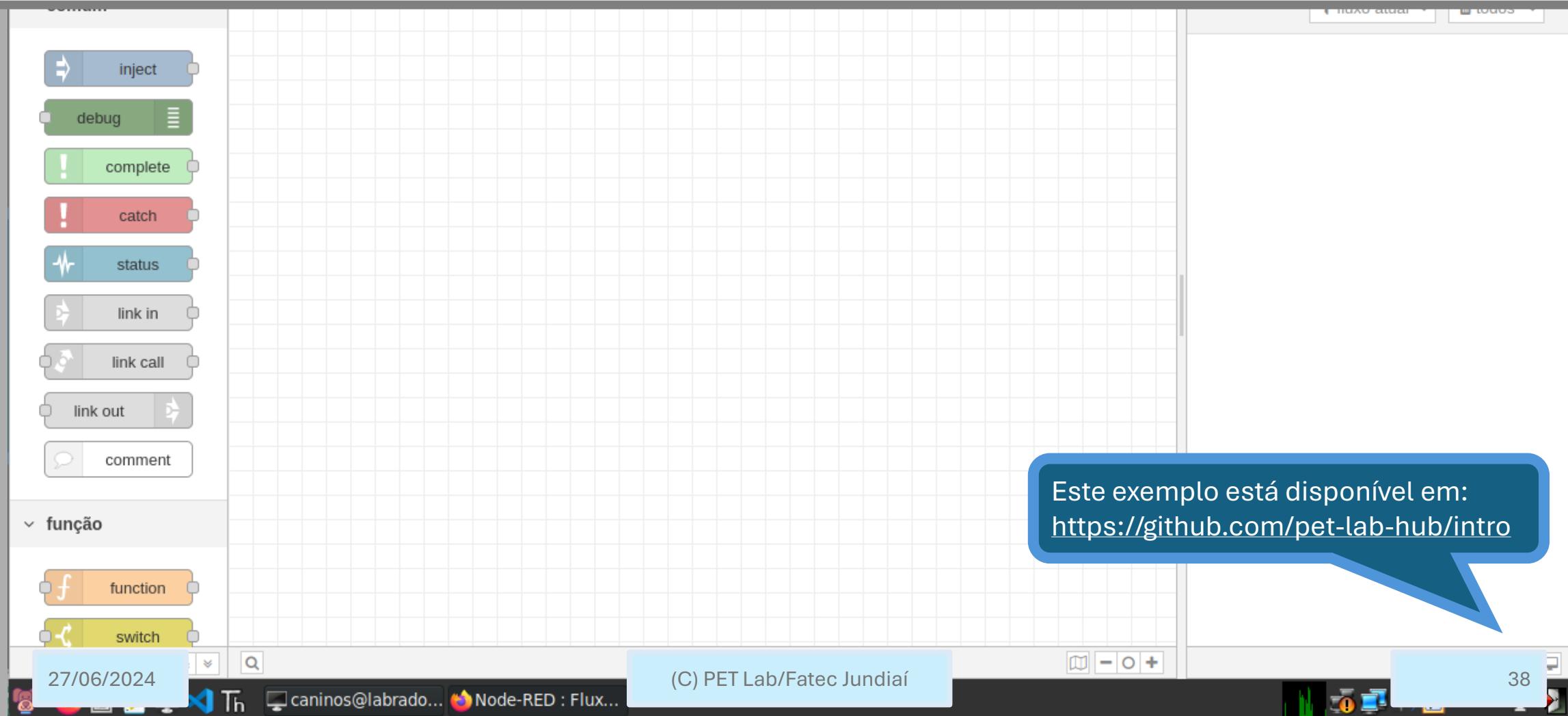
2 Senha:
caninos

3 Já sabe né?
😊

(C) PET Lab/Fatec Jundiaí

37

Exemplo 1 [básico]



10.66.29.60 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/a15badd50f929050

Node-RED

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

implementar depurar

fluxo atual todos

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

39

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 interaccao
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

27/06/2024

(C) PET Lab/Fatec Jundiaí

Unlocked

40

```
graph LR; N1[1 Altere o nome como desejado] --> N2[2 Opcionalmente forneça uma descrição]; N2 --> N3[3 Acione botão Feito para salvar e voltar ao fluxo]
```

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

27/06/2024

(C) PET Lab/Fatec Jundiaí

41

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.

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

42

```
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

27/06/2024

(C) PET Lab/Fatec Jundiaí

43

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

3

Selecione aba **debug**
no painel lateral

filtrar nós

comum

inject

debug

complete

catch

status

link in

link call

link out

comment

função

function

switch

Ola!

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



2

Observe a notificação (*toast popup*) no console do Node-Red

Carimbo de data/hora

debug 1

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.

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

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

45

```
graph LR; inject[inject] --> Carimbo1[Carimbo de data/hora]; inject --> Carimbo2[Carimbo de data/hora]; Carimbo1 --> msg[Msg]; Carimbo2 --> msg; msg --> publish[publish]
```

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

debug 1

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

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

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. The top right corner has a box labeled '1 Implemente as alterações' with a blue arrow pointing to the top bar. The bottom right corner has a box labeled '3 Observe o resultado da injeção' with a blue arrow pointing to the debug log on the right. The debug log shows two messages: one at 18:13:23 with payload "Um ola do Node-RED!" and another at 18:13:27 with payload 1718313207514. A central box contains the text 'Um nó pode receber entradas de um, dois ou mais nós.'

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! ...

47

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

27/06/2024

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

48

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

Depurador:

- 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"

Fluxo de trabalho:

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

Componentes disponíveis:

- comum: inject, debug, complete, catch, status, link in, link call, link out, comment
- função: function, switch

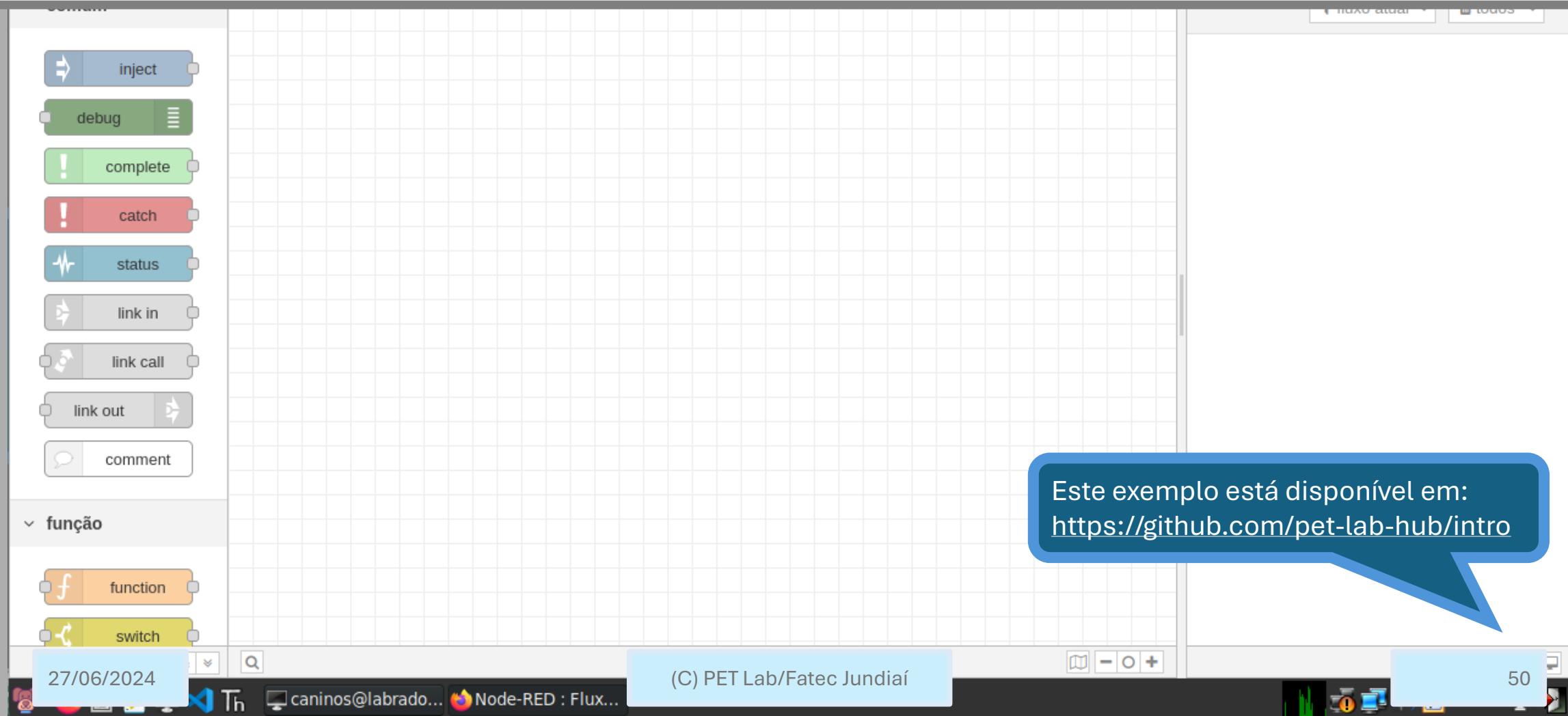
Informações:

- 27/06/2024
- caninos@labrado... Node-RED : Ola! ...

(C) PET Lab/Fatec Jundiaí

49

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] --> timestamp[Carimbo de data/hora]; timestamp --> function[Formatar Data/Hora]; function --> debug1[debug 1]
```

(C) PET Lab/Fatec Jundiaí

27/06/2024

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

51

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 2

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

Fluxo 2

implementar

depurar

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

27/06/2024

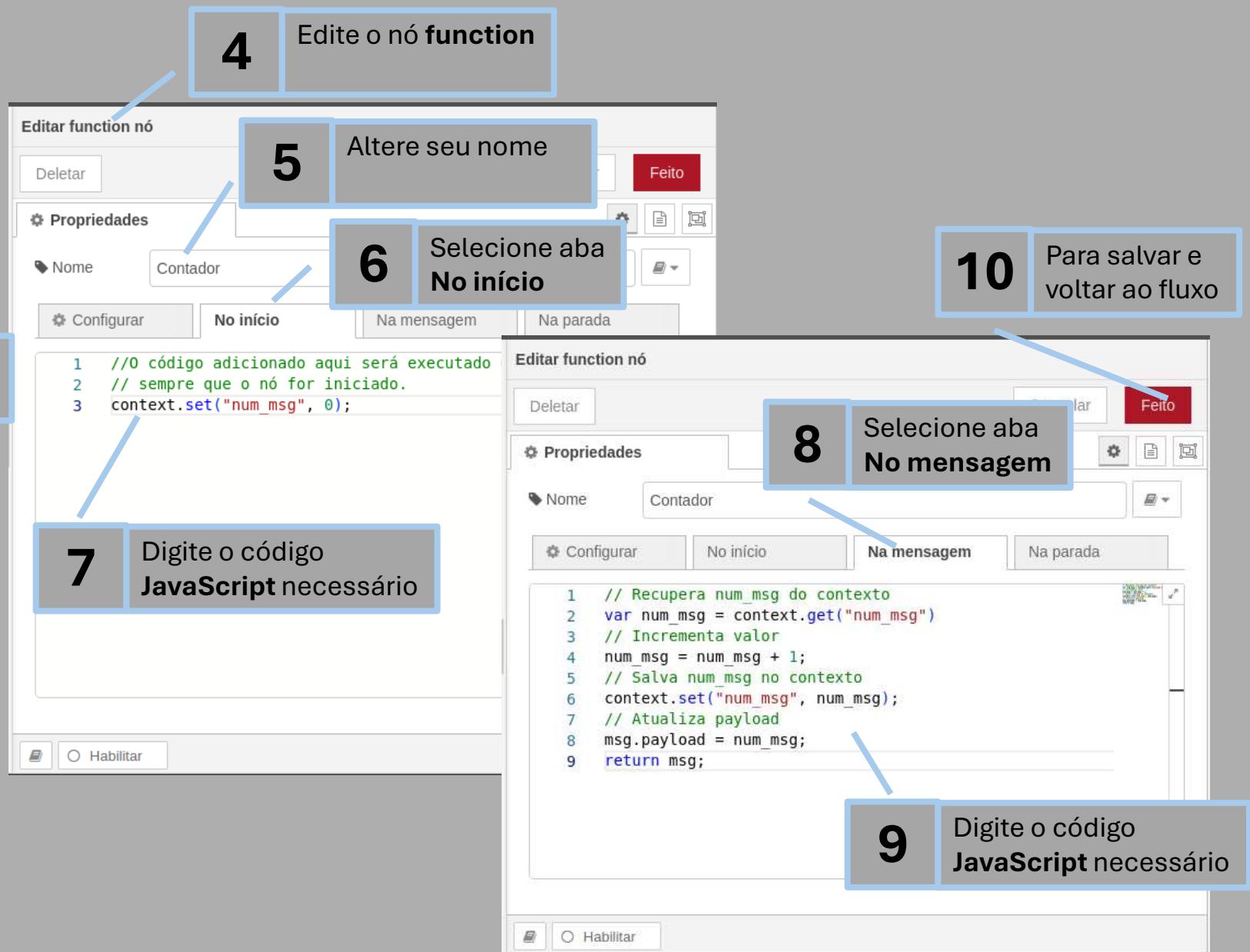
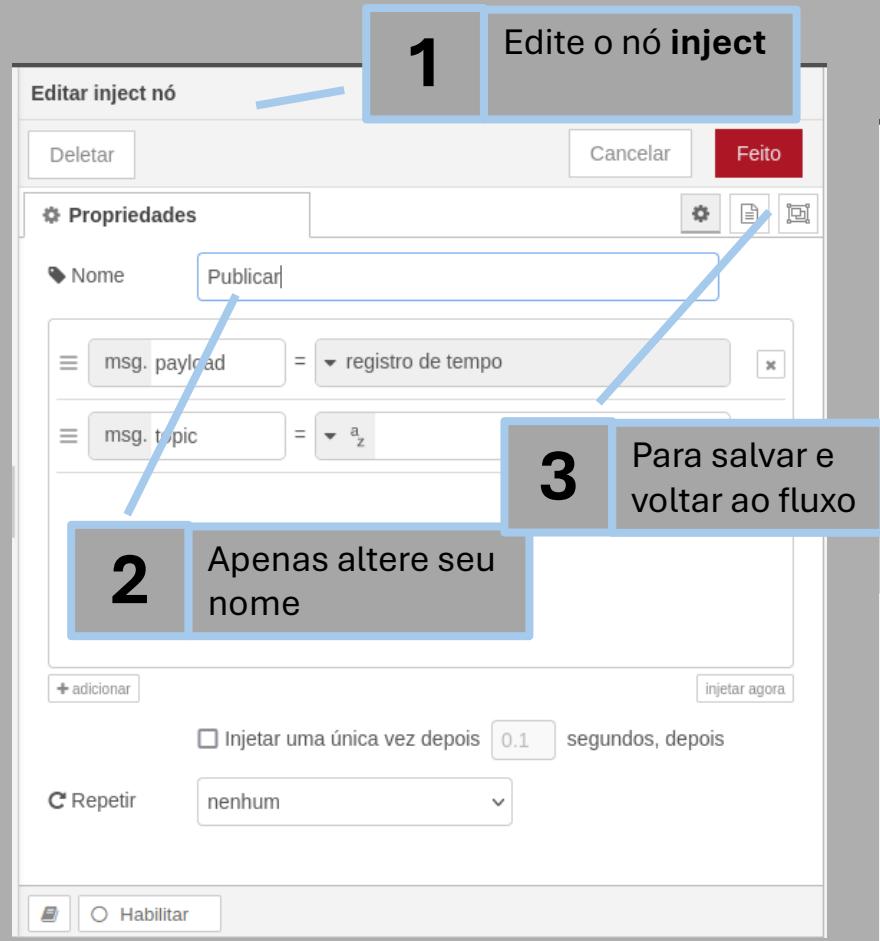
(C) PET Lab/Fatec Jundiaí

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

52

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

6 Indique o IP do broker MQTT desejado

7 Para salvar e voltar ao fluxo

MQTT QoS (*Quality of Service*)
 0 → Envio sem garantia de entrega
 1 → Envio com garantia de entrega, mas pode haver repetição
 2 → Envio com garantia de entrega única

Propriedades

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

Dica: deixe o tópico, qos ou retenha em branco se quiser definir-los por meio das propriedades da mensagem.

Propriedades

- Nome: Nome
- Conexão: Servidor: por exemplo, localhost, Porta: 1883
- Protocolo: MQTT V3.1.1
- Protocolo: Usar sessão limpa (checkbox checked)

Em todos os fluxos

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

13/06/2024, 22:00:03 nó: debug 2
msg.payload : number

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

13/06/2024, 22:00:11 nó: debug 2
msg.payload : number

Ola!

MQTT Count

inject

debug

complete

catch

status

link in

link call

link out

comment

função

rede

entrada

contagem

Contador

debug 2

27/06/2024

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

(C) PET Lab/Fatec Jundiaí

57

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

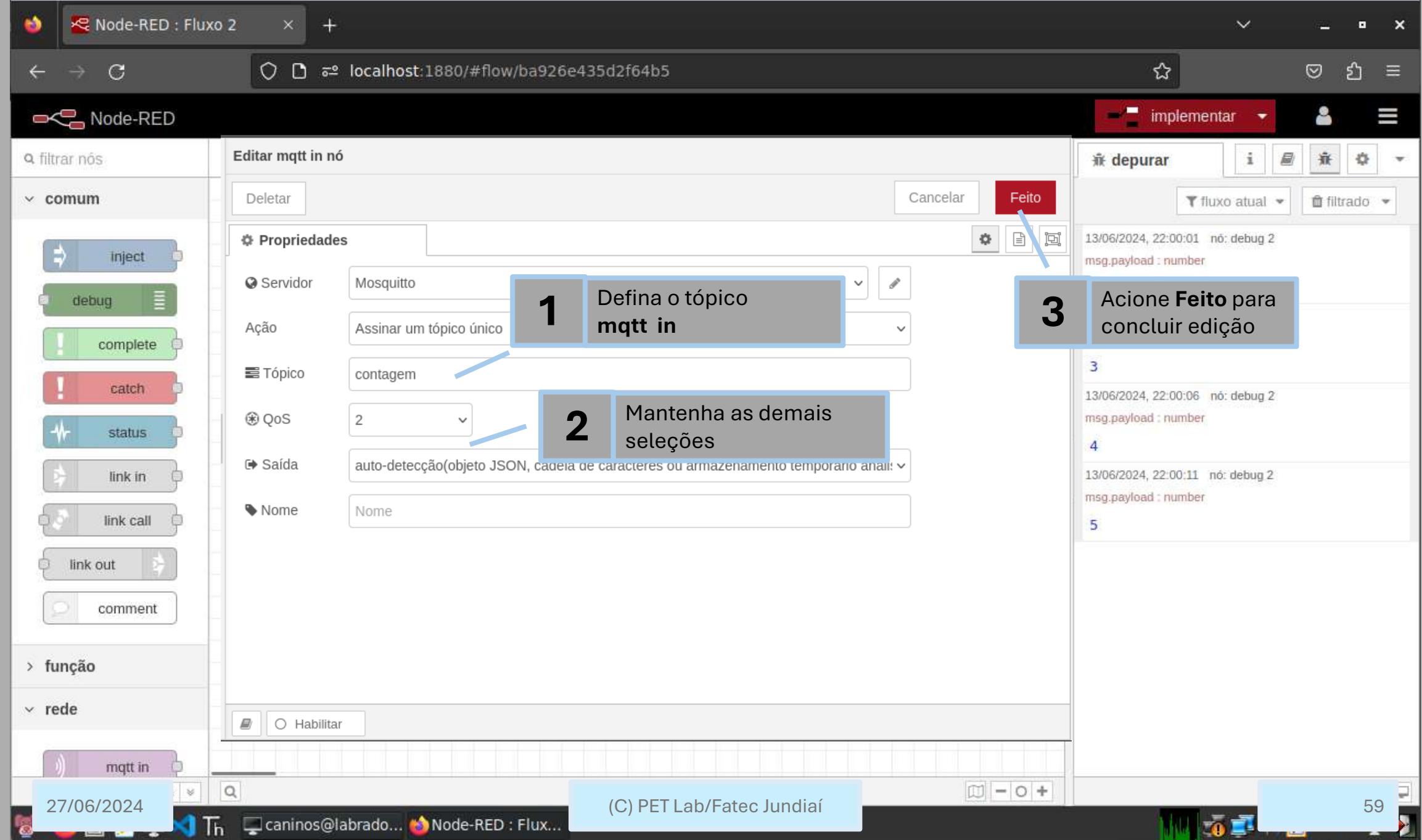
13/06/2024, 22:00:01 nó: debug 2
msg.payload : number
2
13/06/2024, 22:00:03 nó: debug 2
msg.payload : number
3
13/06/2024, 22:00:06 nó: debug 2
msg.payload : number
4
13/06/2024, 22:00:11 nó: debug 2
msg.payload : number
5

27/06/2024

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

(C) PET Lab/Fatec Jundiaí

58



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 --> contagem[contagem]; contagem --> debug2[debug 2]; debug2 --> debug3[debug 3];
```

1

Implemente as alterações

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**.

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

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

61

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

...

Saída

J: payload

Para

janela de depuração

console do sistema

estado do nó (32 caracteres)

Nome

Par

5

Altere no nome

8

Para salvar e
voltar ao fluxo

Cancelar

Feito

Propriedades

...

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 Faça as ligações conforme indicado

Observe que o nó **switch** oferece uma porta de saída para cada condição adicionada em sua configuração.

17/06/2024, 11:38:12 - nó: debug 2
msg.payload : number

2

17/06/2024, 11:38:13 - nó: debug 3
contagem : msg.payload : number

2

17/06/2024, 11:38:13 - nó: Par
contagem : msg : number

2

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

3

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

3

17/06/2024, 11:38:20 - nó: Impar
contagem : msg.payload : number

3

```
graph LR; inject[inject] --> Publicar[Publicar]; Publicar --> Contador[Contador]; Contador --> contagem[contagem]; contagem --> debug2[debug 2]; contagem --> interruptor[interruptor]; interruptor --> Par[Par]; interruptor --> Impar[Impar]
```

Q. filtrar nós

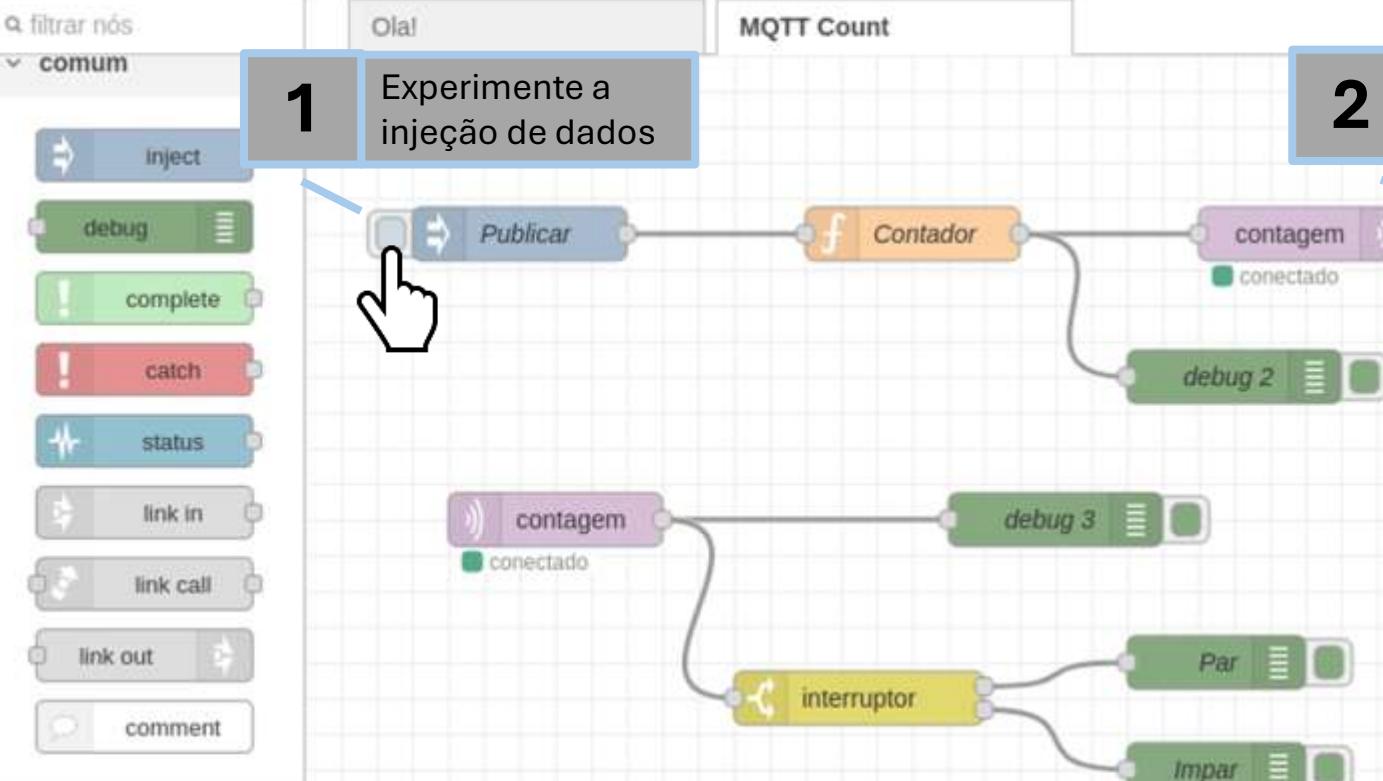
1

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

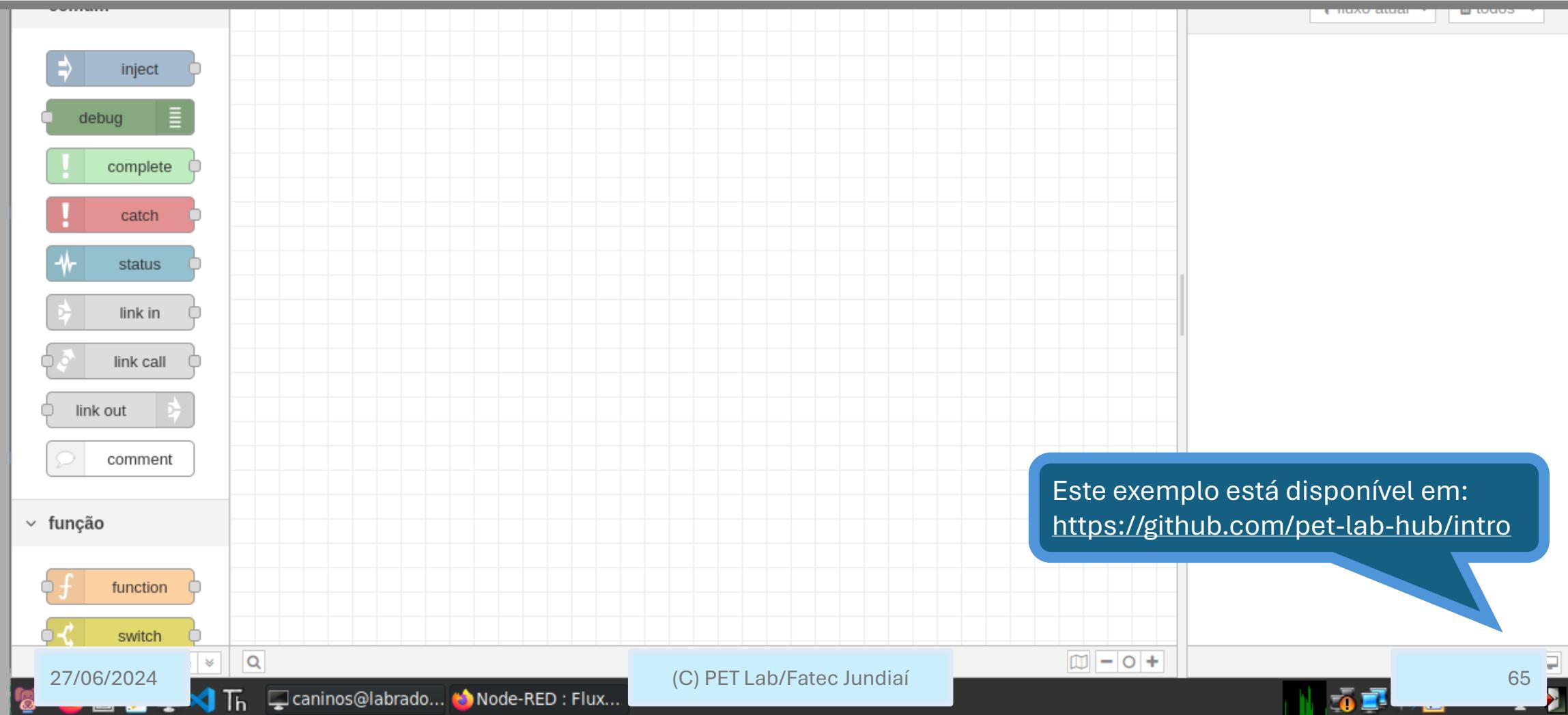
3
17/06/2024, 11:38:20 node: debug 3
contagem : msg payload : number

17/06/2024, 11:38:20 nó: Impar
contagem: msg.payload: number

```
17/06/2024, 12:05:47 no: debug 2  
msg.payload : number  
4
```

```
17/06/2024, 12:05:47 nº: debug 3  
contagem : msg.payload : number  
4  
17/06/2024, 12:05:47 nº: Par  
contagem : msg.payload : number
```

Exemplo 3 [HTTP & Node-RED]



192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : MQTT Count

localhost:1880/#flow/ba926e435d2f64b5

Node-RED

implementar

depurar

fluxo atual

filtrado

filtrar nós

Ola!

MQTT Count

comum

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

funcção

- function
- switch

1 Acione o + para adicionar um novo fluxo

The screenshot shows the Node-RED interface with a flow titled "MQTT Count". The flow starts with an "Olá!" node, followed by a "Publicar" node publishing to the topic "Contador". This is followed by a "Contador" function node. A connection from the "Contador" node leads to a "contagem" debug node. From the "contagem" node, two paths emerge: one to a "debug 2" node and another to an "interruptor" switch node. The "interruptor" node has two outputs: one to a "Par" debug node and one to an "Impar" debug node. The "contagem" node also has a "conectado" status indicator. A large blue callout box with the number "1" and the text "Acione o + para adicionar um novo fluxo" is positioned on the right side of the canvas.

```
graph LR; Olala[Olá!] --> Publicar[Publicar]; Publicar --> Contador[Contador]; Contador --> Contagem[contagem]; Contagem --> Debug2[debug 2]; Contagem --> Interruptor[interruptor]; Interruptor --> Par[Par]; Interruptor --> Impar[Impar];
```

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

66

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : HTTP & MQTT

localhost:1880/#flow/e73cab525a3ed036

Node-RED

filtrar nós

Ola!

MQTT Count

comum

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

função

- function
- switch

27/06/2024

(C) PET Lab/Fatec Jundiaí

Unlocked

implementar

depurar

fluxo atual

filtrado

20/06/2024, 18:48:08, nó: dobrar 1
msg.payload
2024

20/06/2024, 18:48:29, nó: debug 4
msg.payload : Object
{ string: "Sistemas Embarcados",
int: 10 }

1 Altere o nome como desejado

2 Opcionalmente forneça uma descrição

3 Para salvar e voltar ao fluxo

Nome: HTTP & Node-RED

Descrição:

1

Cancelar Feito

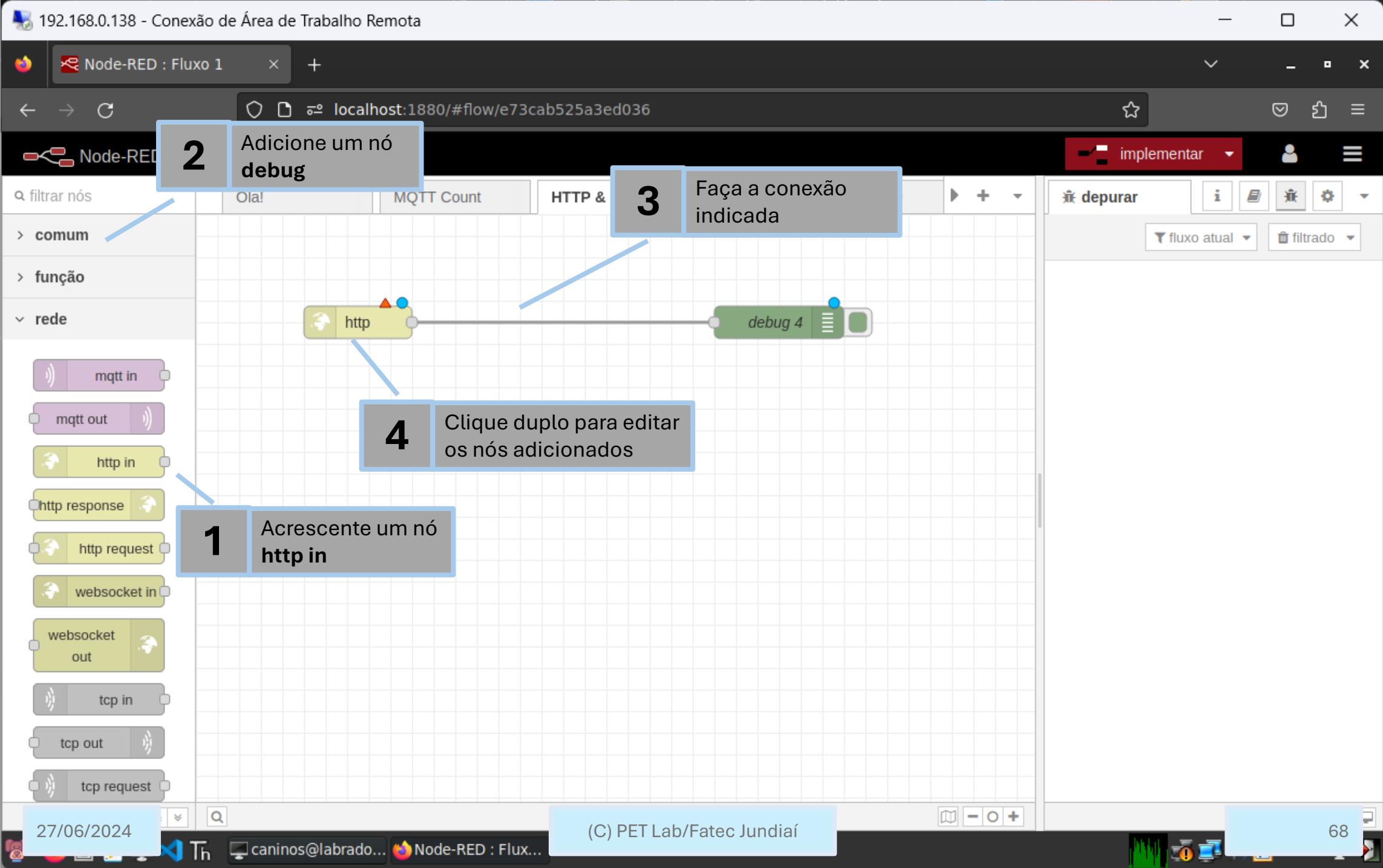
HTTP & Node-RED

1

2

3

67



Editar http in nó

1 Selecione o método **POST**

2 Defina a **url** de recepção de dados

3 Opcionalmente defina um nome para o **endpoint**

4 Acione Feito para concluir a edição

Propriedades

Método: POST
 Aceitar Subir arquivos?
 URL /dados
 Nome Nome

Deletar **Cancelar** **Feito**

Editar debug nó

5 Marque a opção **estado do nó**

6 Escolha **contador de mensagens**

7 Opcionalmente defina um nome para o nó

8 Acione Feito para concluir a edição

Propriedades

Saída: msg. payload
 Para janela de depuração
 console do sistema
 estado do nó (32 caracteres)

contador de mensagem
 Nome debug 4

Deletar **Cancelar** **Feito**

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/e73cab525a3ed036

Node-RED

filtrar nós

Ola! MQTT Count HTTP & [post] /dados debug 4

implementar depurar

comum função rede mqtt in mqtt out http in http response http request websocket in websocket out tcp in tcp out tcp request

1 Implemente as alterações

2 Experimente o envio de dados por meio de um **formulário HTML**

Um **formulário HTML**, combinado com **JavaScript**, pode enviar dados **JSON** diretamente para fluxos do Node-RED. Aplicações capazes de enviar dados por meio do **protocolo HTTP** também podem fazer isso.

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

70

```
graph LR; POST[/dados] --> DEBUG[debug 4]
```

JSON

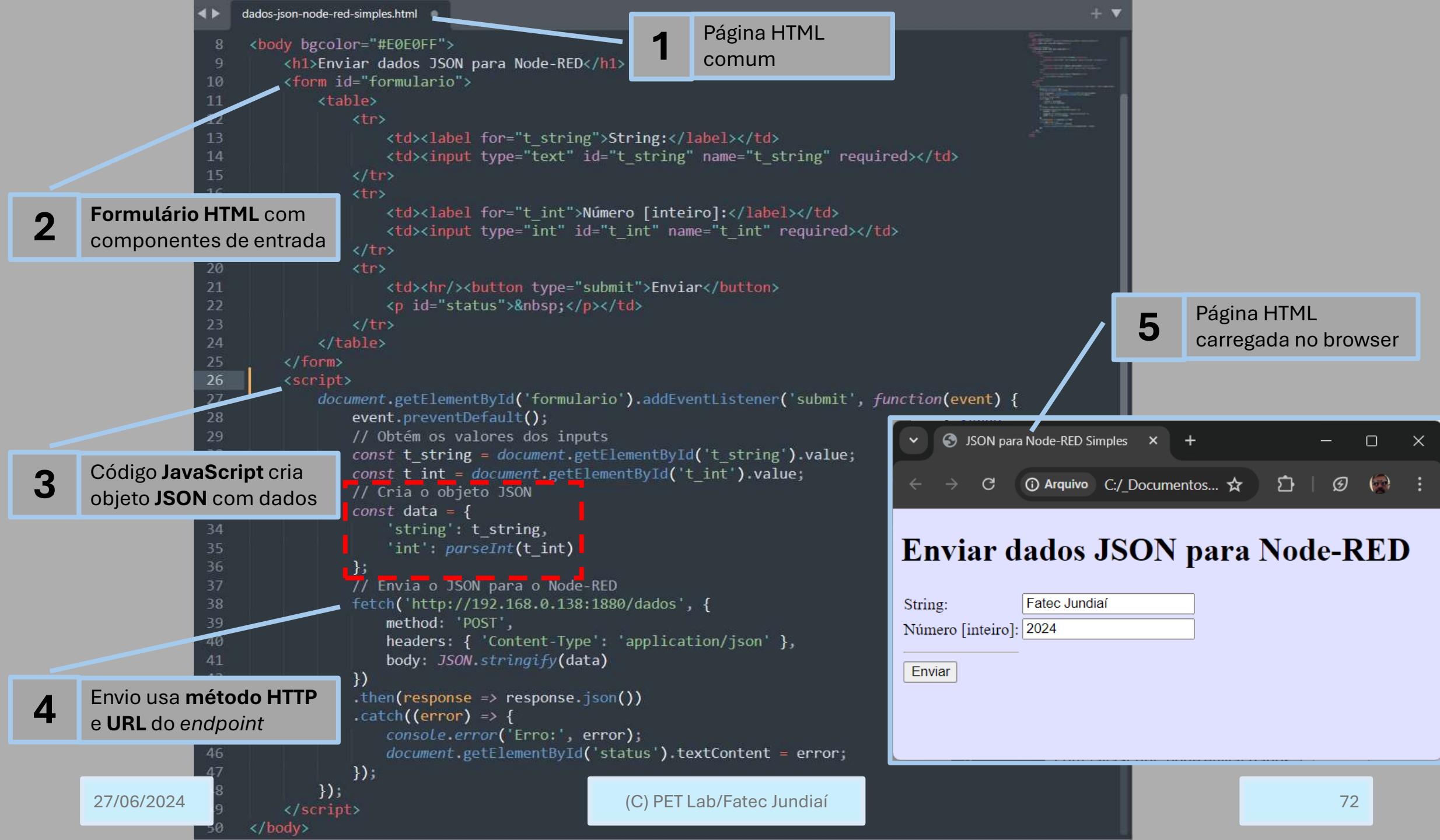
JavaScript Object Notation

- 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  
}
```



192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : HTTP & MQTT

localhost:1880/#flow/e73cab525a3ed036

Node-RED

1 Implemente as alterações

2 Experimente o envio de dados por meio de form HTML

Formulários HTML, combinados com JavaScript, permitem o envio direto de dados para fluxos do Node-RED (envio ponto-a-ponto).
[Vantagem] Solução não requer message brokers (MQTT ou outros)
[Desvantagem] Compartilhamento de dados se torna mais complexo.

debug 4

20/06/2024, 18:48:08 nó: debug 4
msg.payload : Object
► { string: "Fatec Jundiaí", int: 2024 }
20/06/2024, 18:48:29 nó: debug 4
msg.payload : Object
► { string: "Sistemas Embarcados", int: 10 }

depurar

fluxo atual

filtrado

JSON para Node-RED Simples

Enviar dados JSON para Node-RED

String: Sistemas Embarcados
Número [inteiro]: 10
Enviar

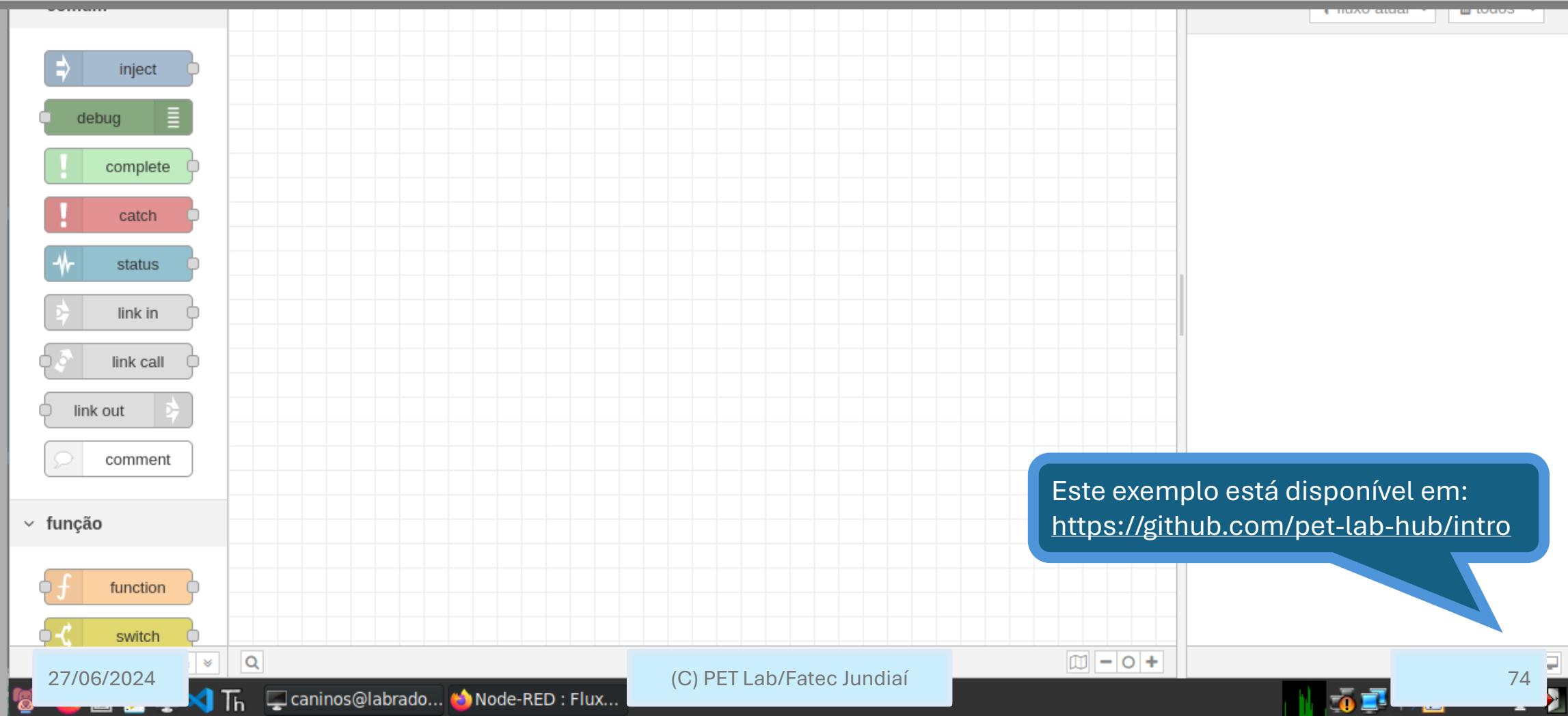
(C) PET Lab/Fatec Jundiaí

27/06/2024

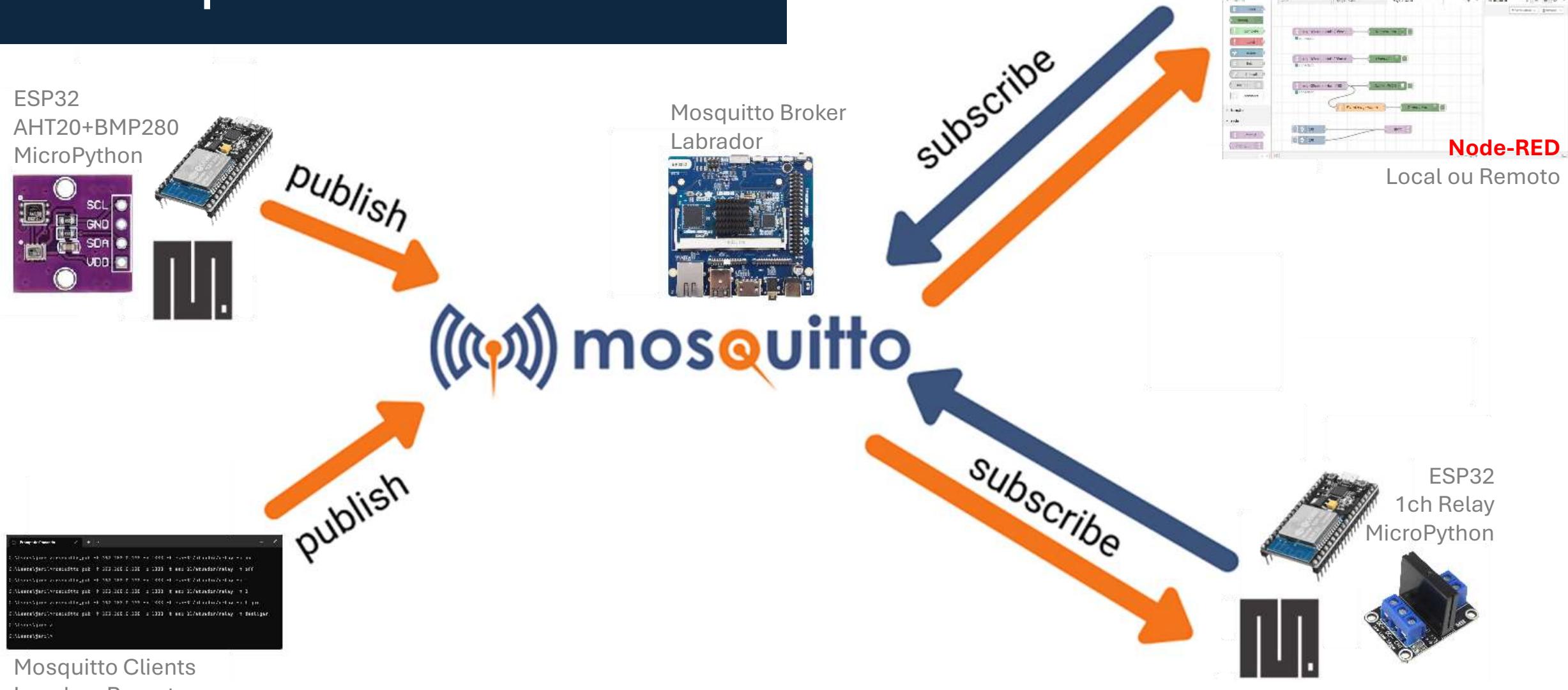
caninos@labrado... Node-RED : HTTP...

73

Exemplo 4 [MQTT & IoT]



Exemplo 3::Cenário



192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : HTTP & MQTT

localhost:1880/#flow/e73cab525a3ed036

Node-RED

1 Acione o + para adicionar um novo fluxo

implementar

depurar

fluxo atual filtrado

20/06/2024, 18:48:08 nó: debug 4
msg.payload : Object
▶ { string: "Fatec Jundiaí", int: 2024 }

20/06/2024, 18:48:29 nó: debug 4
msg.payload : Object
▶ { string: "Sistemas Embarcados", int: 10 }

inserir

Olá!

MQTT Count

HTTP & Node-RED

comum

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

função

- function
- switch

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

76

The screenshot shows the Node-RED interface with a single flow. It starts with an 'inject' node, followed by a 'debug' node with the ID '4'. A second 'debug' node with the ID '2' is connected to the output of the first. The flow is triggered by a 'POST /dados' message. The 'debug' nodes output the received payload as objects:

```
20/06/2024, 18:48:08 nó: debug 4
msg.payload : Object
▶ { string: "Fatec Jundiaí", int: 2024 }

20/06/2024, 18:48:29 nó: debug 4
msg.payload : Object
▶ { string: "Sistemas Embarcados", int: 10 }
```

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

implementar

depurar

fluxo atual

filtrado

filtrar nós

Ola!

MQTT

comum

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

função

- function
- switch

Editor Fluxo: MQTT Data

1 Altere o nome como desejado

2 Opcionalmente forneça uma descrição

3 Para salvar e voltar ao fluxo

Nome: MQTT Data

Descrição:

1

Cancelar Feito

(C) PET Lab/Fatec Jundiaí

Unlocked

77

27/06/2024

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

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**

Propriedades

Servidor: Mosquitto

Ação: Adicionar novo mqtt-broker...

Tópico: esp-02/sensor/aht20/temp

QoS: 2

Saída: auto-detecção(objeto JSON, cadeia de caracteres ou armazenamento temporário)

Nome: Nome

Habilitar

Cancelar | **Feito**

Propriedades

Nome: Mosquito Remoto

Conexão | **Segurança** | **Mensagens**

Servidor: 10.66.0.15 | Porta: 1883

Conectar automaticamente
 Usar TLS

3.1.1

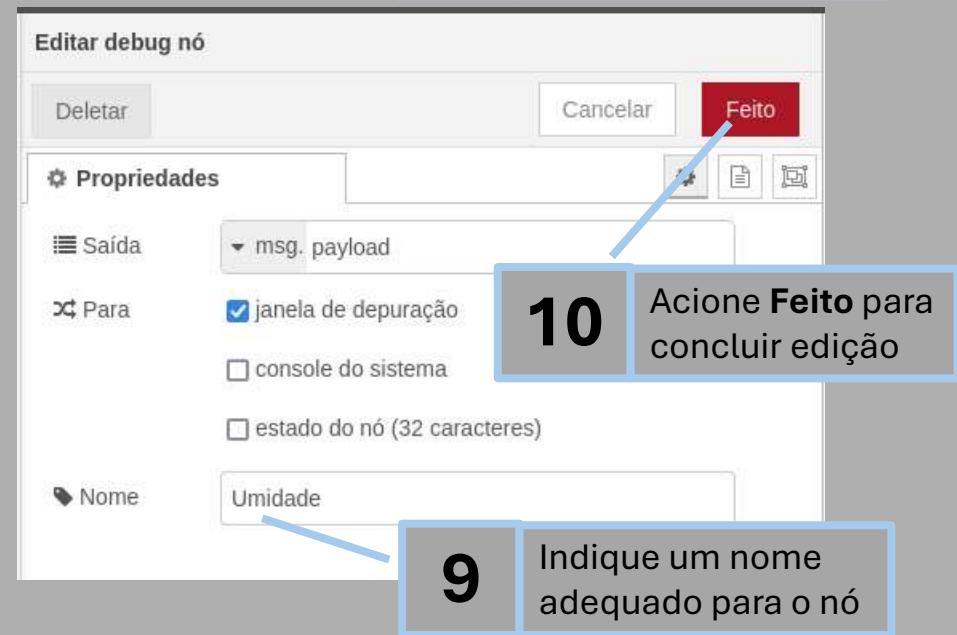
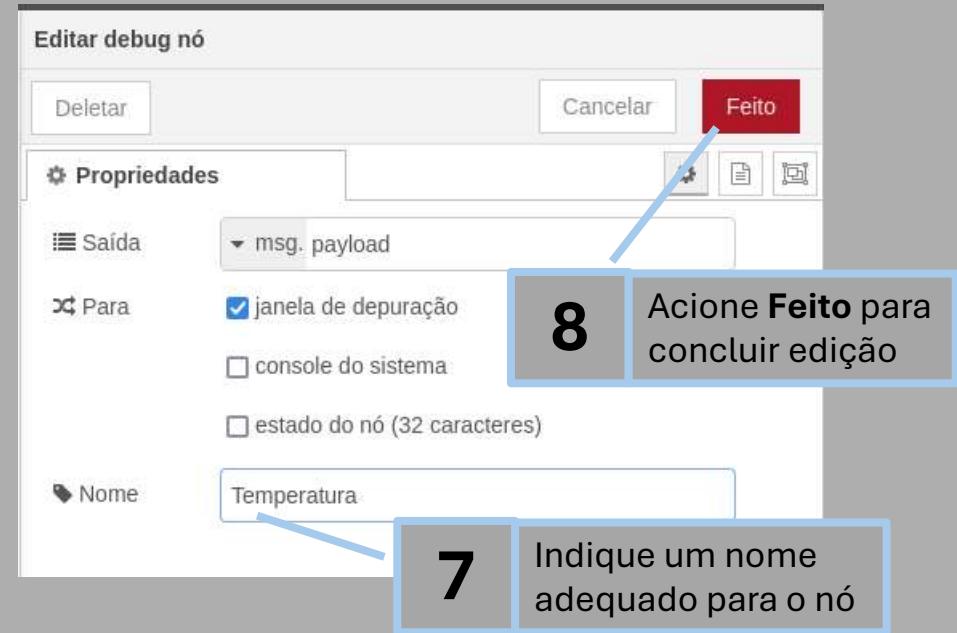
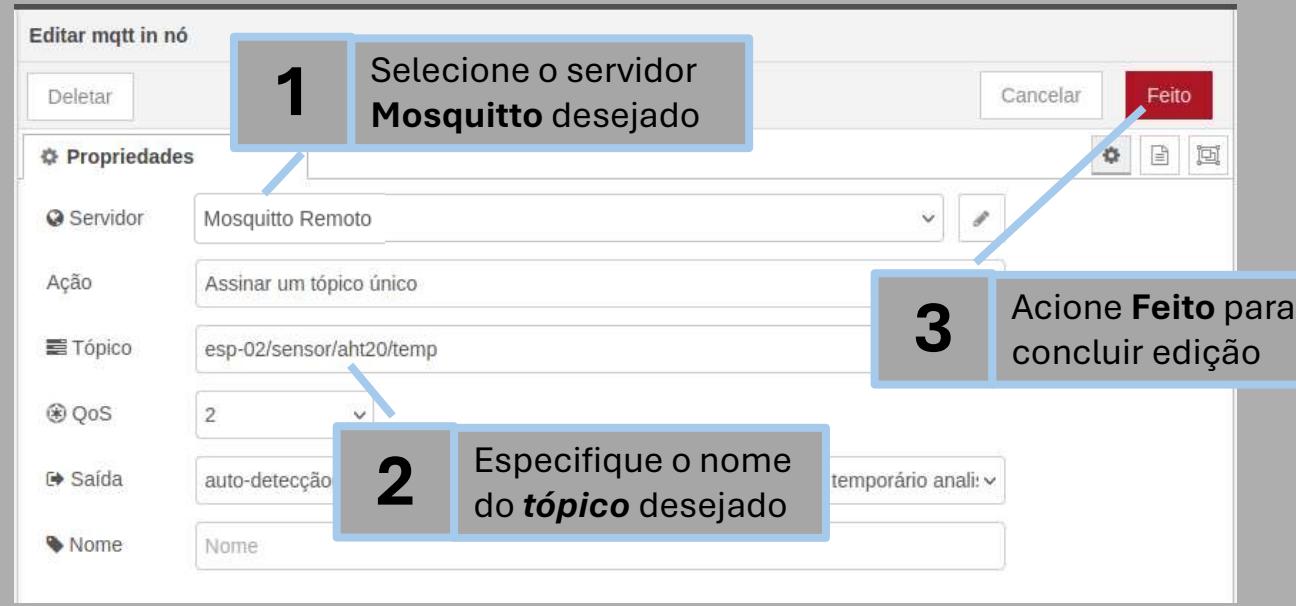
Em branco para geração automática

Mantenha-se vivo: 60

Sessão: Usar sessão limpa

Habilitar | **0** | Em todos os fluxos

(C) PET Lab/Fatec Jundiaí



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

funcção

rede

Ola! MQTT Count MQTT Data + depurar fluxo atual filtrado

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.

Temperatura

Umidade

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

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

27/06/2024 caninos@labrado... Node-RED : Flux... (C) PET Lab/Fatec Jundiaí 81

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

Editar mqtt in nó

1 Selecione o servidor **Mosquitto** desejado

Deletar	Cancelar	Feito
⚙️ Propriedades		
Servidor	Mosquitto Remoto	
Ação	Assinar um tópico único	
Tópico	esp-02/sensor/bmp280	
QoS	2	▼
Saída	auto-detecção(obje	
Nome	Nome	

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

Deletar	Cancelar	Feito
⚙️ Propriedades		
➡️ Saída		
⬇️ msg. payload		
<input checked="" type="checkbox"/> janela de depuração		
<input type="checkbox"/> console do sistema		
<input type="checkbox"/> estado do nó (32 caracteres)		
🔖 Nome		
Dados JSON		

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

10.66.29.210 - Conexão de Área de Trabalho Remota

Node-RED : Fluxo 1

localhost:1880/#flow/884d430cb0401c89

Node-RED

1 Implemente as modificações para ativar o fluxo

2 Observe a configuração e estado nos nós

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

Aqui, utilizamos um nó mqtt in para dados múltiplos publicados em um único tópico.

Dados múltiplos recebidos como um objeto JSON.

Temperatura

Umidade

Dados JSON

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"

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

17:39

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

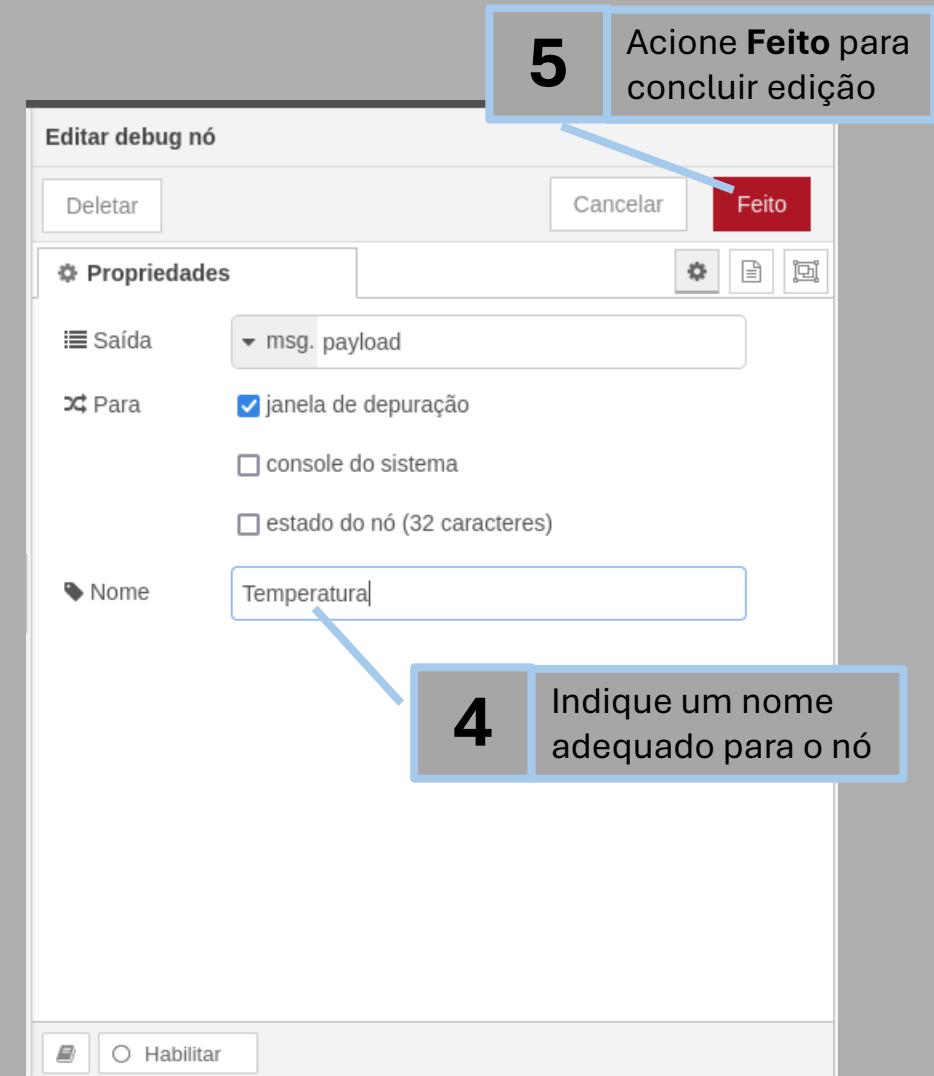
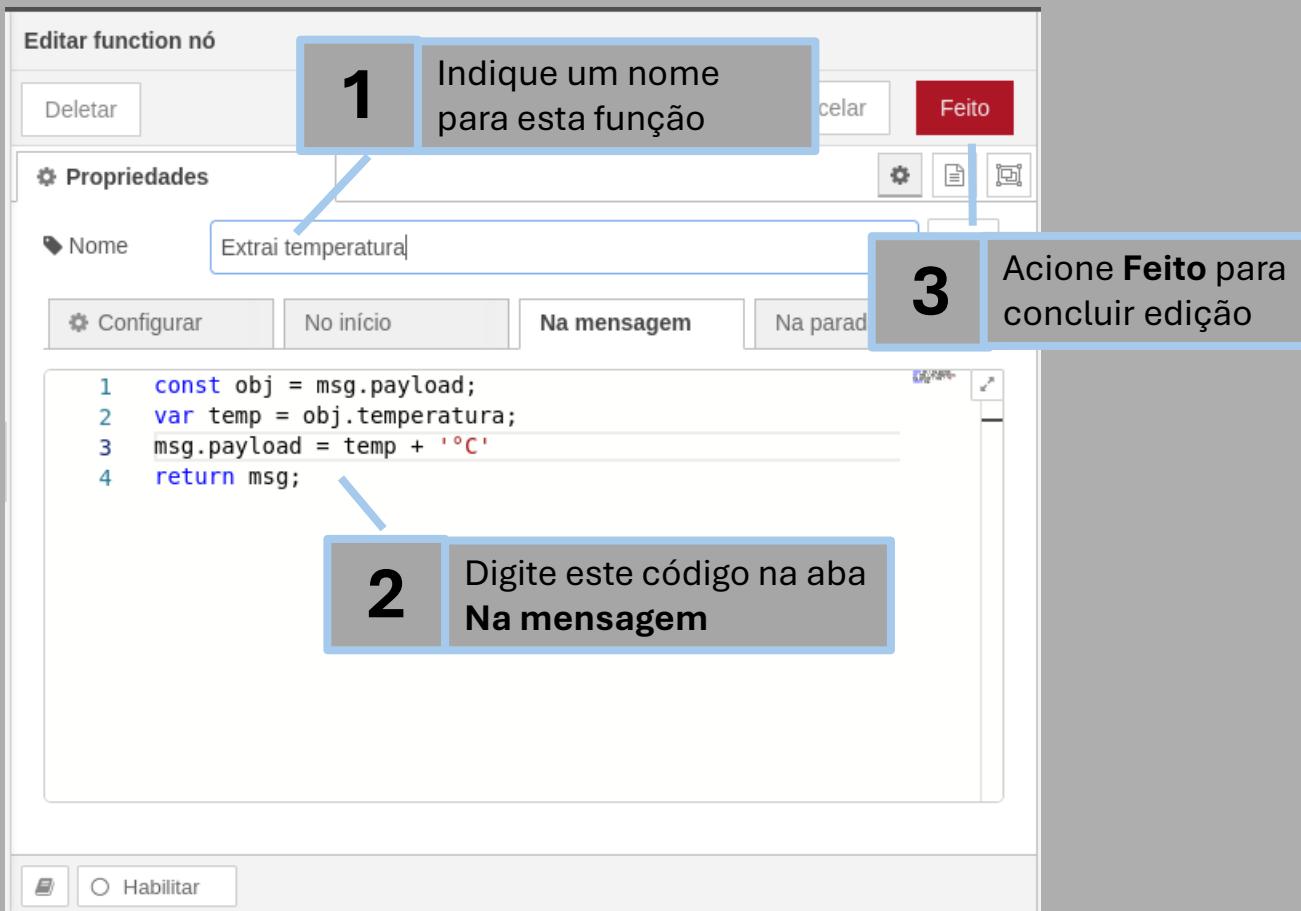
3 Adicione um nó **function** e outro nó **debug**

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"

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



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

- function
- switch

Ola!

MQTT Count

MQTT Data

depurar

fluxo atual

filtrado

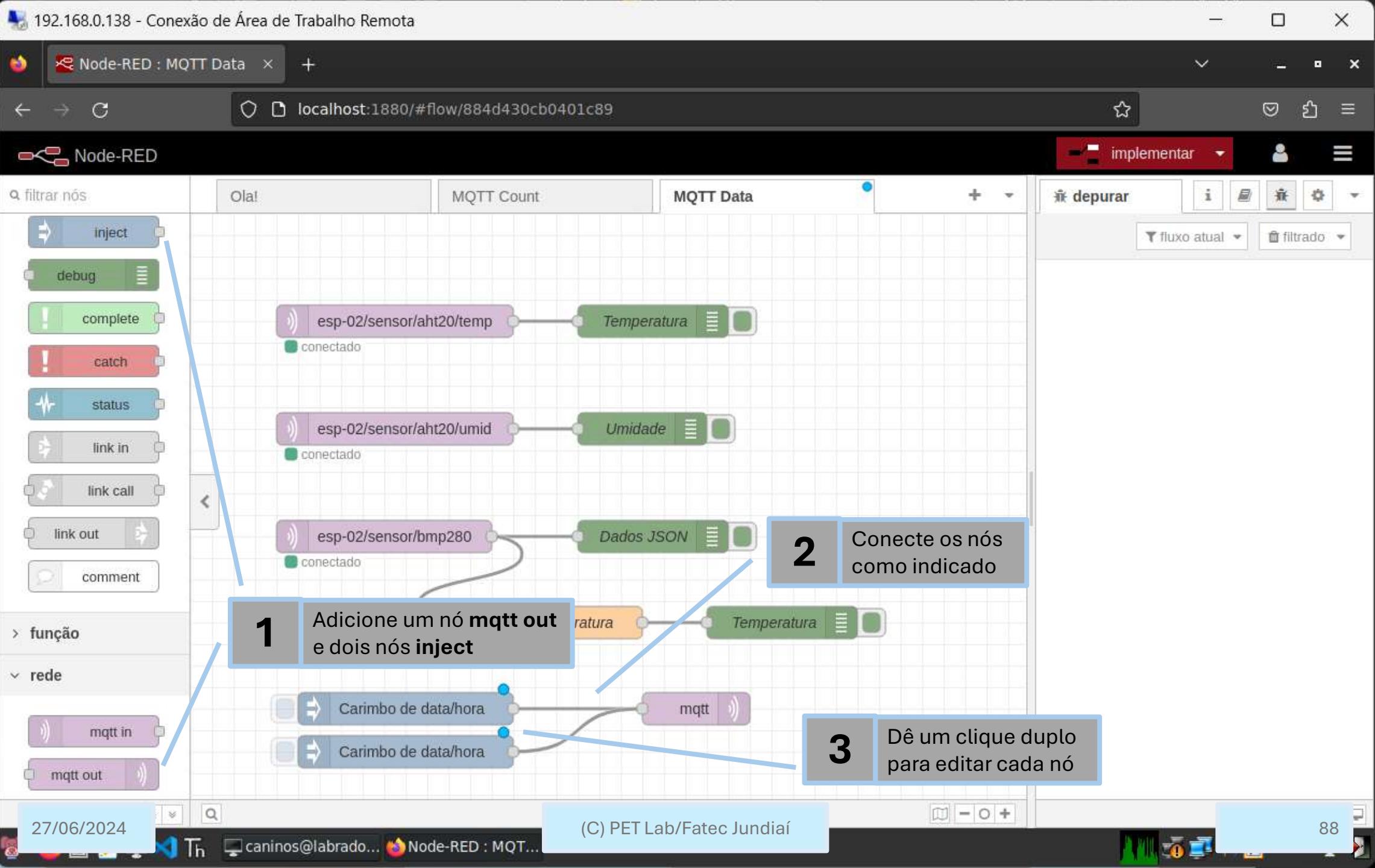
1 Implemente as modificações para ativar o fluxo

2 Observe a configuração e estado nos nós

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

```
graph LR; A[esp-02/sensor/aht20/temp] --> B[Temperatura]; C[esp-02/sensor/aht20/umid] --> D[Umidade]; D --> E[Dados JSON]; E --> F[Temperatura]; G[esp-02/sensor/bmp280] --> H[Dados JSON]; H --> I[Extrai temperatura]; I --> F;
```

20.03545
18/06/2024, 17:48:42 nó: Umidade
esp-02/sensor/aht20/umid : msg.payload : number
42.70668
18/06/2024, 17:49:12 nó: Dados JSON
esp-02/sensor/bmp280 : msg.payload : Object
▶ { pressao: "936.2573", temperatura: "24.7" }
18/06/2024, 17:49:42 nó: Temperatura
esp-02/sensor/bmp280 : msg.payload : string[6]
"24.7°C"



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**

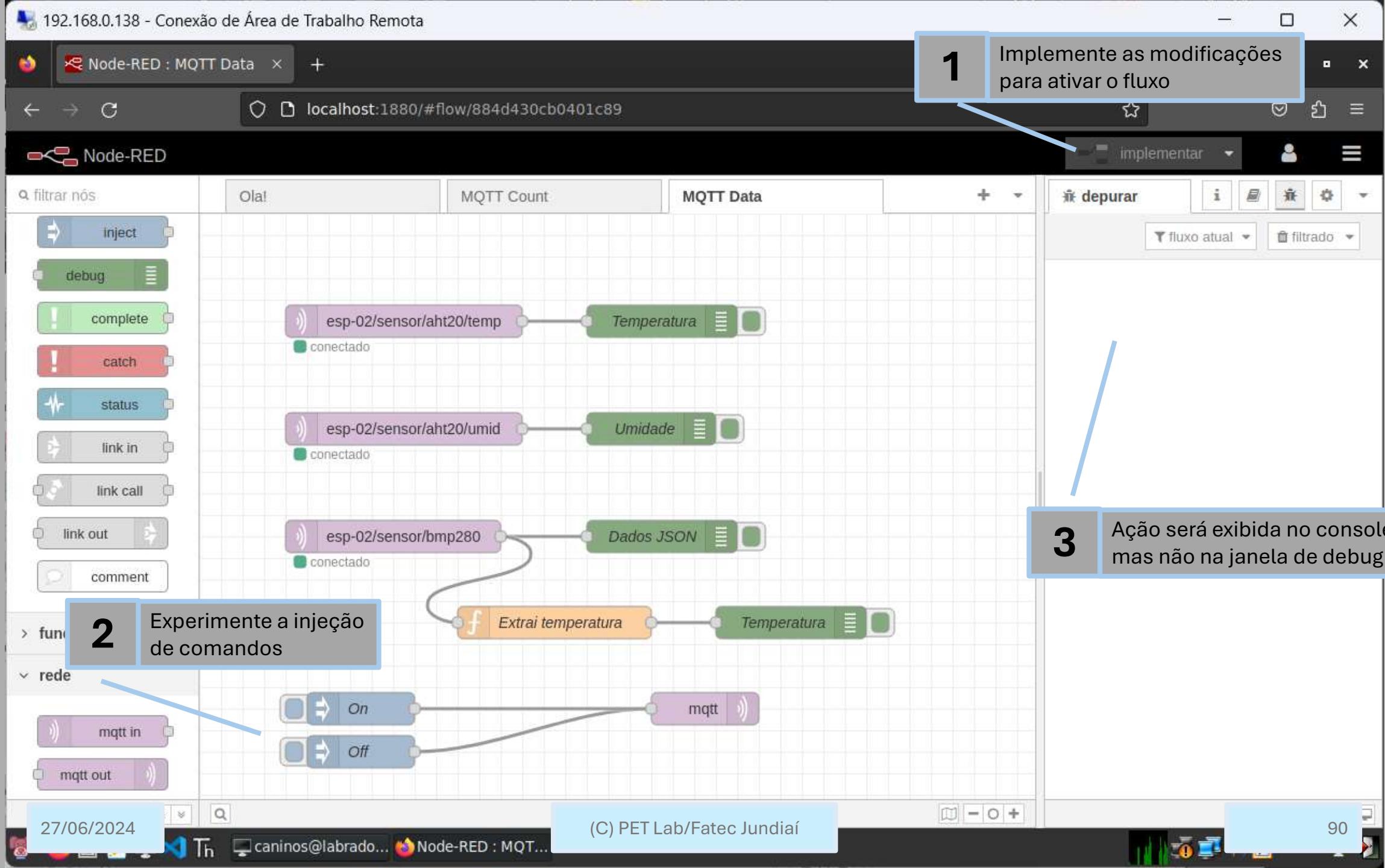
6 Selecione cadeia de caracteres e digite **desligar**

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



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

1 Acrescente um nó **function** e outro **mqtt out**

2 Conecte-os como indicado

3 Clique duplo em cada um para editá-los

```
graph TD; espTemp[esp-02/sensor/aht20/temp] --> T1[Temperatura]; espUmid[esp-02/sensor/aht20/umid] --> U1[Umidade]; espBmp[esp-02/sensor/bmp280] --> D1[Dados JSON]; T1 --> F1(function 1); F1 --> T2[Temperatura]; U1 --> F2(function Extrai temperatura); F2 --> T3[Temperatura]; On[On] --> M1[mqtt out]; Off[Off] --> M1;
```

function

switch

change

range

template

delay

trigger

exec

filter

rede

mqtt.in

mqtt out

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

91

3 Acione **Feito** para concluir a edição

1 Ajuste o nome do nó **function**

2 Digite este código que avalia o **payload**, modificando-o

Se a temperatura é maior que 24°C, o **msg.payload** é “ligar”.
Para valores de temperatura iguais ou menores que este limiar, o **msg.payload** é “desligar”.

7 Acione **Feito** para concluir a edição

4 Selecione o *broker* adequado

5 Especifique o **tópico** desejado

6 Indique o nível de serviço requerido

192.168.0.138 - Conexão de Área de Trabalho Remota

Node-RED : MQTT Data

localhost:1880/#flow/884d430cb0401c89

Node-RED

filtrar nós

função

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

rede

- mqtt.in
- mqtt out

HTTP & Node-RED

MQTT Data

depurar

fluxo atual

filtrado

2 Aguarde (ou simule) a recepção de dados neste tópico

1 Implemente as alterações

3 Temperatura, publicada no tópico, controlada remotamente

The screenshot shows a Node-RED application window with the following components:

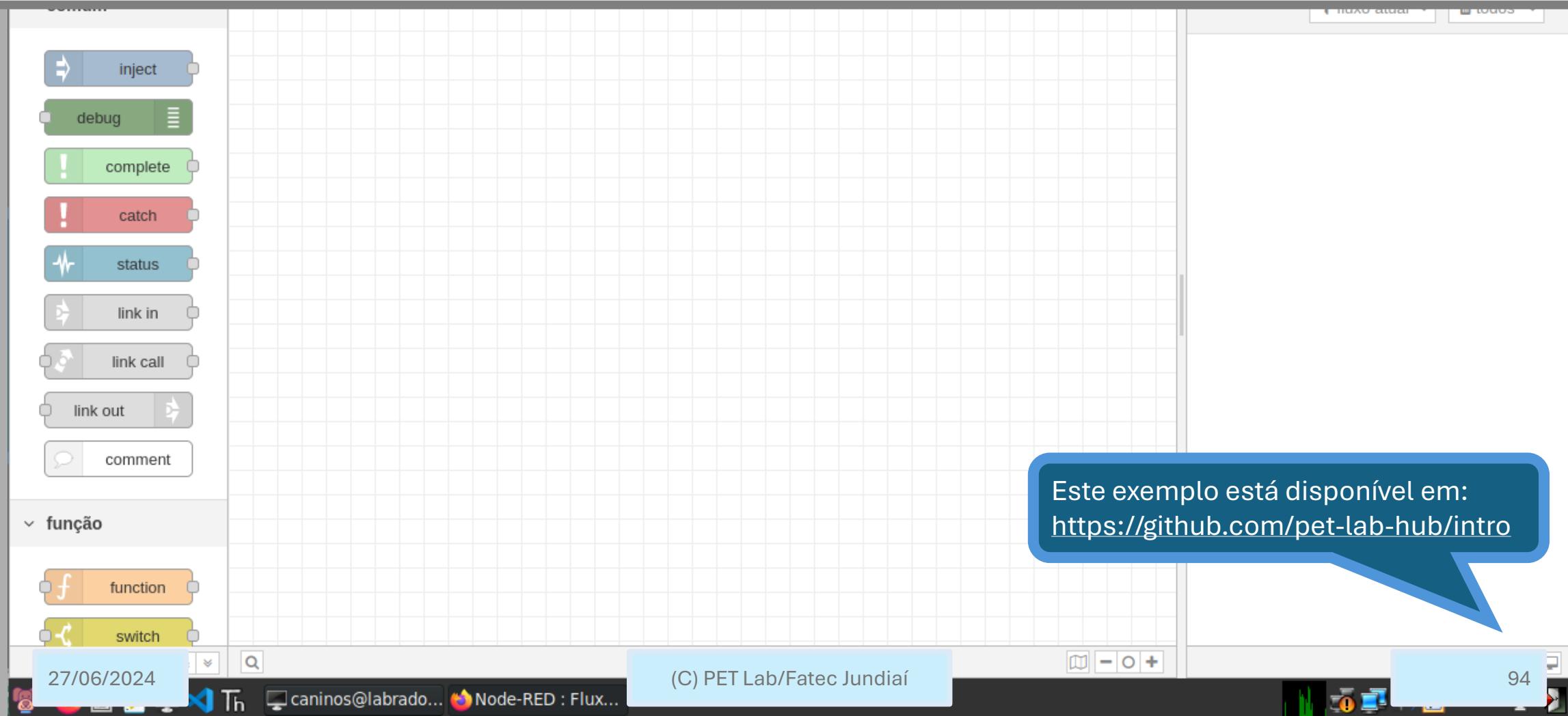
- Nodes:** A "function" node labeled "esp-02/sensor/aht20/temp" receives data from an "esp-02" sensor and sends it to a "Temperatura" output node. A "Controle de Temperatura" function node receives data from the "Temperatura" node and sends it to an "esp-01/atuador/relay" output node. Another "function" node labeled "esp-02/sensor/aht20/umid" receives data from the same sensor and sends it to a "Umidade" output node. A third "function" node labeled "esp-02/sensor/bmp280" receives data from the sensor and sends it to a "Dados JSON" output node. This "Dados JSON" node then connects to another "function" node labeled "Extrai temperatura", which in turn connects to the "Temperatura" output node.
- Control:** Two "On/Off" switch nodes, "On" and "Off", receive inputs from the "Temperatura" and "Umidade" nodes respectively. These switches are connected to an "mqtt" output node, which publishes data to a topic.

The "MQTT Data" tab shows the following log entries:

- 20/06/2024, 22:47:07 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
23.5
- 20/06/2024, 22:47:17 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
24.1
- 20/06/2024, 22:47:25 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
24.7
- 20/06/2024, 22:47:36 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
23.9

The status bar at the bottom displays the date (27/06/2024), the user (caninos@labrado...), and the title (Node-RED : MQTT...).

Exemplo 5 [Dashboards]

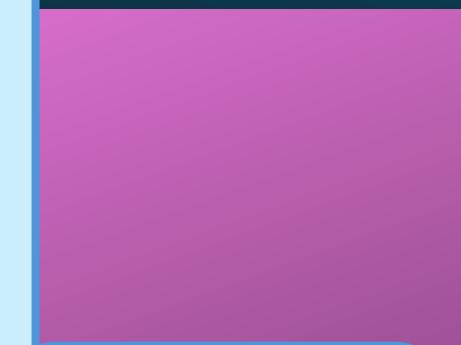


dashboard

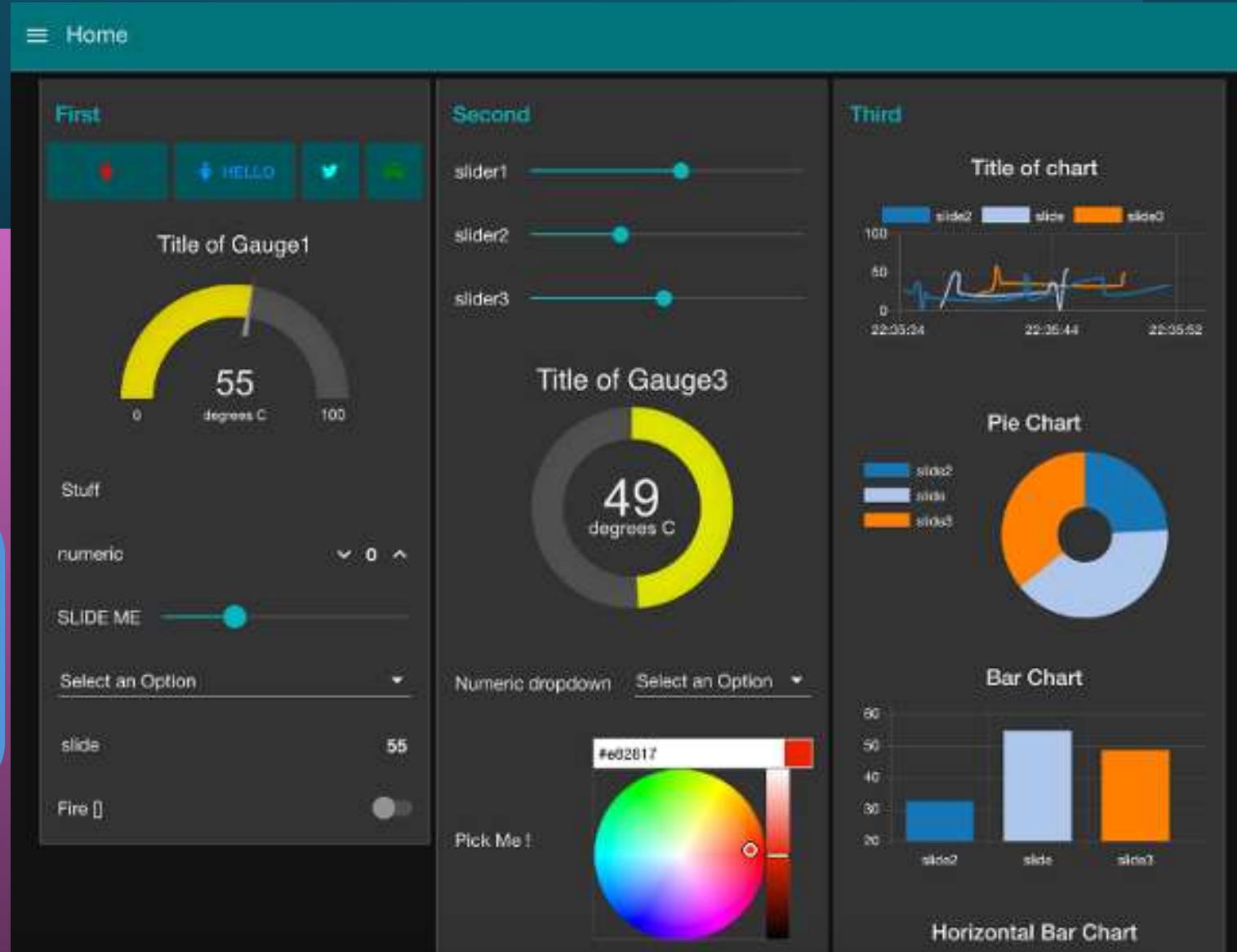
Um *dashboard* ou **painel (de visualização) de dados** é uma ferramenta visual que exibe informações importantes, consolidadas, para fácil interpretação.

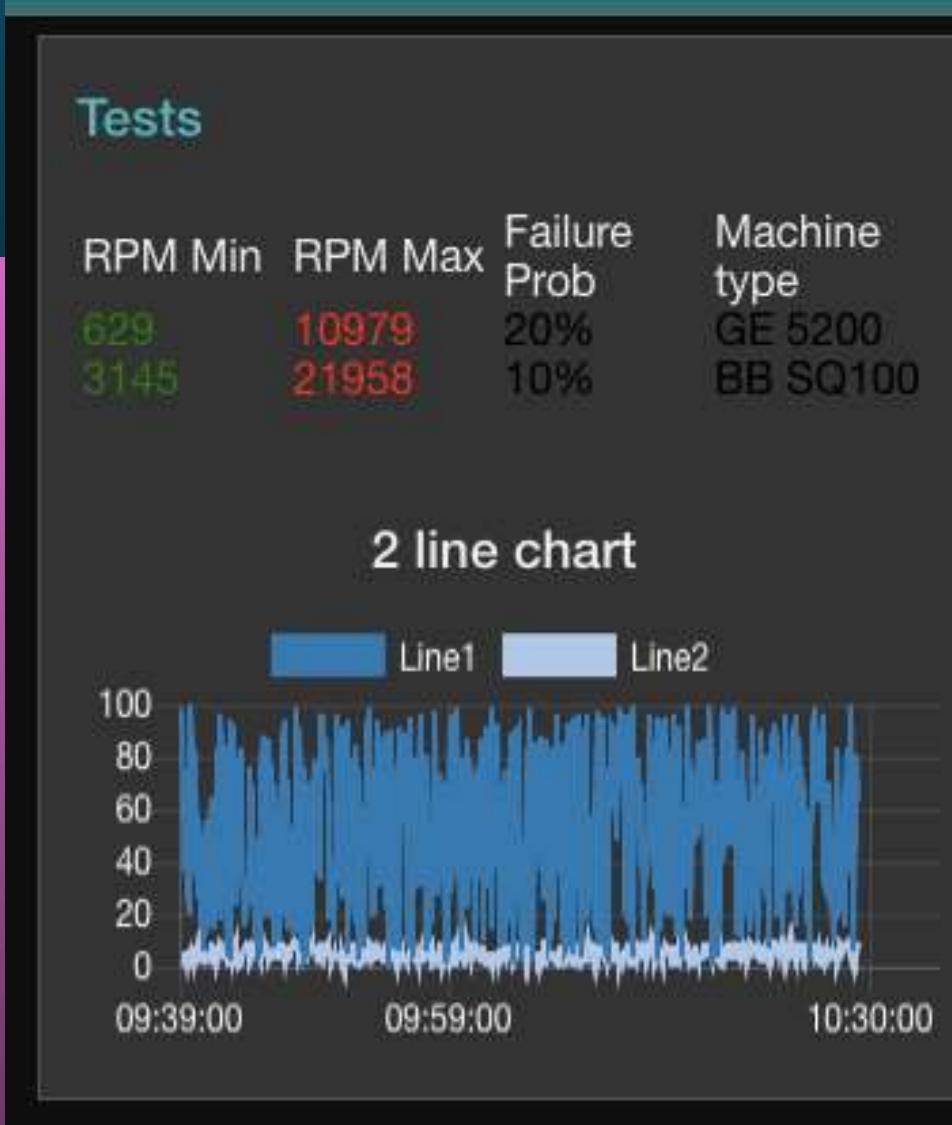
Geralmente apresenta dados e métricas em tempo real, podendo incluir gráficos, tabelas, indicadores de desempenho (**KPIs**), mapas e outros elementos para permitir aos usuários monitorar o comportamento de processos ou atividades em um único lugar.

São amplamente utilizados em diversos contextos (indústria, negócios, saúde etc.).



Cada instância do Node-RED pode exibir um dashboard composto de vários grupos de componentes.





O dashboard é atualizado em tempo real pelos fluxos em execução.

<https://noderedguide.com/tutorial-node-red-dashboards-creating-your-own-ui-widget-ii-using-external-charts/>

≡ PWS Example

PWS Queries

Enter StationID

KNJRIDGE24

CURRENT PWS OBSERVATIONS

Historical Date

13/03/2020

Historical Date AvgTemp

58

F

0

100

Gauge

mostrador analógico

7 DAY PWS HOURLY SUMMARY

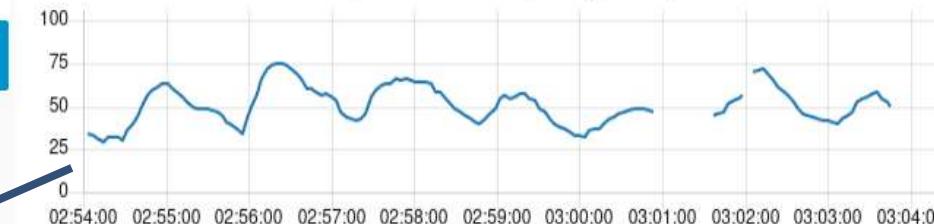
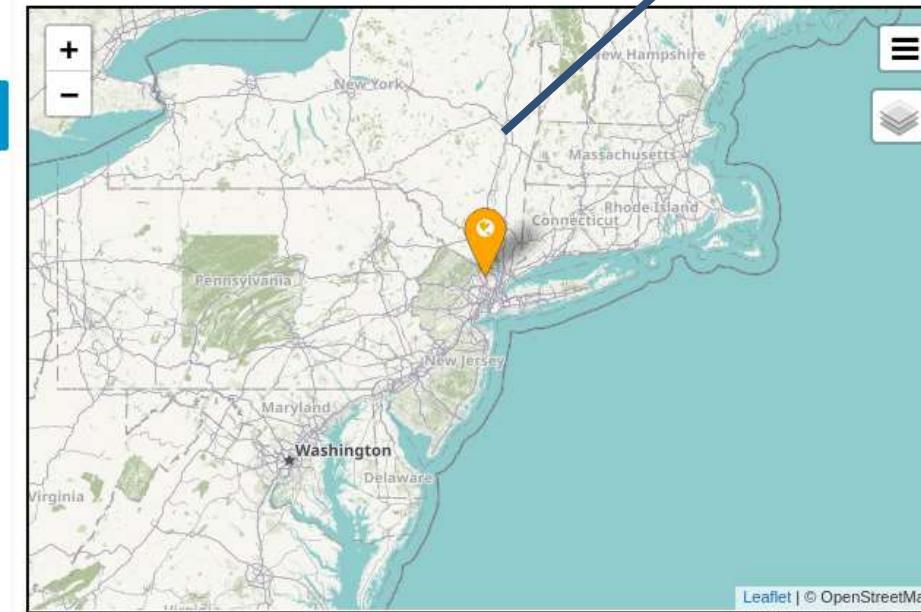


Chart gráfico

Observations	Results
stationID	KNJRIDGE24
obsTimeUtc	2020-03-15T18:19:00Z
obsTimeLocal	2020-03-15 14:19:00
neighborhood	Southeast Ridgewood
softwareType	
country	US

Text caixa de texto

PWS Results



Map mapa

Forecast

List lista

TODAY FORECAST



Sunday : Abundant sunshine. Highs in the low 50s and lows in the upper 20s.



Monday : Mostly sunny. Highs in the mid 40s and lows in the upper 30s.



Tuesday : Morning showers. Highs in the upper 50s and lows in the mid 30s.



Wednesday : More sun than clouds. Highs in the low 50s and lows in the low 40s.

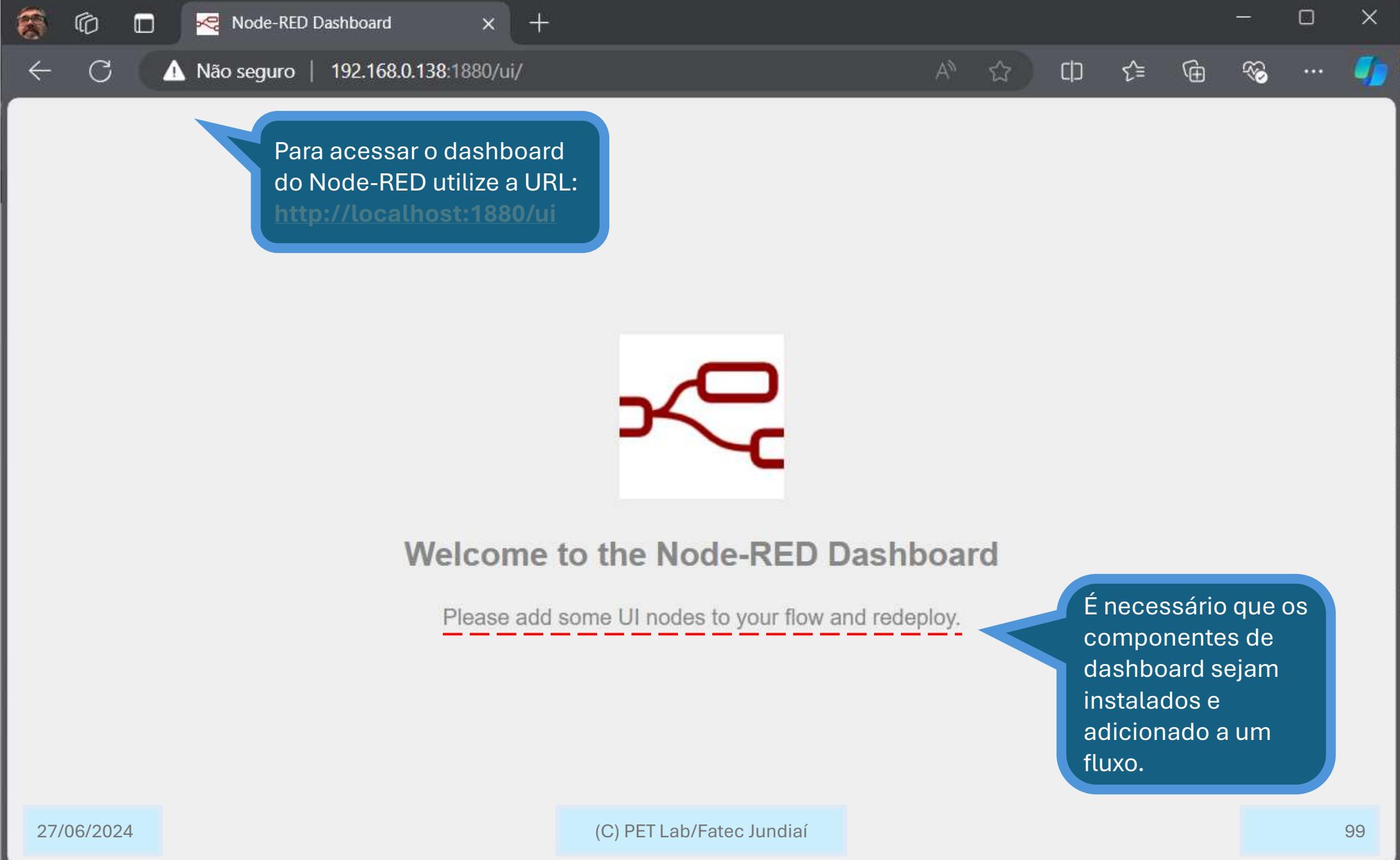


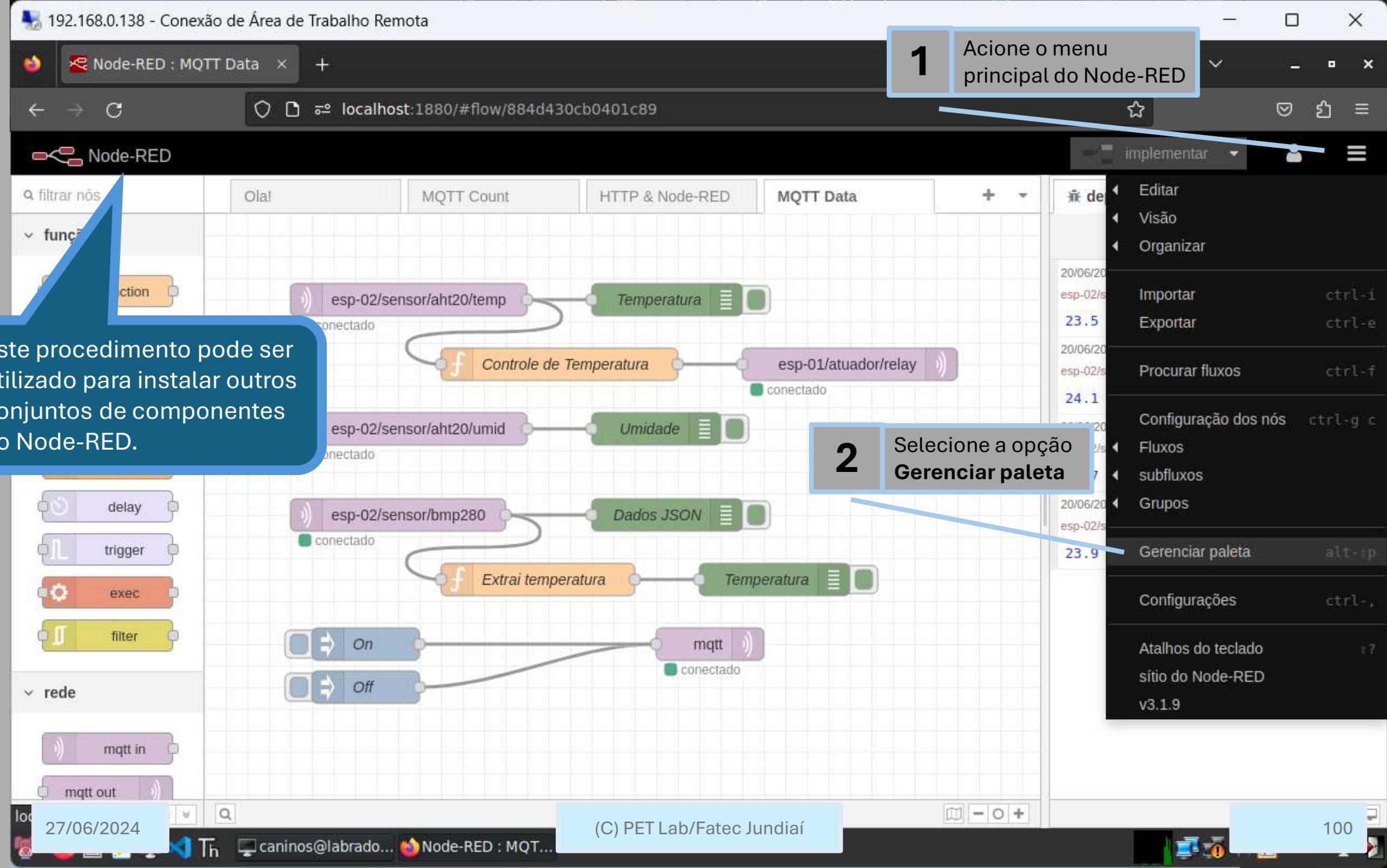
Thursday : Cloudy, periods of rain. Highs in the upper 50s and lows in the low 50s.

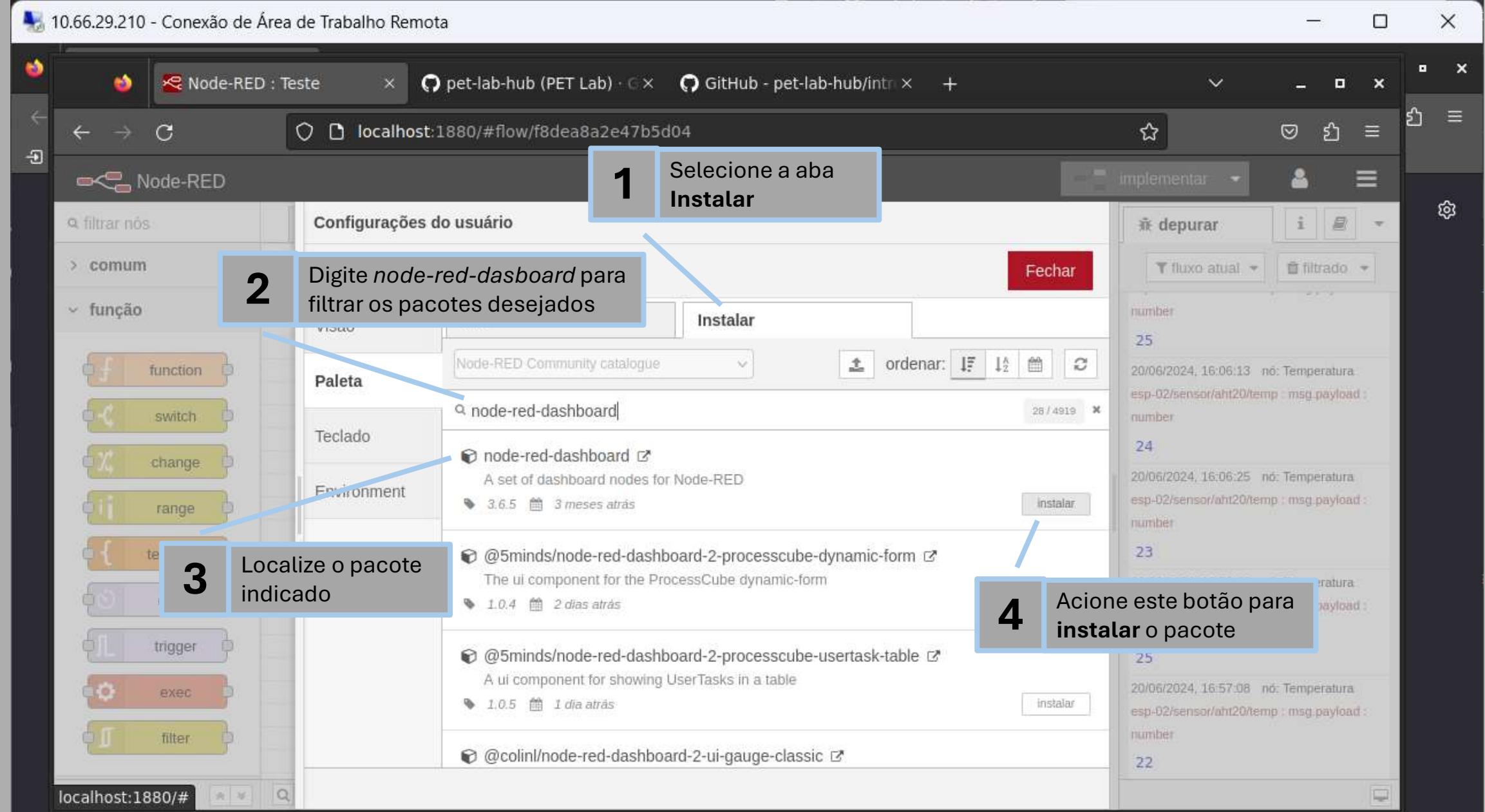


Friday : Showers. Highs in the mid 70s and lows in the mid 40s.

Table tabela







Node-RED : Teste

localhost:1880/#flow/f8dea8a2e47b5d04

Instalando 'node-red-dashboard'

Antes de instalar, leia a documentação do nó. Alguns nós têm dependências que não podem ser resolvidas automaticamente e podem exigir a reinicialização do Node-RED.

Cancelar Abrir informação do nó Instalar

Node-RED Community catalogue

ordenar: ▲ ▼

node-red-dashboard

1 Confirme a instalação

node-red-dashboard A set of dashboard nodes for Node-RED 3.6.5 3 meses atrás

@5minds/node-red-dashboard-2-processcube-dynamic-form The ui component for the ProcessCube dynamic-form 1.0.4 2 dias atrás

@5minds/node-red-dashboard-2-processcube-usertask-table A ui component for showing UserTasks in a table 1.0.5 1 dia atrás

@colinl/node-red-dashboard-2-ui-gauge-classic

instalar

depurar

fluxo atual

filtrado

number

25

20/06/2024, 16:06:13 nº: Temperatura esp-02/sensorfahrt20/bmp msg payload number

24

20/06/2024, 16:00:25 nº: Temperatura esp-02/sensorfahrt20/temp msg payload number

23

20/06/2024, 16:56:48 nº: Temperatura esp-02/sensorfahrt20/bmp msg payload number

25

20/06/2024, 16:57:08 nº: Temperatura esp-02/sensorfahrt20/temp msg payload number

22

1 Confirme a instalação

Node-RED : Teste

localhost:1880/#flow/f8dea8a2e47b5d04

Configurações do usuário

Visão

Paleta

1 Aguarde a instalação

Instalar

catalogue

ordenar:

node-red-dashboard

Registro de visão

@5minds/node-red-dashboard-2-processcube-dynamic-form

The ui component for the ProcessCube dynamic-form

1.0.4 2 dias atrás

instalar

@5minds/node-red-dashboard-2-processcube-usertask-table

A ui component for showing UserTasks in a table

1.0.5 1 dia atrás

instalar

@colinl/node-red-dashboard-2-ui-gauge-classic

number

25

20/06/2024, 16:06:13 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number

24

20/06/2024, 16:06:25 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number

23

20/06/2024, 16:56:48 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number

25

20/06/2024, 16:57:08 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number

22

27/06/2024

caninos@labrado... Mozilla Firefox

(C) PET Lab/Fatec Jundiaí

103

Node-RED : Teste

localhost:1880/#flow/f8dea8a2e47b5d04

Configurações do usuário

Visão Nós Instalar

Paleta Node-RED Community catalogue ordenar: 1 2 3 4 5

Teclado

Environment

node-red-dashboard
A set of dashboard nodes for Node-RED
3.6.5 3 meses atrás instalado

@5minds/node-red-dashboard-2-processcube-dynamic-form
The ui component for the ProcessCube dynamic-form
1.0.4 2 dias atrás instalar

@5minds/node-red-dashboard-2-processcube-usertask-table
A ui component for showing UserTasks in a table
1.0.5 1 dia atrás instalar

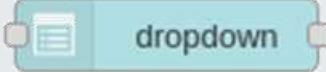
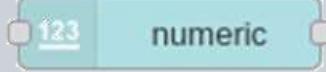
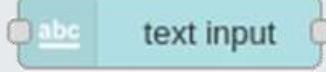
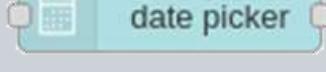
@colinl/node-red-dashboard-2-ui-gauge-classic

1 Aguarde a instalação

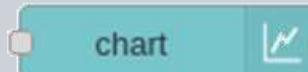
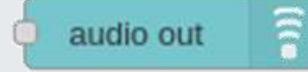
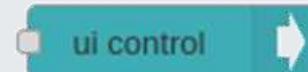
depurar fluxo atual filtrado

number 25 20/06/2024, 16:06:13 nó: Temperatura esp-02/sensor/aht20/temp msg.payload: number 24 20/06/2024, 16:06:25 nó: Temperatura esp-02/sensor/aht20/temp msg.payload: number 23 20/06/2024, 16:56:48 nó: Temperatura esp-02/sensor/aht20/temp msg.payload: number 25 20/06/2024, 16:57:08 nó: Temperatura esp-02/sensor/aht20/temp msg.payload: number 22

Node-RED::referência

Nós de Entrada	Descrição
 button	Botão com aparência configurável para o acionamento de alguma função.
 dropdown	Permite a definição de uma lista selecionável de pares <i>rótulo:valor</i> com atribuição de um array objetos <code>[{"rótulo": valor}, ...]</code> na propriedade msg.options .
 switch	Chave <i>on/off</i> que pode ter dois ícones e/ou cores conforme seu estado.
 slider	Ajuste deslizante para valor numérico, com limites e passo ajustável.
 numeric	Campo de entrada numérico, com botões de controle <i>Up/Down</i> .
 text input	Campo de entrada de texto, com rótulo opcional e que suporta entrada de senha, e-mail e modos de cor.
 date picker	Componente para exibição de data (um objeto Date , cujo formato pode ser definido na aba Site com uso de formatação <i>moment.js</i>).
 form	Componente composto de subcomponentes, cujos valores podem ser submetidos como uma mensagem simples.

Node-RED::referência

Nós de Saída	Descrição
	Componente para exibição de texto fixo (rótulo), com conteúdo e layout configuráveis.
	Mostrador analógico com modos <i>standard</i> , <i>donut</i> , <i>compass</i> e <i>wave</i> com várias opções de configuração.
	Componente para exibição de gráficos de linha, barra ou torta com várias opções de configuração.
	Componente que permite tocar áudio (<i>wav</i> ou <i>mp3</i>) ou enviar <i>Text to Speech</i> (TTS) para o cliente.
	Permite criar alertas visíveis para o usuário no console, como um <i>toast popup</i> ou janela dispensável.
	[configuração] Possibilita algum controle dinâmico do dashboard por meio de mensagens, tal como a seleção ou habilitação de abas/guias.
	[configuração] Permite a criação de outros componentes com uso de HTML, CSS, JavaScript e Angular.

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

1 Acrescente três nós gauge e um switch

2 Conecte-os como indicado

3 Clique duplo em cada um para editá-los

slider

numeric

text input

date picker

colour picker

form

text abc

gauge

chart

audio out

notification

ui control

template

esp-02/sensor/aht20/temp

Temperatura

Controle de Temperatura

esp-01/atuador/relay

esp-02/sensor/aht20/umid

Umidade

Dados JSON

esp-02/sensor/bmp280

Extrai temperatura

On

Off

atuador/led

switch

2024, 19:00:14 nó: Umidade
esp-02/sensor/aht20/umid : msg.payload : number
06

20/06/2024, 19:01:14 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
17

20/06/2024, 19:01:14 nó: Umidade
esp-02/sensor/aht20/umid : msg.payload : number
27.66

20/06/2024, 19:02:14 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
39.54

14 nó: Umidade
esp-02/sensor/aht20/umid : msg.payload : number
19.27

20/06/2024, 19:03:14 nó: Temperatura
esp-02/sensor/aht20/temp : msg.payload : number
33.29

27/06/2024

(C) PET Lab/Fatec Jundiaí

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

107

```
graph LR; A[esp-02/sensor/aht20/temp] --> B[Temperatura]; A --> C[gauge]; B --> D[Controle de Temperatura]; C --> D; D --> E[esp-01/atuador/relay]; A --> F[esp-02/sensor/aht20/umid]; F --> G[Umidade]; G --> H[gauge]; F --> I[Dados JSON]; I --> J[gauge]; I --> K[Extrai temperatura]; K --> L[Temperatura]; K --> M[switch]; M -- On --> N[On]; M -- Off --> O[Off]; N --> P[atuador/led]; O --> P;
```

1 Selecione Adicionar novo dashboard group...

2 Acione o botão Editar

3 Defina as propriedades do novo grupo

4 Acione o botão Adicionar

5 Defina as características do gauge

6 Acione o botão Feito

Os componentes de um *dashboard* precisam ser colocados em grupos (*groups*) e abas (*tabs*). Podem existir diversos grupos e abas para organizar os componentes na UI.

Editar gauge nó

1 Selecione o grupo já criado

2 Defina as características do gauge

3 Acione o botão Feito

4 Selecione o grupo já criado

5 Defina as características do switch

6 Acione o botão Feito

Deletar

Propriedades

Group: [Home] AHT20

Size: auto

Type: Gauge

Label: Umidade Relativa

Value format: {{value}}

Units: %

Range: min 0 max 100

Colour gradient: Red, Yellow, Green

Sectors: 0 ... 20 ... 60 ... 100

Habilitar

Editar switch nó

4 Selecione o grupo já criado

5 Defina as características do switch

6 Acione o botão Feito

Deletar

Propriedades

Group: [Home] AHT20

Size: auto

Label: Led

Tooltip: Controle do Led

Icon: Default

→ Pass through msg if payload matches valid state:

When clicked, send:

On Payload:

Off Payload:

Habilitar

1 Selecione Adicionar novo dashboard group...

2 Acione o botão Editar

3 Defina as propriedades do novo grupo

4 Acione o botão Adicionar

5 Defina as características do gauge

6 Acione o botão Feito

1

Implemente as alterações

Node-RED

Implementado com sucesso

implementar

filtrar nós

slider

numeric

text input

date picker

colour picker

form

text

gauge

chart

audio out

notification

ui control

</> template

Olá!

MQTT Count

HTTP & Node-RED

MQTT Data

Sensor Virt

depurar

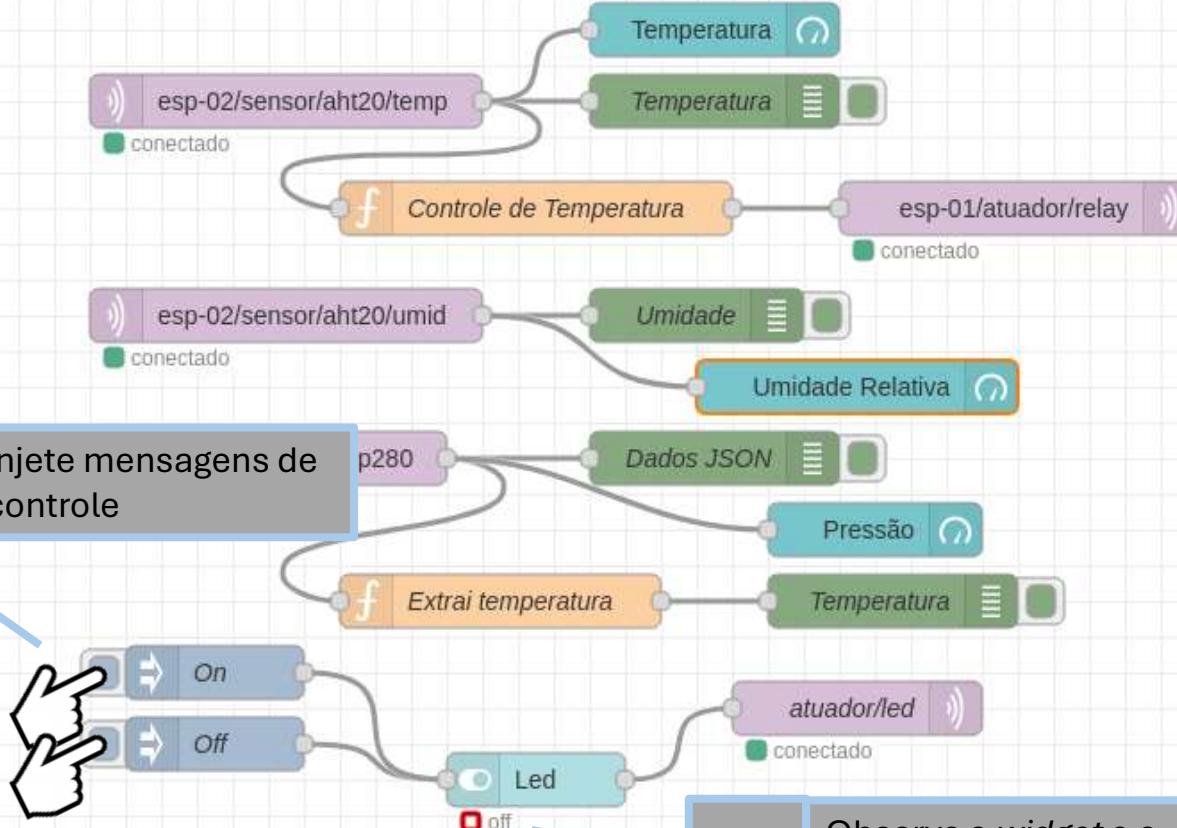
i

e

s

g

v



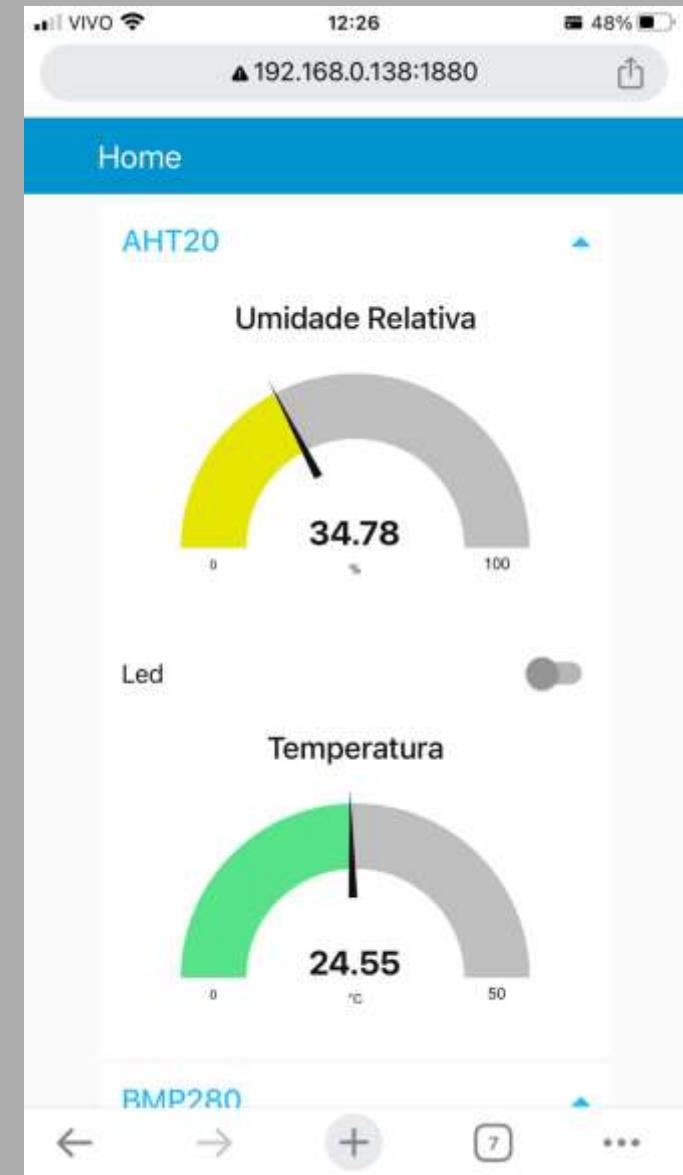
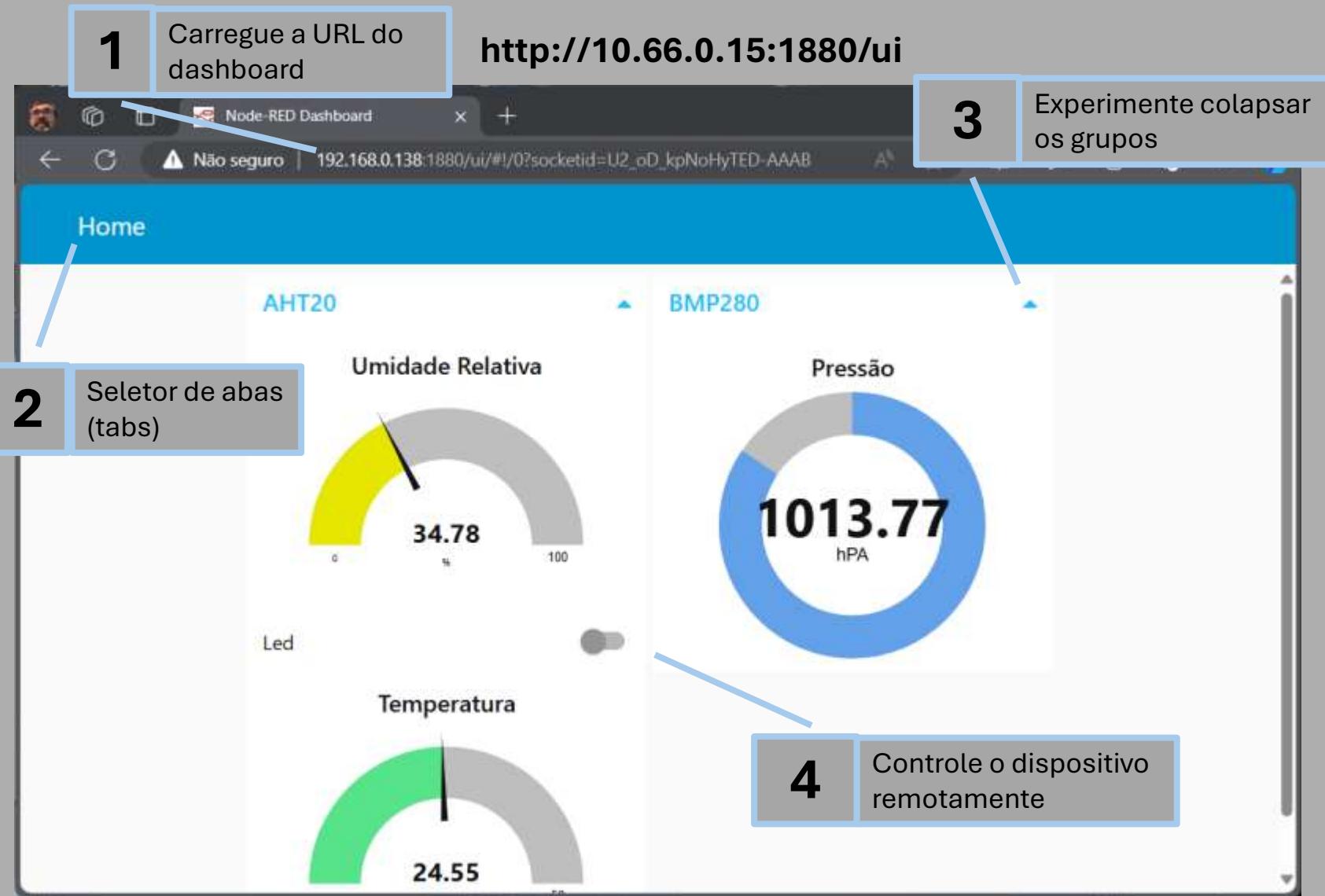
2

Injete mensagens de
controle



3

Observe o *widget* e o
estado do dispositivo



Próximos Passos

Próximos Passos

Node-RED

- Aprofundamento e outras aplicações da ferramenta, tais como:
 - 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 para integração com hardware.
 - Integração de HW e SW com **FastIoT** (plataforma de prototipagem rápida da Fatec Jundiaí).

Para saber mais

- [Instalação] <https://nodered.org/docs/getting-started/installation>
- [Introdução ao Node-RED]
<http://steves-internet-guide.com/node-red-overview/>
- [Tutoriais] <https://cookbook.nodered.org>
- [Criação de Nós] <https://nodered.org/docs/creating-nodes/>
- [Guia de Programação] <http://noderedguide.com>
- [API] <https://nodered.org/docs/api>

obrigado

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