

Automação Industrial

Node-RED

O que é o Node-RED e qual a sua aplicação?

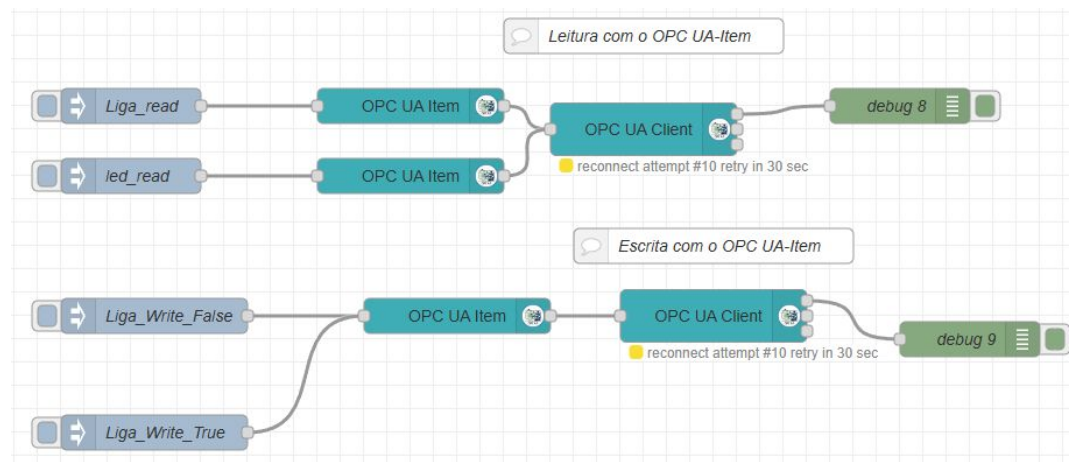
Prof. Miguel Bozer da Silva

O que é o Node-RED?

O Node-RED é uma ferramenta de **programação visual** (low code) com foco em aplicações de **integração de APIs para serviços online** e para Internet das Coisas (IoT).

Ele é baseado em JavaScript e roda em ambiente Node.js

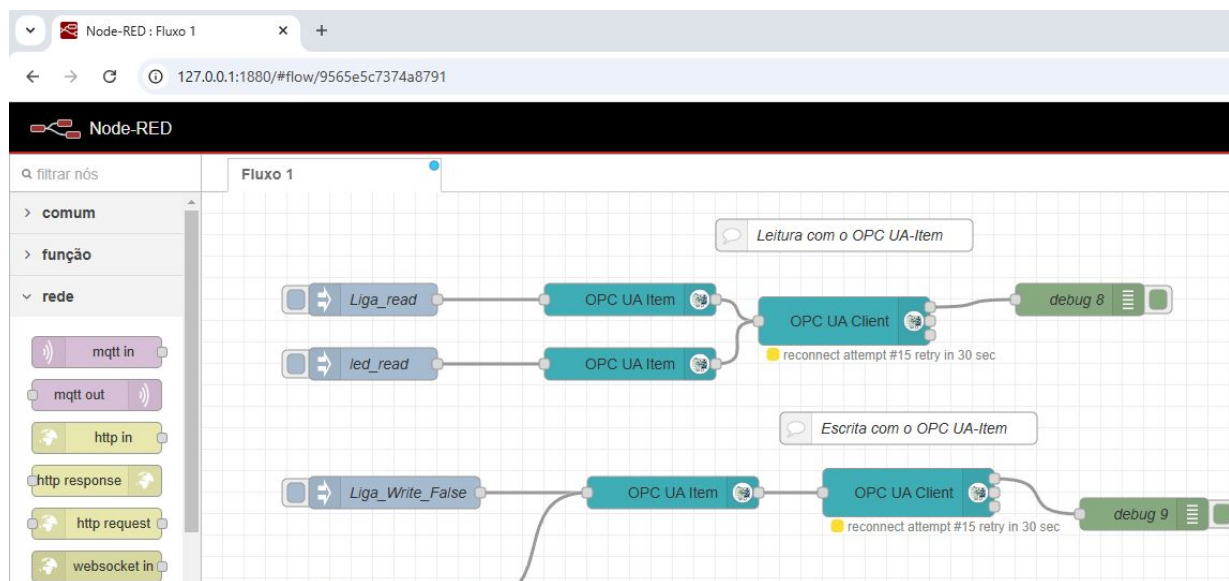
Os “programas” feitos em Node-RED são chamados de **fluxos** (**flows**). Eles são salvos em formato JSON.



Onde eu programo em Node-RED?

Você pode programar em Node-RED em um ambiente em nuvem, remoto ou no ambiente instalado na sua própria máquina local.

Em ambos os casos, a interface do ambiente de programação funciona como um servidor local, que é acessado pelo navegador (browser).



Onde eu programo em Node-RED?

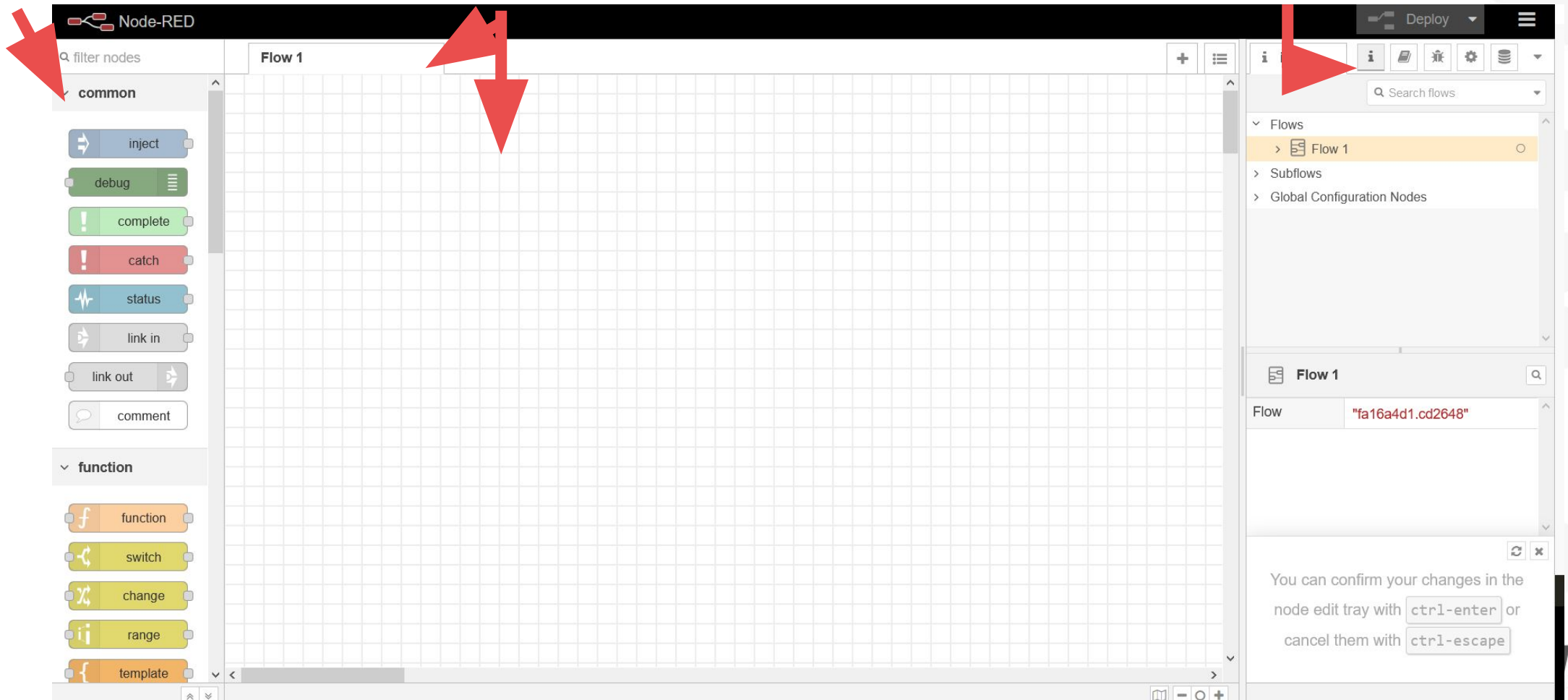
Nas aulas nós vamos usar o desenvolvimento local!

Como eu programo em Node-RED?

Aba de nós

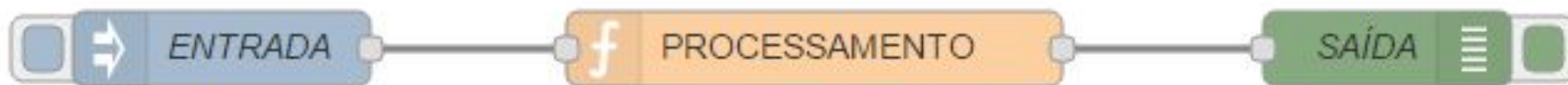
Aba de desenvolvimento do fluxo

Aba de informações



Como eu programo em Node-RED?

Tipos de nós: existem basicamente três tipos de nós, os de **entrada**, de **processamento** e de **saída**.



Mensagens: no Node-RED chamamos um programa de fluxo (Flow). Este fluxo é criado através de nós que criam, recebem e processam, **mensagens**. Cada mensagem tem uma carga útil (Payload) que pode assumir diferentes valores e tipos.

Como eu programo em Node-RED?

common

inject

debug

complete

catch

status

link in

link out

comment

function

function

switch

change

range

template

delay

trigger

exec

rbe

network

mqtt in

mqtt out

http in

http response

http request

websocket in

websocket out

tcp in

tcp out

tcp request

udp in

udp out

sequence

split

join

sort

batch

storage

file

file in

watch

tail

parser

1,2 csv

<> html

<> json

<> xml

Y yaml

Instalando o Node-RED

Node-RED no meu computador

Prof. Miguel Bozer da Silva

Instalando o NodeJS

Download Node.js®

Get Node.js® v22.12.0 (LTS) for Windows using Docker with npm

```
1 # Docker has specific installation instructions for each operating system.
2 # Please refer to the official documentation at https://docker.com/get-started/
3
4 # Pull the Node.js Docker image:
5 docker pull node:22-alpine
6
7 # Create a Node.js container and start a Shell session:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.12.0".
12
13 # Verify npm version:
14 npm -v # Should print "10.9.0".
```

PowerShell Copy to clipboard

Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)

Or get a prebuilt Node.js® for Windows running a x64 architecture.

Windows Installer (.msi) Standalone Binary (.zip)

Red arrows point from the annotations to the OS dropdown, the architecture dropdown, and the Windows Installer button.

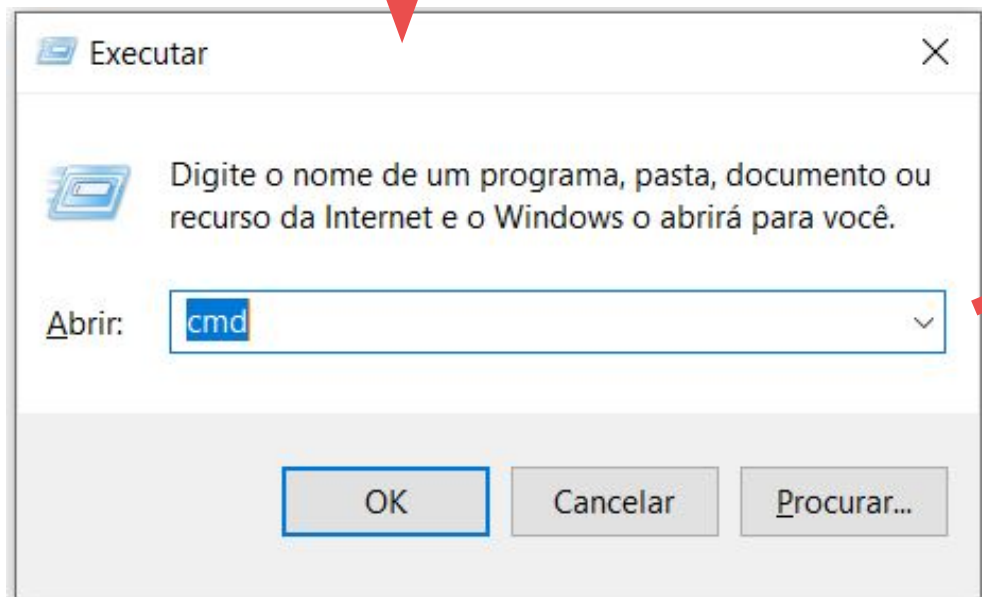
Selecionar o sistema operacional

Selecionar x64 ou x32 (de acordo com seu computador)

Baixar o instalador

Verificando a instalação do Node JS

Tecla Windows+R



Instalando o Node-RED

Instalando:

```
D:\node-red
```

```
npm install -g --unsafe-perm node-red
```

Executando:

```
D:\node-red
```

```
node-red
```

```
4 Aug 17:53:46 - [info]
```

```
Welcome to Node-RED
```

```
=====
```

```
4 Aug 17:53:46 - [info] Node-RED version: v1.1.2
```

```
4 Aug 17:53:46 - [info] Node.js version: v10.16.3
```

```
4 Aug 17:53:46 - [info] Windows_NT 10.0.18362 x64 LE
```

```
4 Aug 17:53:46 - [info] Loading palette nodes
```

```
4 Aug 17:53:48 - [info] Settings file : C:\Users\andre\.node-red\se
```

Instalando o Node-RED

Uma vez que você executou o Node-RED ele irá rodar um servidor web localmente (na sua máquina). Como acessar esse servidor web? Usando o navegador, digite o IP local e a porta onde está rodando o Node-RED

```
node-red
28 Mar 20:38:19 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
28 Mar 20:38:19 - [info] Starting flows
28 Mar 20:38:19 - [info] Started flows
28 Mar 20:38:19 - [info] Server now running at http://127.0.0.1:1880/
```

Copie e cole no seu navegador o número que aparece no seu terminal:

http://127.0.0.1:1880/

IP local

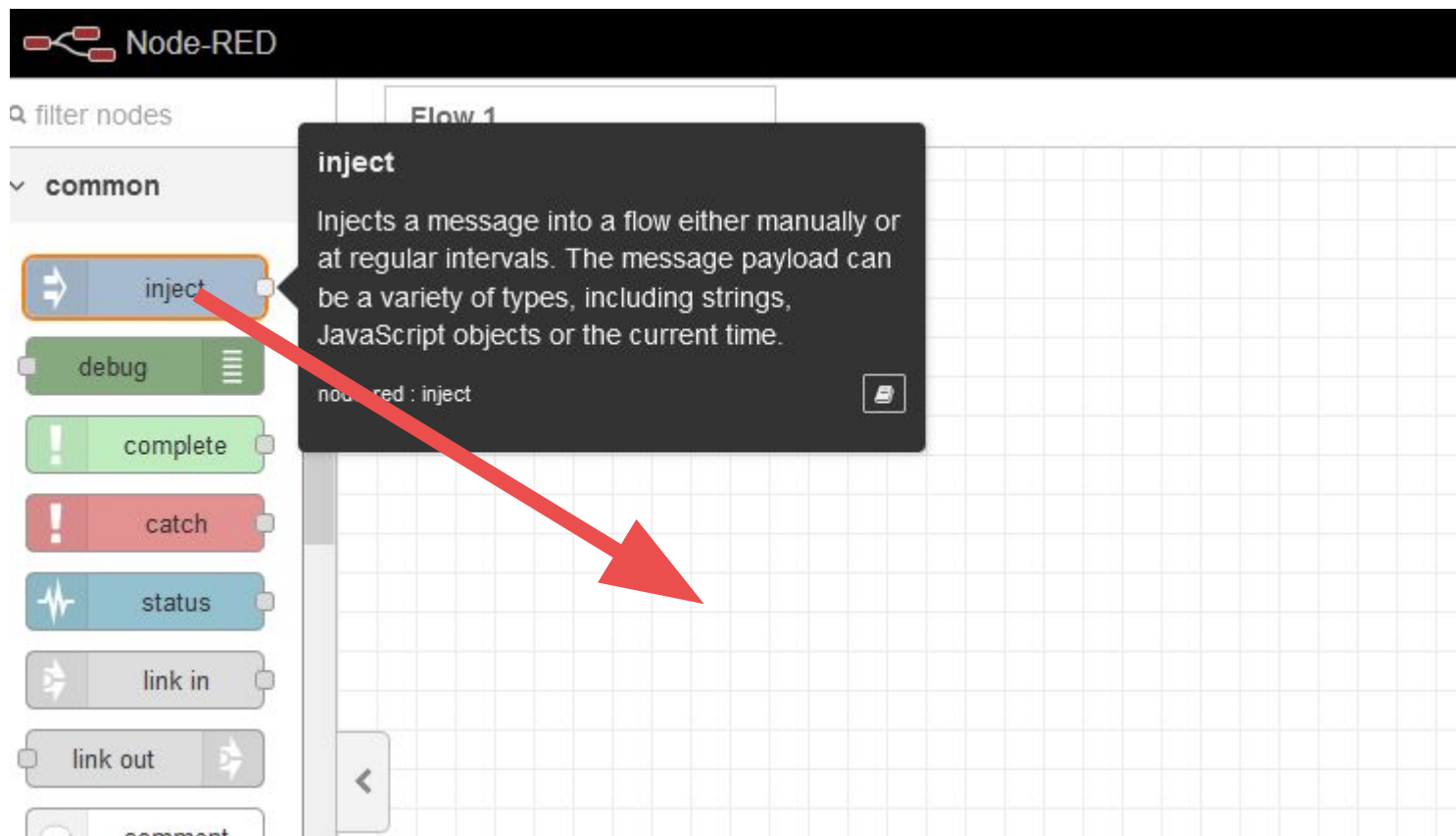
porta

Criando Fluxos com o Node-RED

Prof. Miguel Bozer da Silva

Criando um fluxo com o Node-RED

Segure e arraste o Nó de Inject da aba de nós para a área de desenvolvimento



Criando um fluxo com o Node-RED

Duplo clique no nó:

The screenshot shows the Node-RED interface with a flow named 'Flow 1' containing a 'timestamp' node. The 'Edit inject node' dialog is open, showing the 'Properties' tab. The 'msg.payload' field is highlighted with a red box and labeled 'Selezione string' with a red arrow. The 'msg.topic' field is set to 'a_z'. The 'Inject once after' checkbox is unchecked, and the 'Repeat' dropdown is set to 'none'.

Edit inject node

Delete Cancel Done

Properties

Name Name

msg.payload = timestamp

msg.topic = a_z

+ add

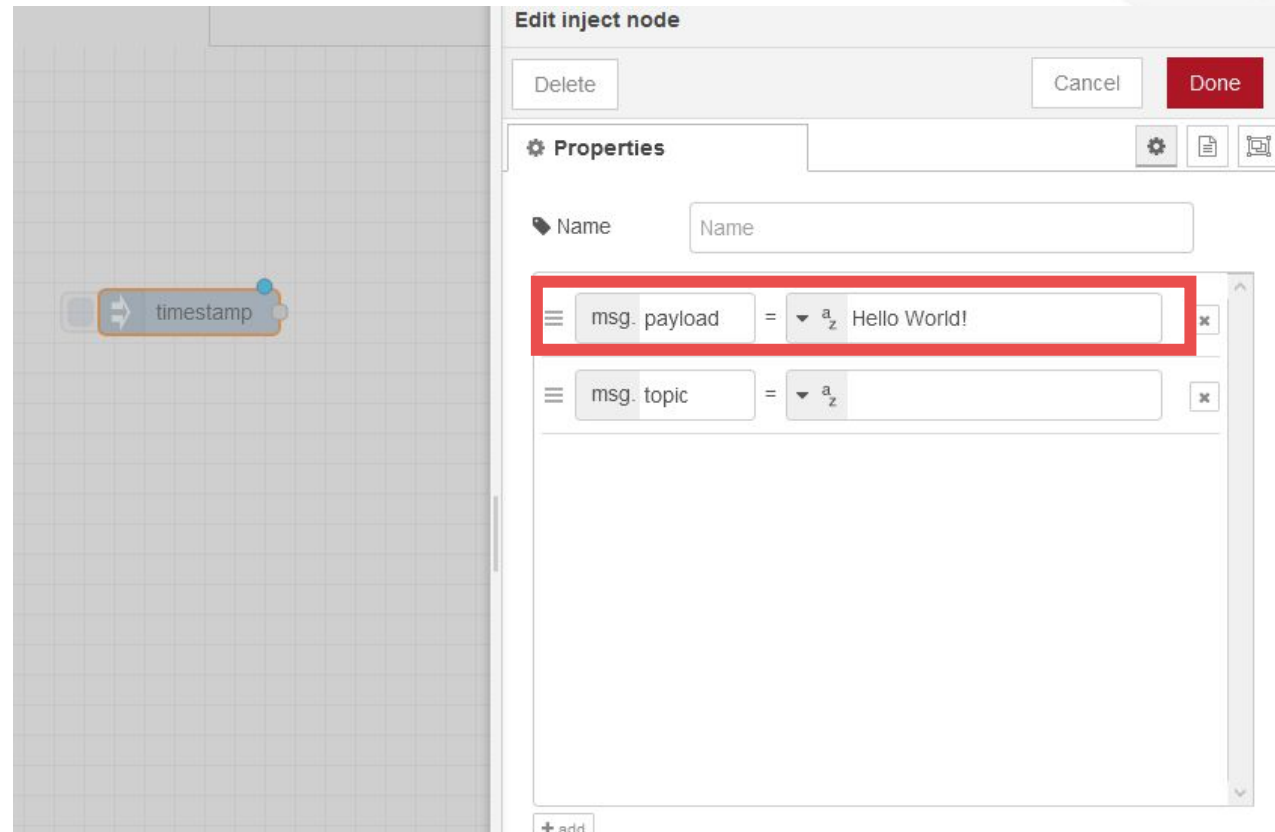
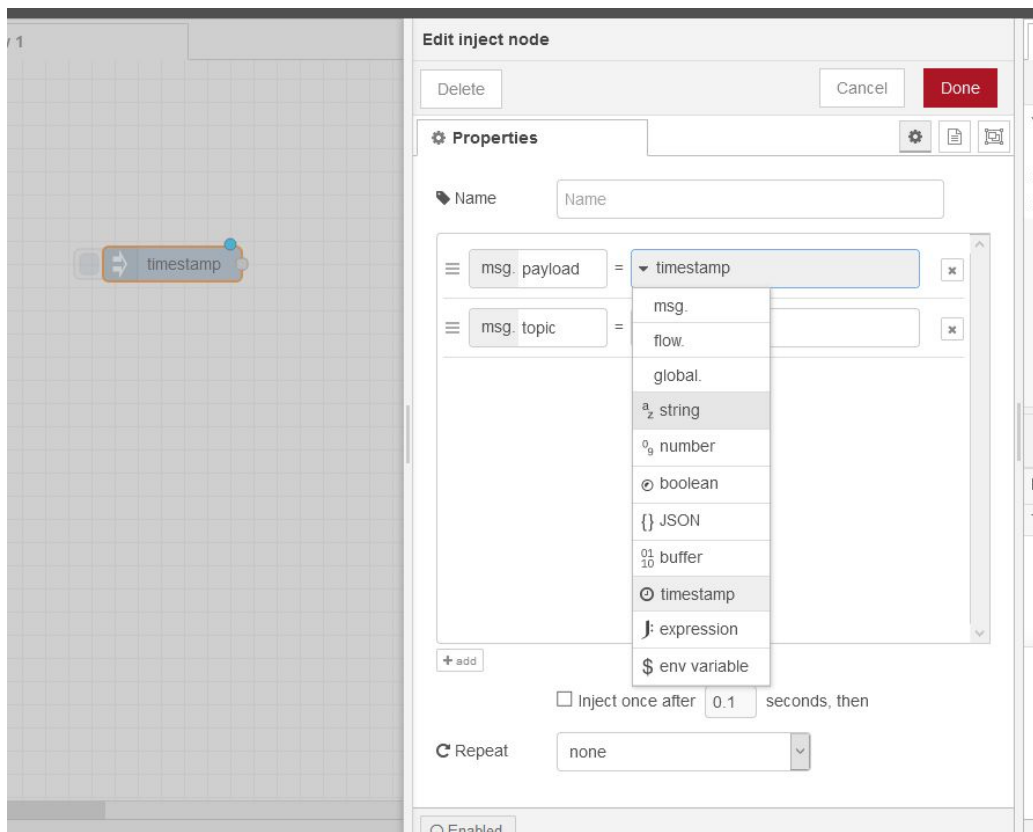
☐ Inject once after 0.1 seconds, then

Repeat none

Selezione string

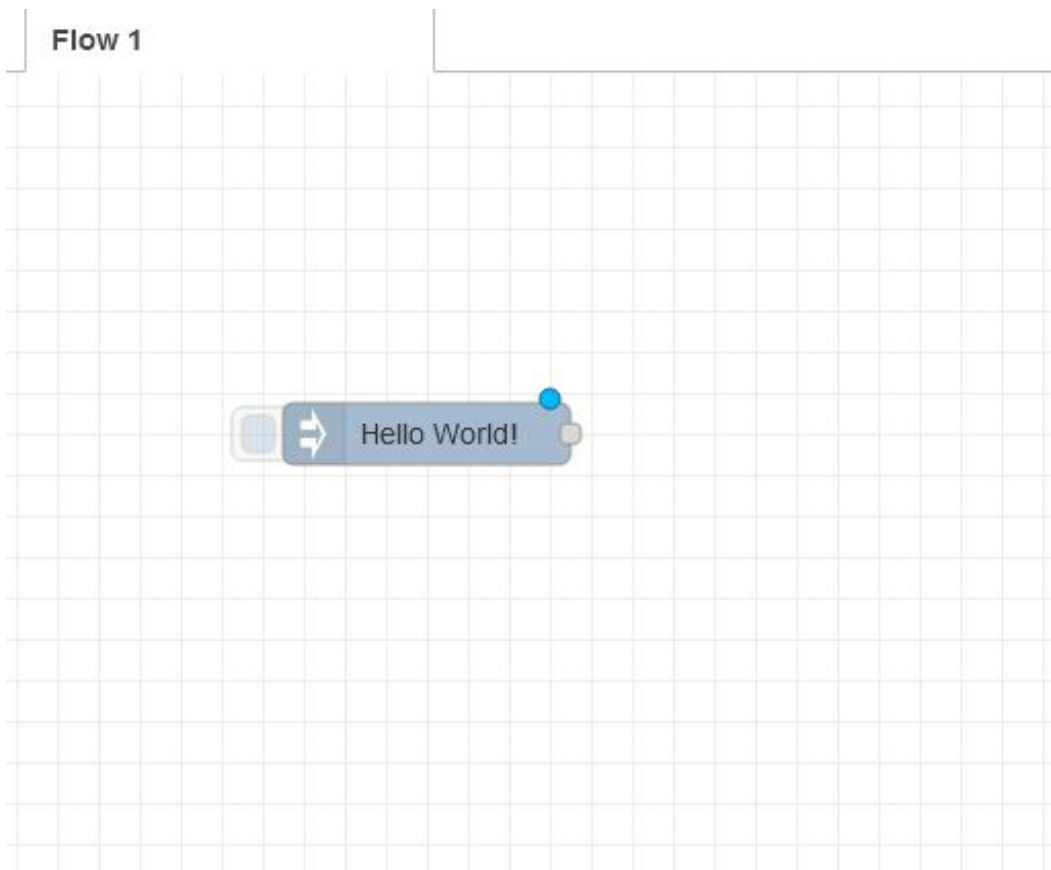
Criando um fluxo com o Node-RED

Após selecionar string, digite a mensagem em payload, e clique em Done:



Criando um fluxo com o Node-RED

O resultado será:



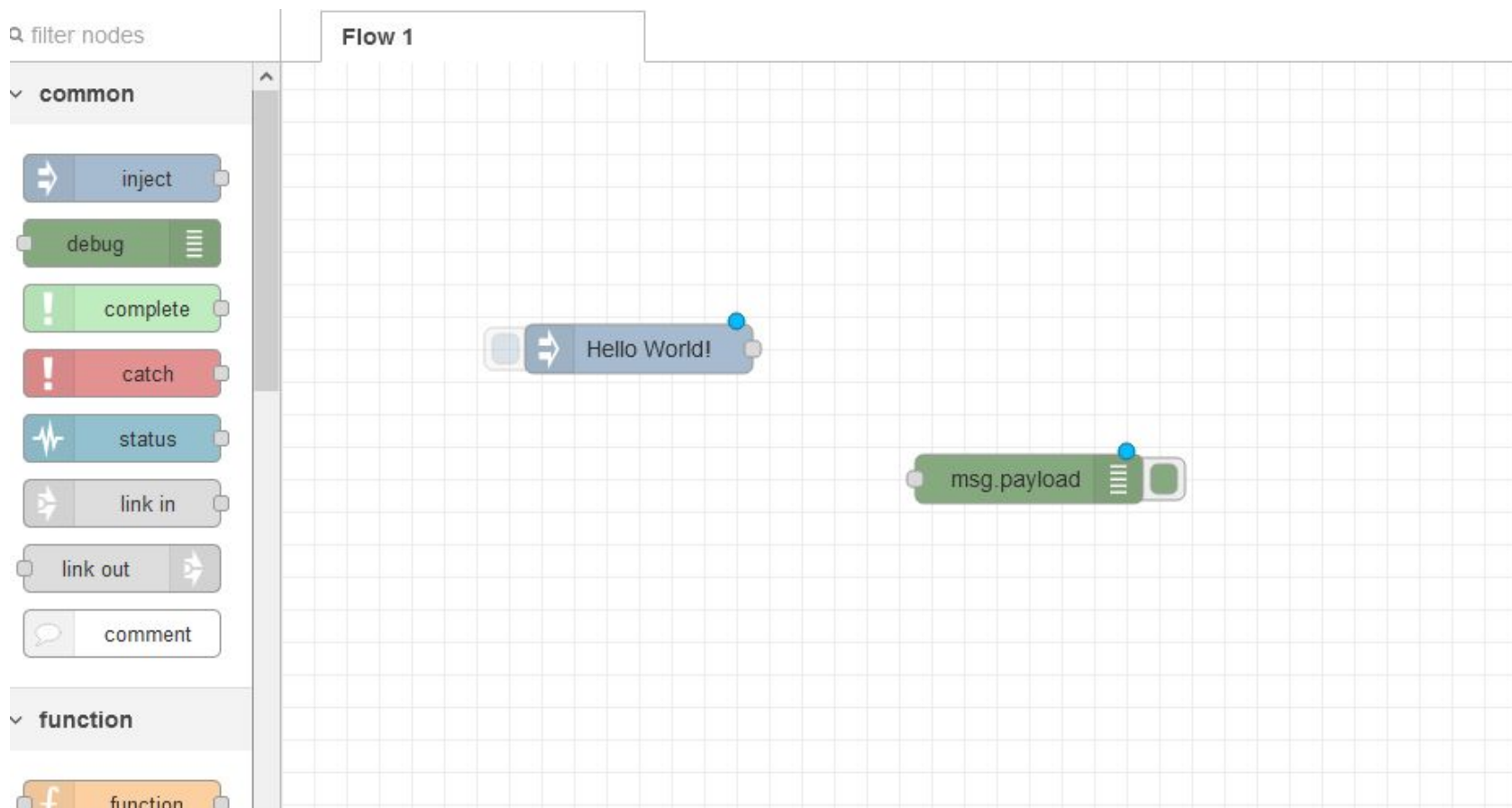
Este nó está “injetando” uma mensagem, do tipo **string**, cujo valor “**Hello World**”;

String é um tipo de variável; Existem variáveis numéricas inteiras (int), variáveis numéricas decimais (float), vetores/arrays entre outros objetos. Uma string é uma variável que armazena **dentro dela valores de texto**.

No Node-RED os nós criam e processam mensagens (msg) que contém um conteúdo (msg.payload). Neste caso, nossa payload é uma string, ou seja, um texto.

Criando um fluxo com o Node-RED

Precisamos de um destino para a nossa mensagem. Vamos usar um nó de debug. Arraste e solte:

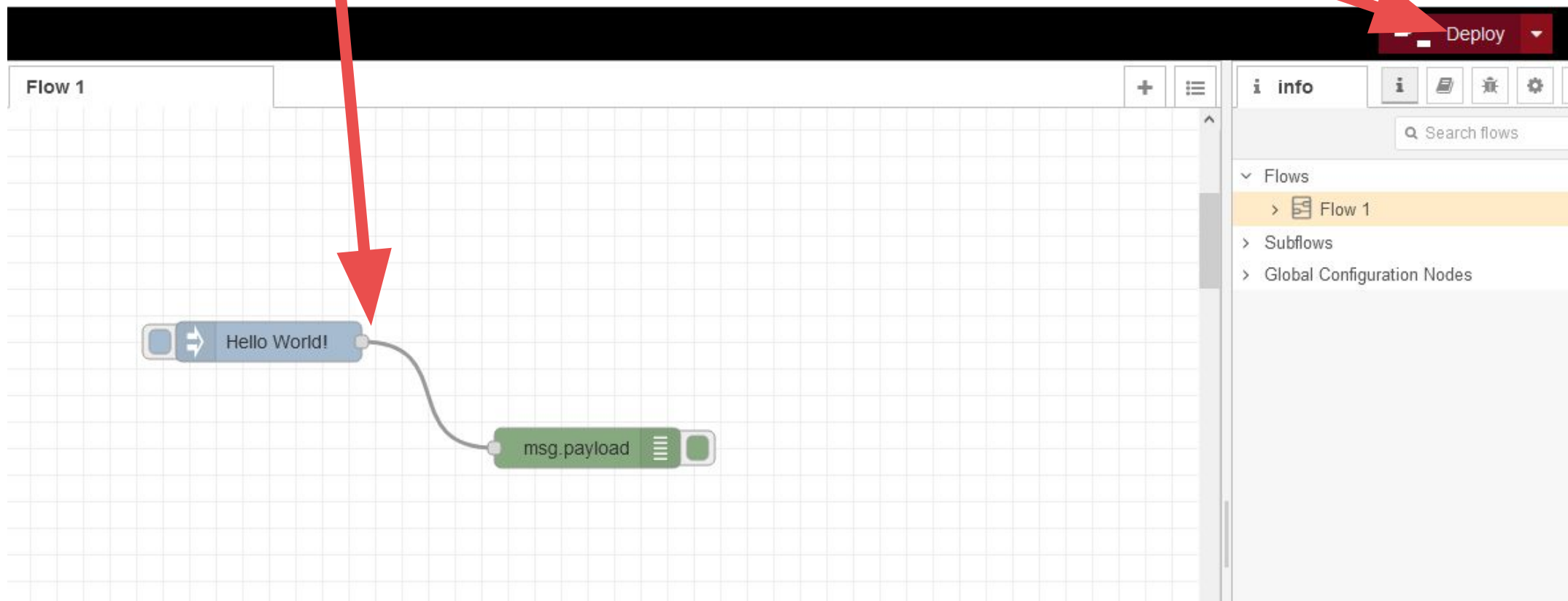


Debug é um termo usado sempre que você quiser testar um código para ver se ele funciona como você esperava.

Criando um fluxo com o Node-RED

Agora conecte os dois nós.

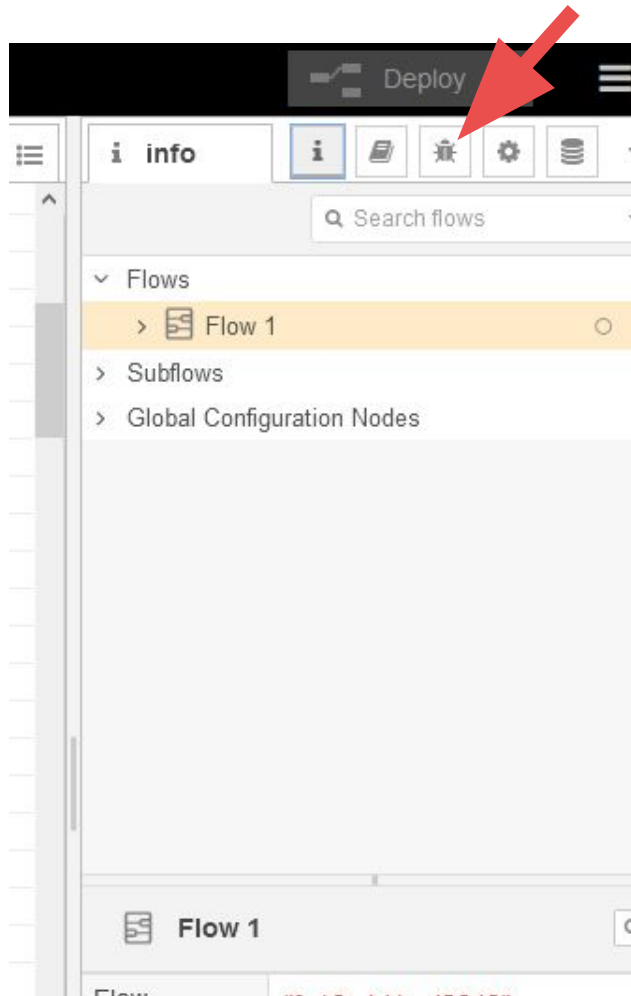
E clique em Deploy.



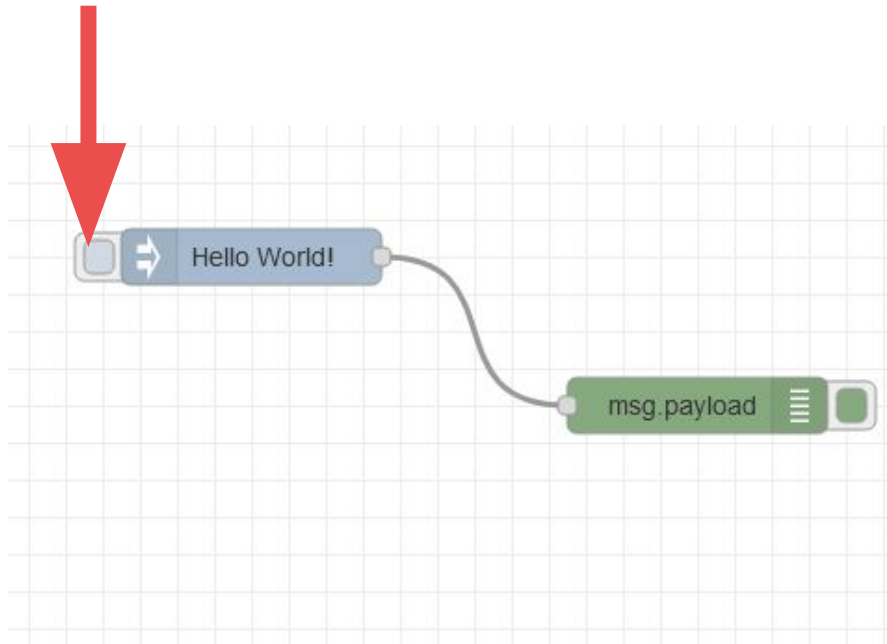
O Deploy irá permitir a execução do fluxo com ele no formato que você o deixou. **Cada alteração do fluxo necessitará de um novo Deploy!**

Criando um fluxo com o Node-RED

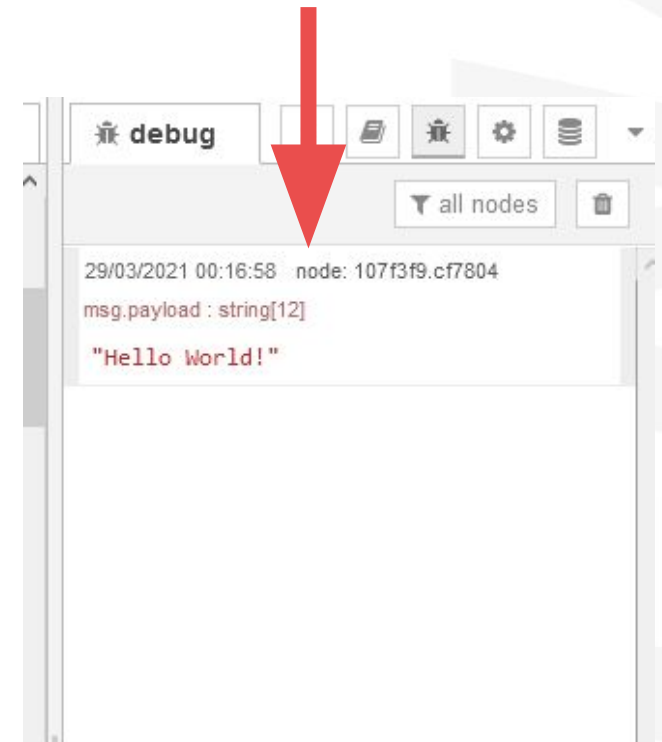
Abra o menu de debug



E injete e mensagem clicando no botão esquerdo do nó de Inject



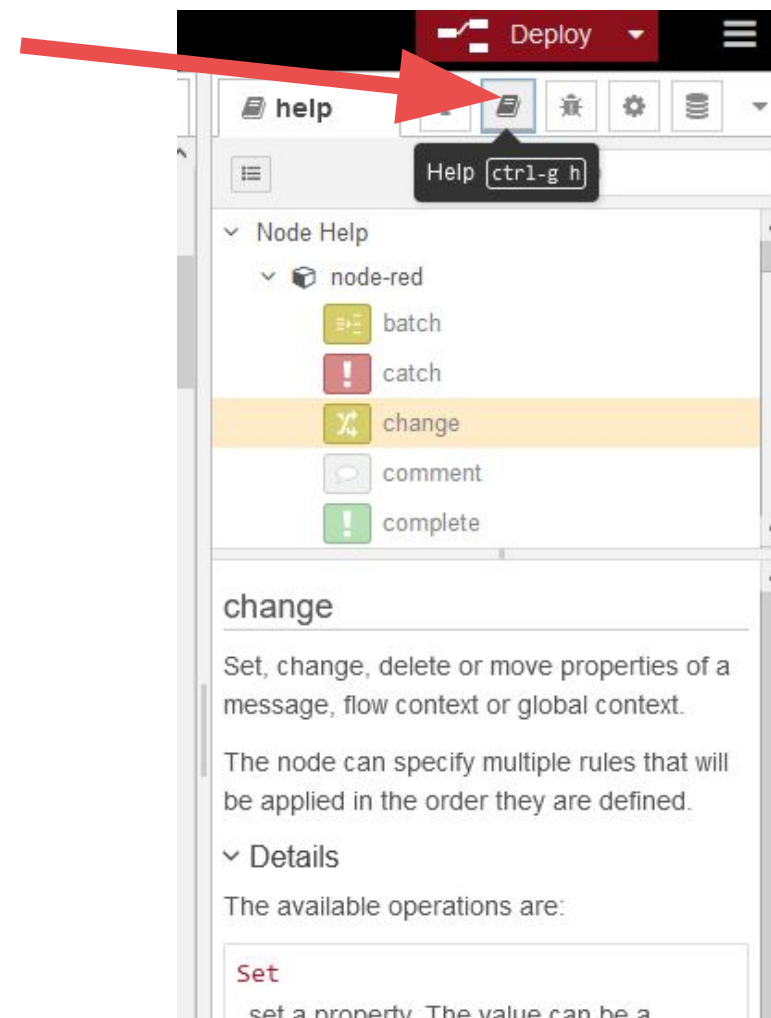
Resultado



Obs.: Podemos observar que foi exibido o resultado de msg.payload.

Como eu programo em Node-RED?

Menu ajuda (**help**) vocês encontram as especificações do que cada tipo de nó faz. Explore para conhecer mais!

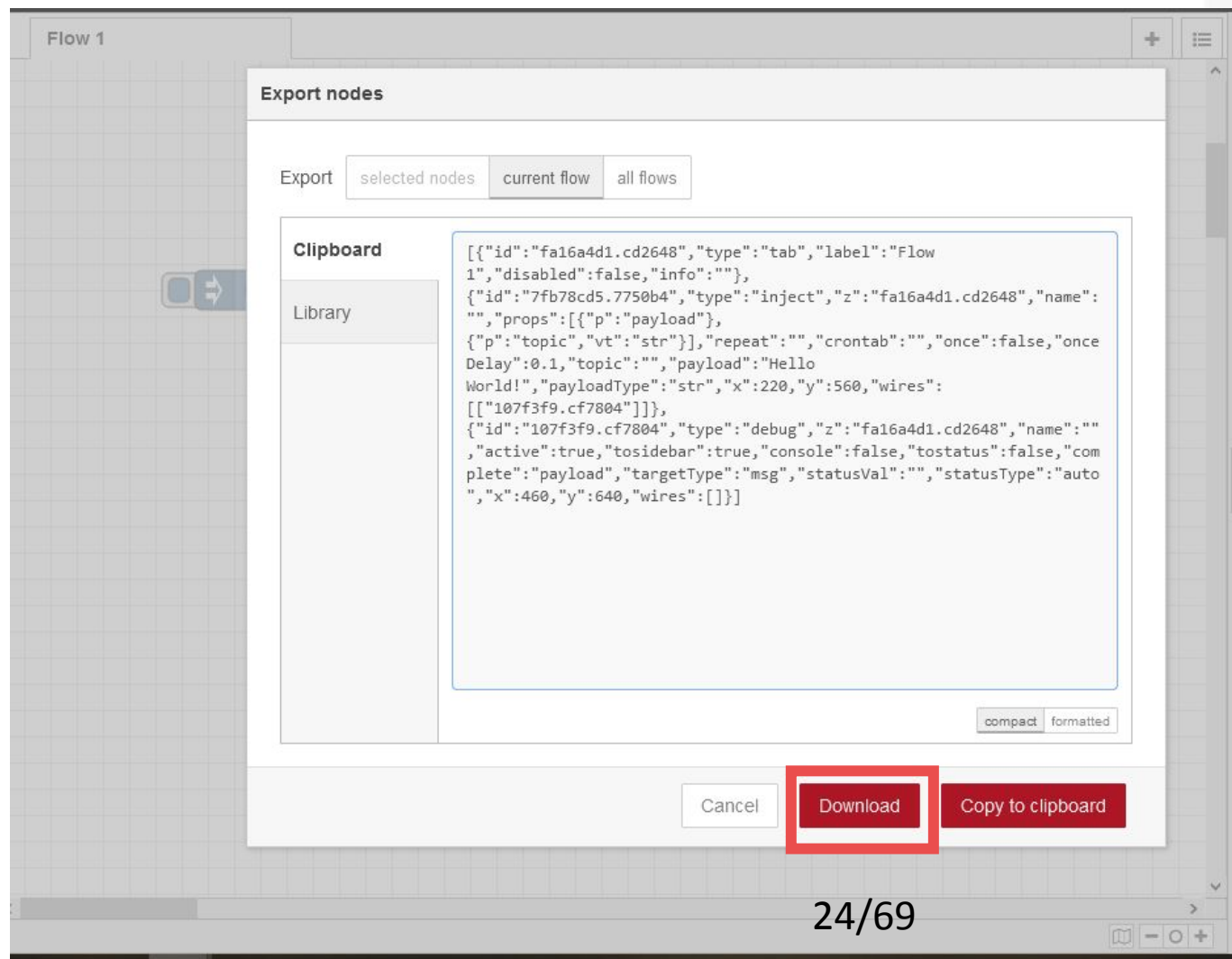
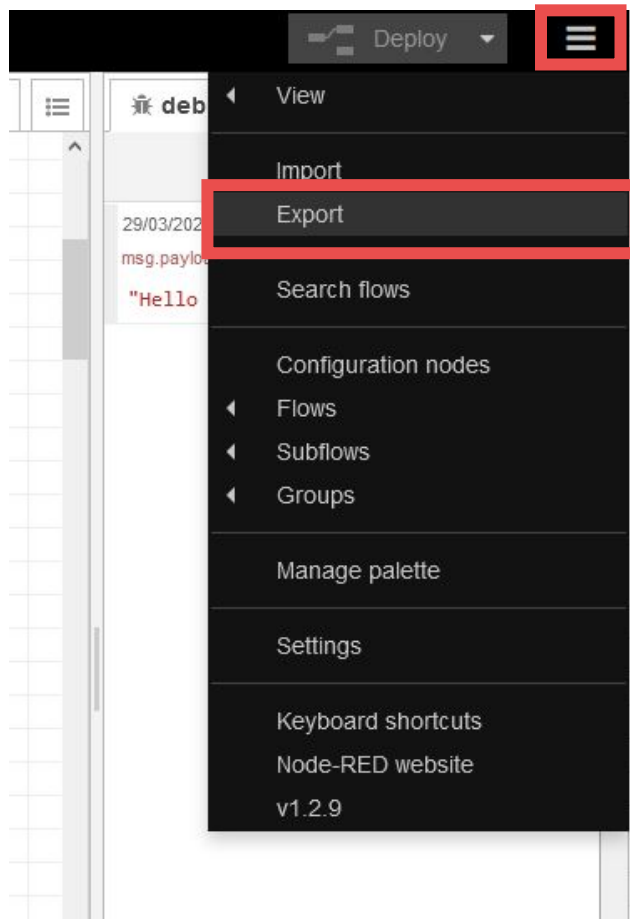


Nota: Exportando e Carregando Fluxos

Prof. Miguel Bozer da Silva

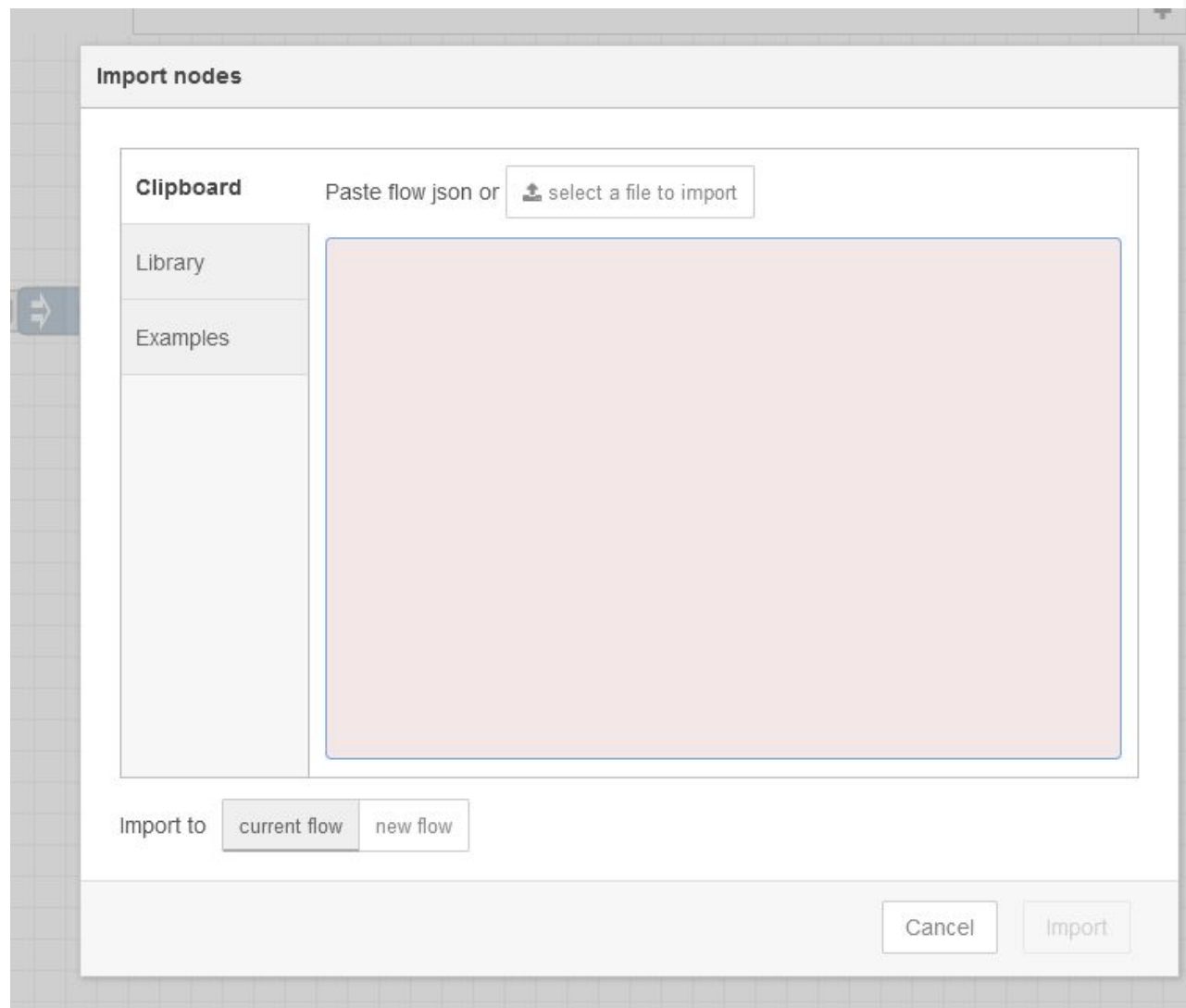
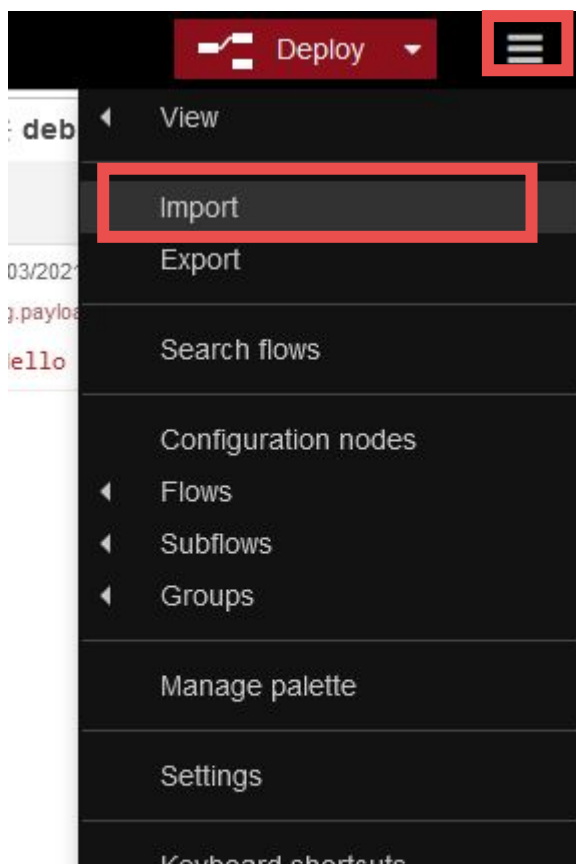
Exportando um fluxo do Node-RED

Salvando um fluxo:



Importando um fluxo do Node-RED

Carregando um fluxo:

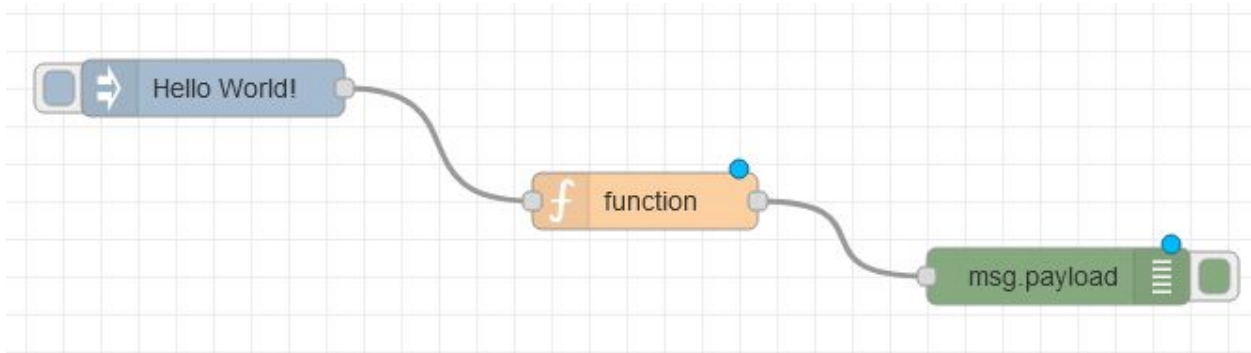


Funções customizadas

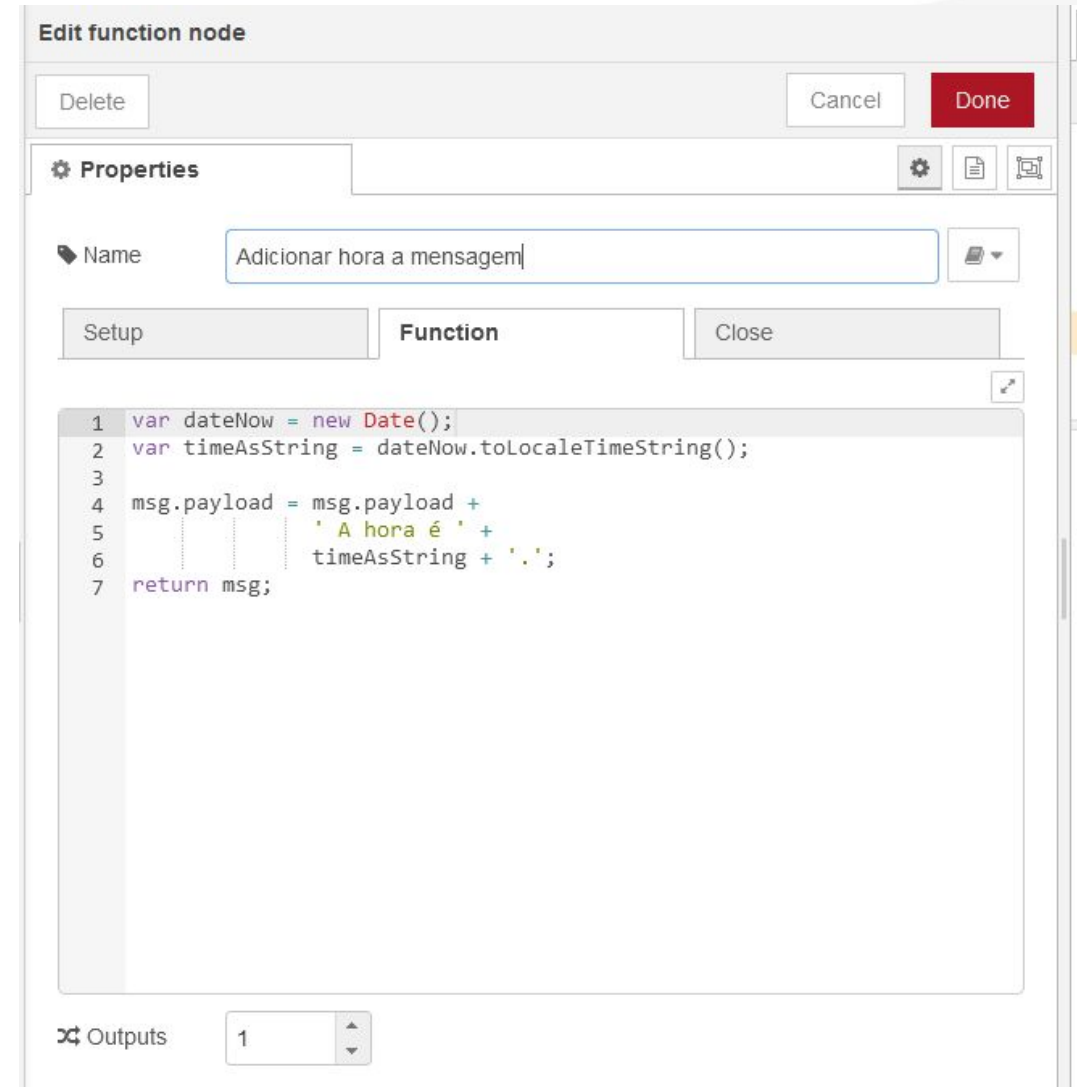
Prof. Miguel Bozer da Silva

Funções Customizadas

Arraste e solte o nó de function para o meio do fluxo 1:

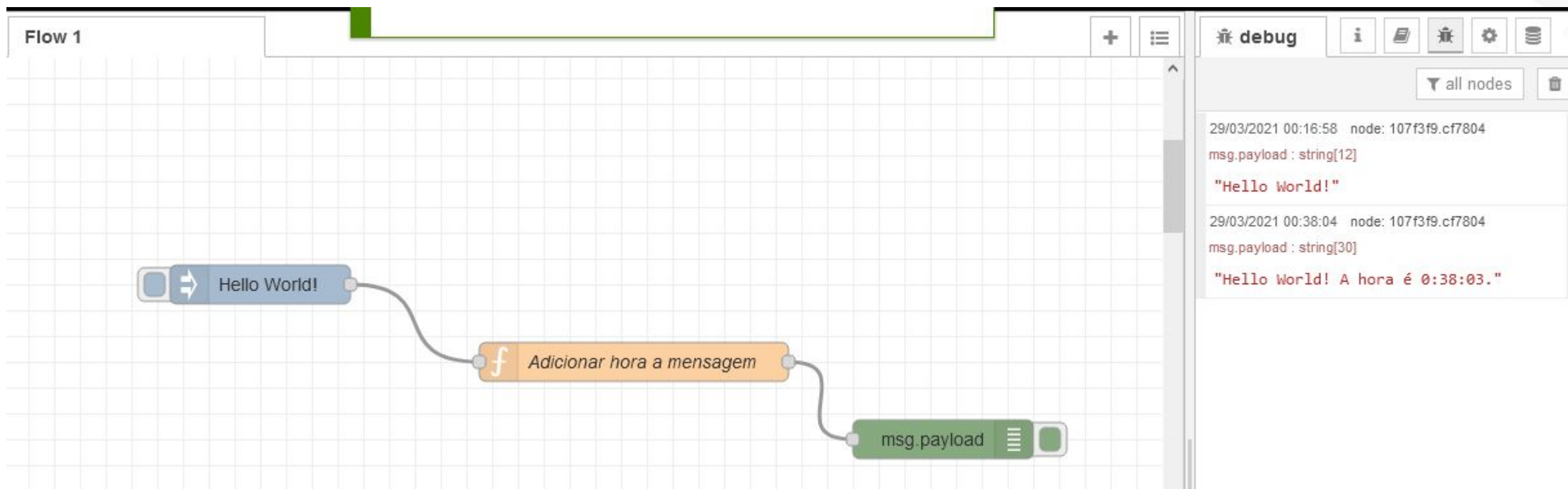


Duplo clique no nó function. Vamos inserir código JavaScript para ele pegar a hora. O nome do nó será “Adicionar hora a Mensagem”. Ao terminar, clique em Done.



Funções Customizadas

Clique em Deploy. Em seguida, com o menu de Debug aberto, aperte no botão de injeção ao lado do nó Hello World. Pronto!

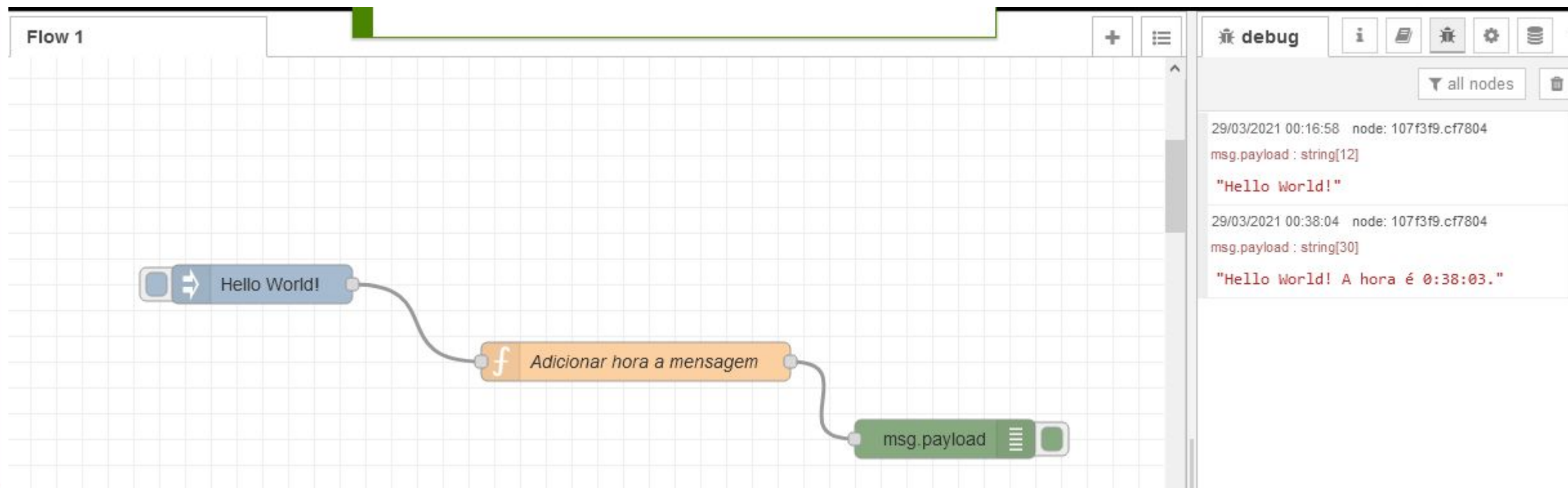


Funções Customizadas

```
var dateNow = new Date();  
var timeAsString = dateNow.toLocaleTimeString();  
msg.payload = msg.payload + 'A hora é ' + timeAsString + '.';  
  
return msg;
```

Funções Customizadas

Clique em Deploy. Em seguida, com o menu de Debug aberto, aperte no botão de injeção ao lado do nó Hello World. Pronto!



Exercício

Edit inject node

Delete Cancel Done

Properties

Name Name

msg.payload = a_z Hello World!

msg.topic = a_z

+ add

☐ Inject once after 0.1 seconds, then

Repeat interval

every 2 seconds

Vamos fazer alterar as configurações do nó de Inject do nosso problema conforme a imagem ao lado. Clique em Deploy após as alterações. **O que acontece agora?**

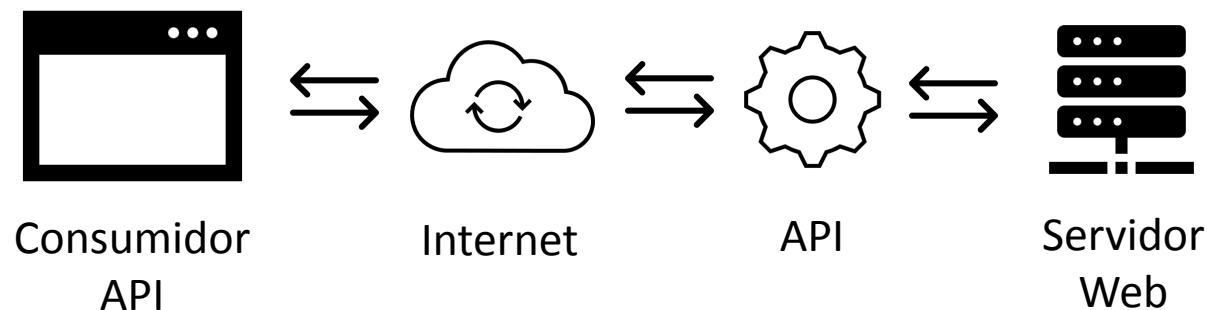
O que mais podemos fazer com o Node-RED?

Prof. Miguel Bozer da Silva

Conectando com APIs

Podemos usar APIs para fazer diferentes integrações:

- **Interface:** A API (Application Programming Interface) age como uma "ponte" que conecta diferentes softwares, funcionando como uma **interface de comunicação**. Os desenvolvedores utilizam a API para pedir dados ou funcionalidades de outro sistema sem precisar saber como ele funciona internamente.



Conectando com APIs

Vamos fazer um exemplo com o Node-RED para compreendermos esse conceito

Open-Meteo API

A Open-Meteo API Calibri (Corpo) oferece dados meteorológicos

Facilidade de Uso

A API é acessível através de uma URL simples

Dados Completos

A API fornece dados meteorológicos abrangentes, incluindo temperatura, umidade, velocidade do vento e previsão.

Conectando com APIs

Consultando a API

A URL para obter o clima atual é:

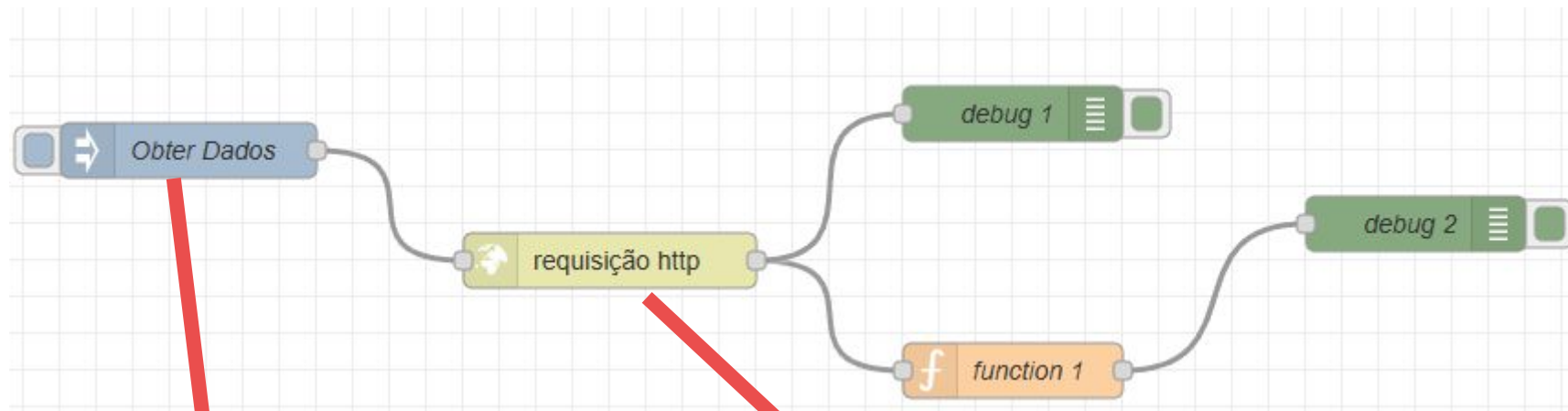
```
https://api.open-meteo.com/v1/forecast?latitude={lat}&longitude={lon}&at_weather=true
```

Onde {lat} é a latitude e {lon} é a longitude da localidade.



Conectando com APIs

Vamos criar o seguinte fluxo no Node-RED:



Propriedades

Nome Obter Dados

msg.payload = milliseconds since epoch

msg.topic = a_z

Propriedades

Método GET

URL

https://api.open-meteo.com/v1/forecast?latitude=-23.6229

https://api.open-meteo.com/v1/forecast?latitude=-23.6229&longitude=-46.5548¤t_weather=true

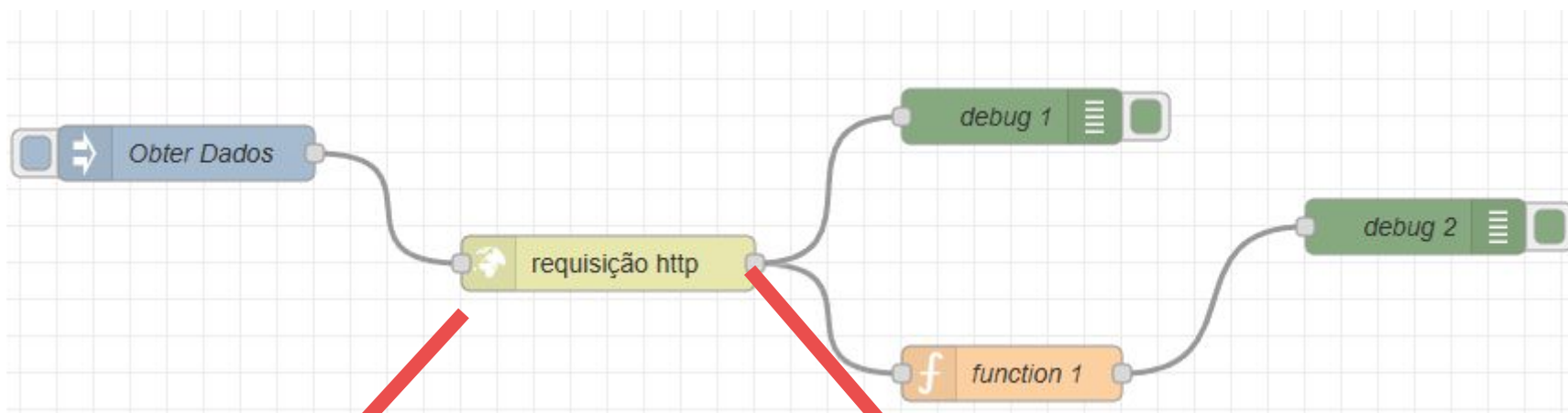
A latitude e longitude de São Caetano do Sul:

Latitude: -23.6229

Longitude: -46.5548

Conectando com APIs

Vamos criar o seguinte fluxo no Node-RED:



Solicitação (Request): É quando o cliente (software que está pedindo os dados) faz uma solicitação à API. Geralmente, as solicitações seguem o protocolo HTTP e têm um endpoint específico (URL)

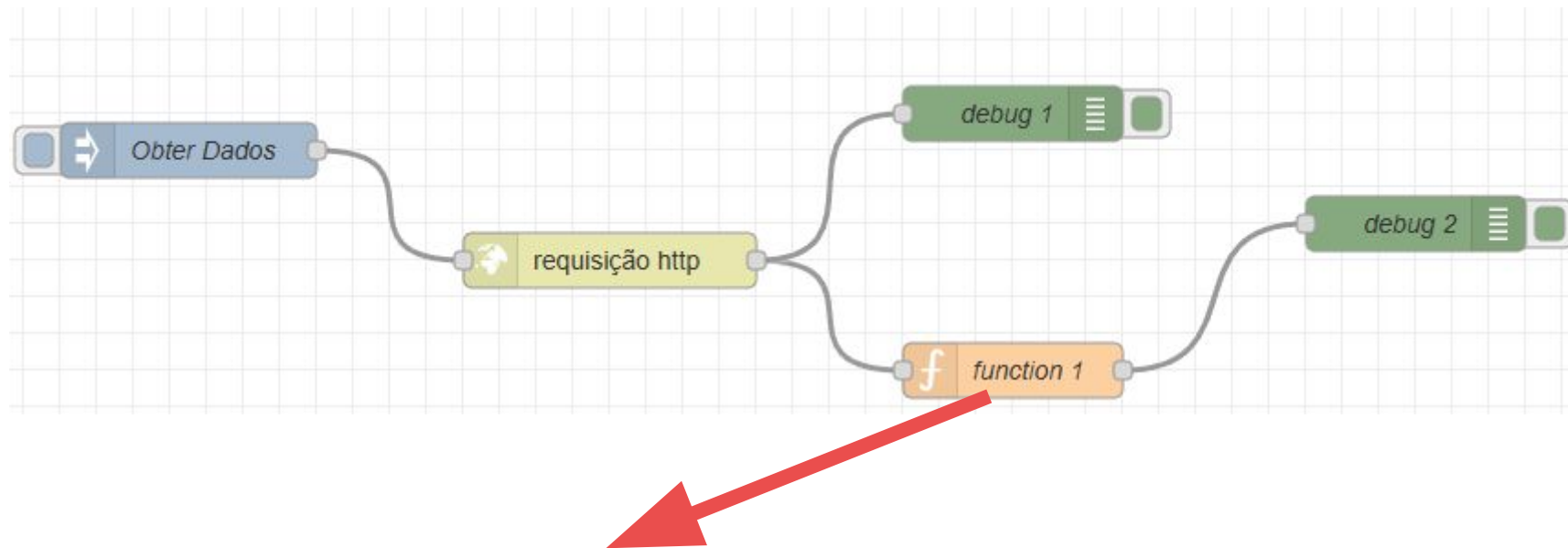
Métodos HTTP:

- GET:** Solicita dados (ex: dados de clima).

Resposta (Response): A resposta que a API envia para o cliente. Pode ser um dado em um formato como **JSON** ou **XML**, ou uma confirmação de que uma ação foi realizada.

Conectando com APIs

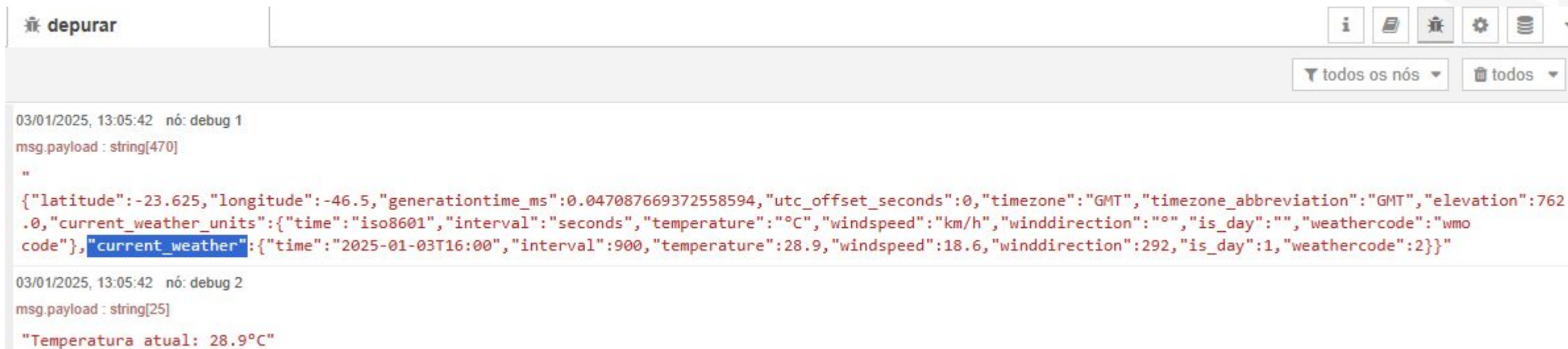
Vamos criar o seguinte fluxo no Node-RED:



```
var weatherData = msg.payload;  
var parsedData = JSON.parse(weatherData);  
msg.payload = "Temperatura atual: " + parsedData.current_weather.temperature + "°C";  
  
return msg;
```

Conectando com APIs

Resultados obtidos:



depurar

03/01/2025, 13:05:42 nó: debug 1
msg.payload : string[470]

```
"  
{  
  "latitude": -23.625,  
  "longitude": -46.5,  
  "generationtime_ms": 0.047087669372558594,  
  "utc_offset_seconds": 0,  
  "timezone": "GMT",  
  "timezone_abbreviation": "GMT",  
  "elevation": 762.0,  
  "current_weather_units": {  
    "time": "iso8601",  
    "interval": "seconds",  
    "temperature": "°C",  
    "windspeed": "km/h",  
    "winddirection": "°",  
    "is_day": "",  
    "weathercode": "wmo code"},  
    "current_weather": {  
      "time": "2025-01-03T16:00",  
      "interval": 900,  
      "temperature": 28.9,  
      "windspeed": 18.6,  
      "winddirection": 292,  
      "is_day": 1,  
      "weathercode": 2}}  
}
```

03/01/2025, 13:05:42 nó: debug 2
msg.payload : string[25]

```
"Temperatura atual: 28.9°C"
```


Conectando com APIs

```
{
  "latitude": -23.625,
  "longitude": -46.5,
  "generationtime_ms": 0.028014183044433594,
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 762.0,
  "current_weather_units": {
    "time": "iso8601",
    "interval": "seconds",
    "temperature": "°C",
    "windspeed": "km/h",
    "winddirection": "°",
    "is_day": "",
    "weathercode": "wmo code"
  },
  "current_weather": {
    "time": "2025-01-03T16:45",
    "interval": 900,
    "temperature": 28.7,
    "windspeed": 17.7,
    "winddirection": 287,
    "is_day": 1,
    "weathercode": 2
  }
}
```

O resultado obtido é um documento JSON (chave-valor)

A conversão feita em JavaScript faz com que esse documento seja convertido em um objeto com diversas propriedades

Conectando com APIs

```
{
  "latitude": -23.625,
  "longitude": -46.5,
  "generationtime_ms": 0.028014183044433594,
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 762.0,
  "current_weather_units": {
    "time": "iso8601",
    "interval": "seconds",
    "temperature": "°C",
    "windspeed": "km/h",
    "winddirection": "°",
    "is_day": "",
    "weathercode": "wmo_code"
  },
  "current_weather": {
    "time": "2025-01-03T16:45",
    "interval": 900,
    "temperature": 28.7,
    "windspeed": 17.7,
    "winddirection": 287,
    "is_day": 1,
    "weathercode": 2
  }
}
```

```
var parsedData = JSON.parse(weatherData);
```



Converte JSON para um objeto

```
parsedData.current_weather.temperature
```



Acessando uma propriedade do objeto

Exercício

Mudar o fluxo anterior para exibir tanto a temperatura quanto a velocidade do vento na cidade de São Caetano do Sul. Após isso, faça isso para outra cidade da sua escolha.

Resultado esperado:

```
03/01/2025, 13:45:24  nó: debug 2
```

```
msg.payload : string[51]
```

```
▼ string[51]
```

```
Temperatura atual: 28.7°C  
Velocidade do vento: 17.7
```