

# AI & CHATBOT

Aula 05 – Introdução ao Node-RED e à  
Integração de Serviços em Nuvem

Prof. Henrique Ferreira  
Prof. Miguel Bozer  
Prof. Guilherme Aldeia  
Prof. Michel Fornaciali  
Prof. Vinicius Holanda

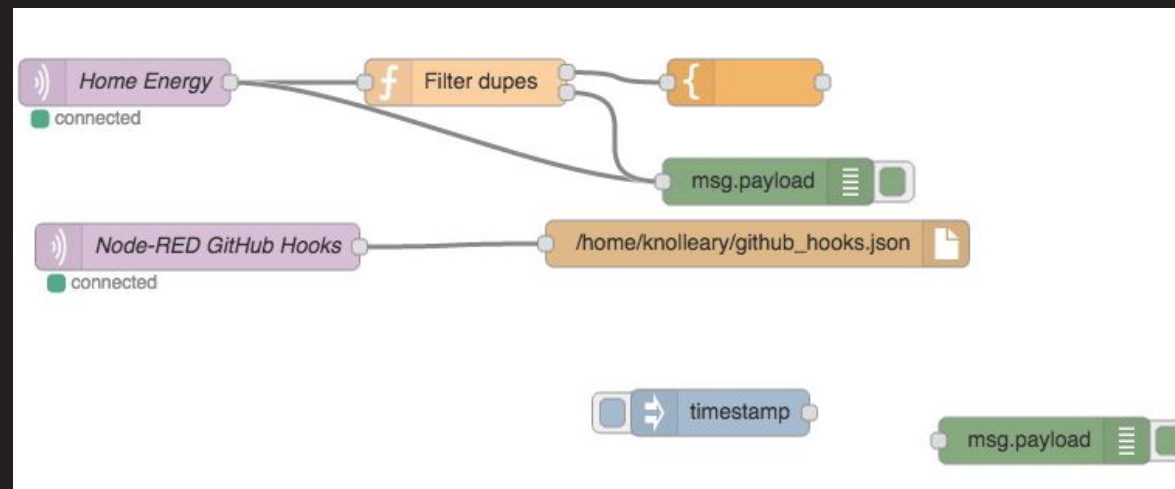
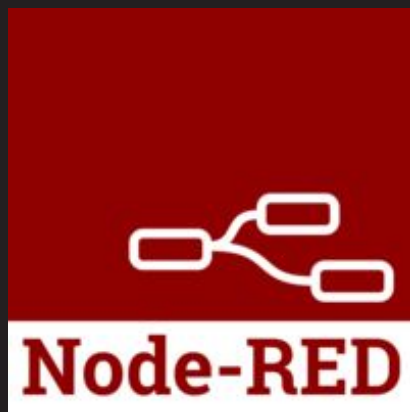
FIAP  
GRADUAÇÃO

# Node-RED 101

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

# O que é o Node-RED?

- O Node-RED é uma ferramenta de **programação visual** 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 ou remoto** (rodando no servidor de uma empresa ou de outra pessoa) 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 aprender tanto o desenvolvimento em nuvem quanto o desenvolvimento local.



## Running locally

Installing Node-RED on your local computer



## Raspberry Pi

Get started using our all-in-one install script for the mighty Raspberry Pi



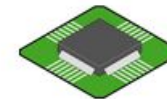
## Docker

Running Node-RED using Docker



## Install from git

Building Node-RED from source. Get the very latest development code and start contributing.



## BeagleBone Boards

Running Node-RED on BeagleBone boards



## Android

A bit experimental, but you can run on Android devices using Termux



## IBM Cloud

Deploying Node-RED from the IBM Cloud catalog in a couple of clicks



## AWS

Get started running on Elastic Beanstalk or EC2



## Microsoft Azure

Running on an Azure Virtual Machine instance

# Como eu programo em Node-RED?

Aba de nós

Aba de desenvolvimento do fluxo

Aba de informações

The image shows the Node-RED web interface. On the left, the 'common' nodes panel is visible, containing nodes like inject, debug, complete, catch, status, link in, link out, and comment. In the center is a large grid workspace for developing flows. On the right, the information sidebar shows a list of flows, with 'Flow 1' selected. Below this, the details for 'Flow 1' are displayed, including its ID 'fa16a4d1.cd2648'. At the bottom of the sidebar, there is a message: 'You can confirm your changes in the node edit tray with `ctrl-enter` or cancel them with `ctrl-escape`'. Three red arrows point to the 'common' nodes panel, the central workspace, and the right-hand information sidebar.

# 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

- csv
- html
- json
- xml
- yaml



# Node-RED no meu computador

Instalando o Node-RED localmente




# Instalando o NodeJS

<https://nodejs.org/en/download/>

## Downloads

Latest LTS Version: **14.16.0** (includes npm 6.14.11)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 <b>Windows Installer</b> <small>node-v14.16.0-x64.msi</small>	 <b>macOS Installer</b> <small>node-v14.16.0.pkg</small>	 <b>Source Code</b> <small>node-v14.16.0.tar.gz</small>

### Windows Installer (.msi)

32-bit

64-bit

### Windows Binary (.zip)

32-bit

64-bit

### macOS Installer (.pkg)

64-bit

### macOS Binary (.tar.gz)

64-bit

### Linux Binaries (x64)

64-bit

### Linux Binaries (ARM)

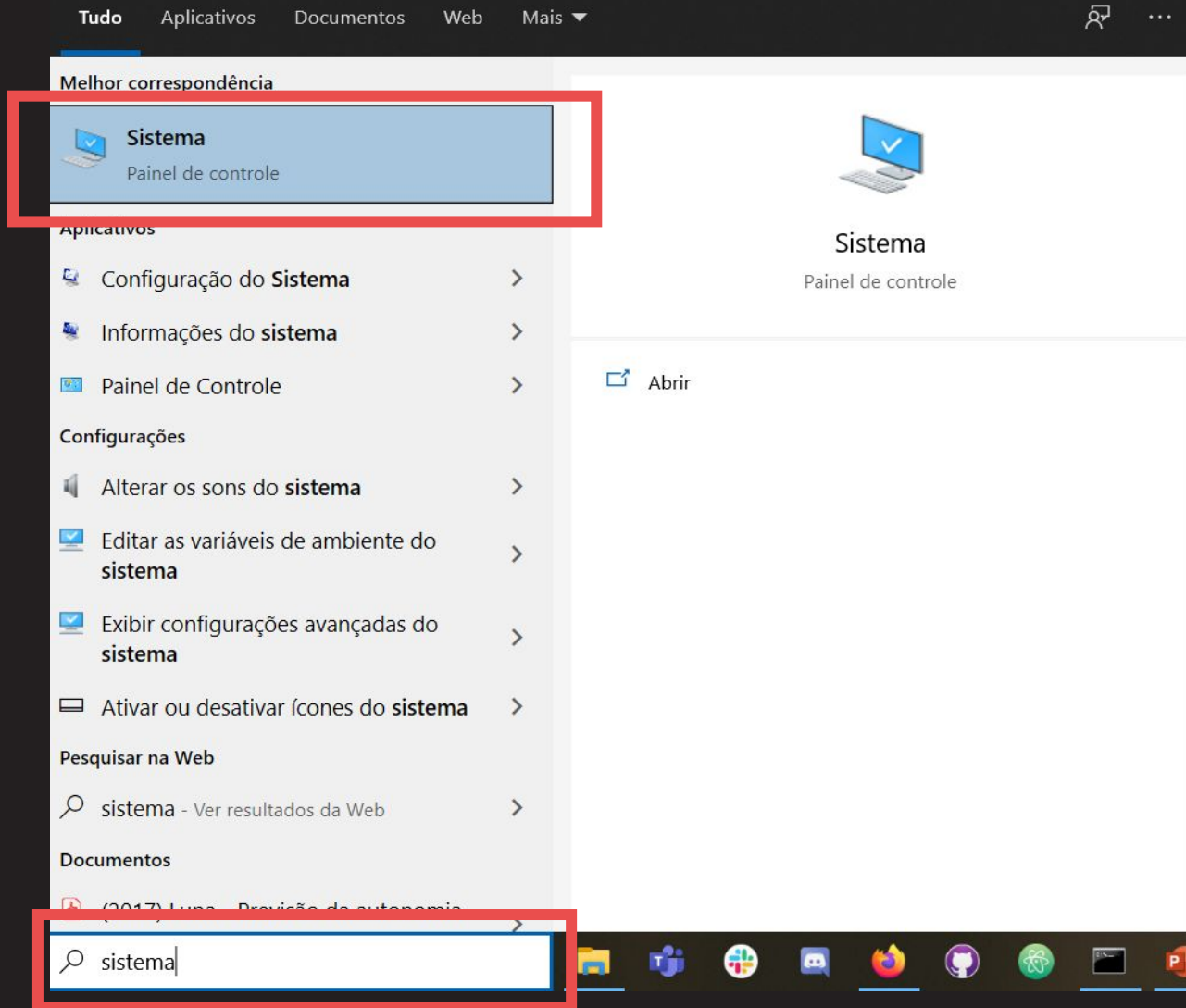
ARMv7

ARMv8

### Source Code

node-v14.16.0.tar.gz

# 32-bit ou 64-bit?



## Sobre

O computador está monitorado e protegido.

[Veja detalhes em Segurança do Windows](#)

## Especificações do dispositivo

### Inspiron 5584

Nome do dispositivo	DESKTOP-E3VFKJ5
Processador	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
RAM instalada	8,00 GB (utilizável: 7,87 GB)
ID do dispositivo	38C82385-F481-4101-AD05-F5A3AB9450BA
ID do Produto	00342-41396-69232-AAOEM

Tipo de sistema	Sistema operacional de 64 bits, processador baseado em x64
-----------------	--

Caneta e toque	Nenhuma entrada à caneta ou por toque disponível para este vídeo
----------------	--

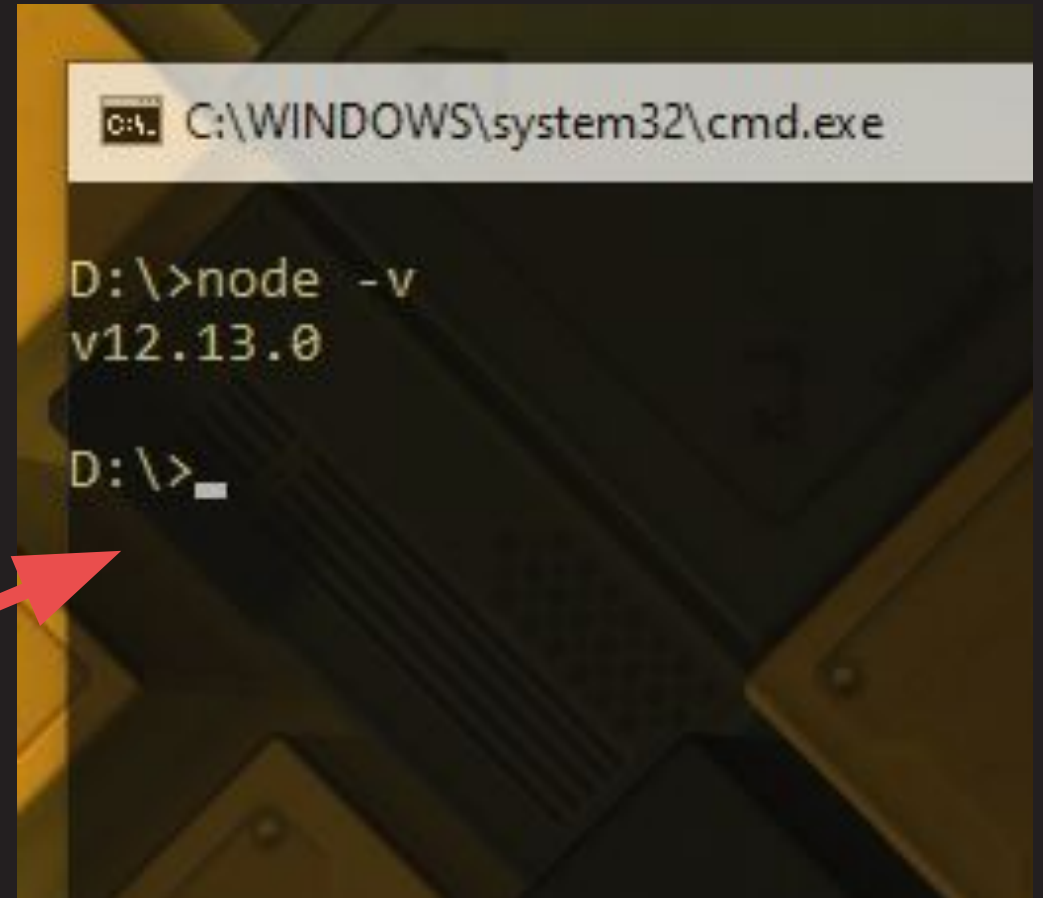
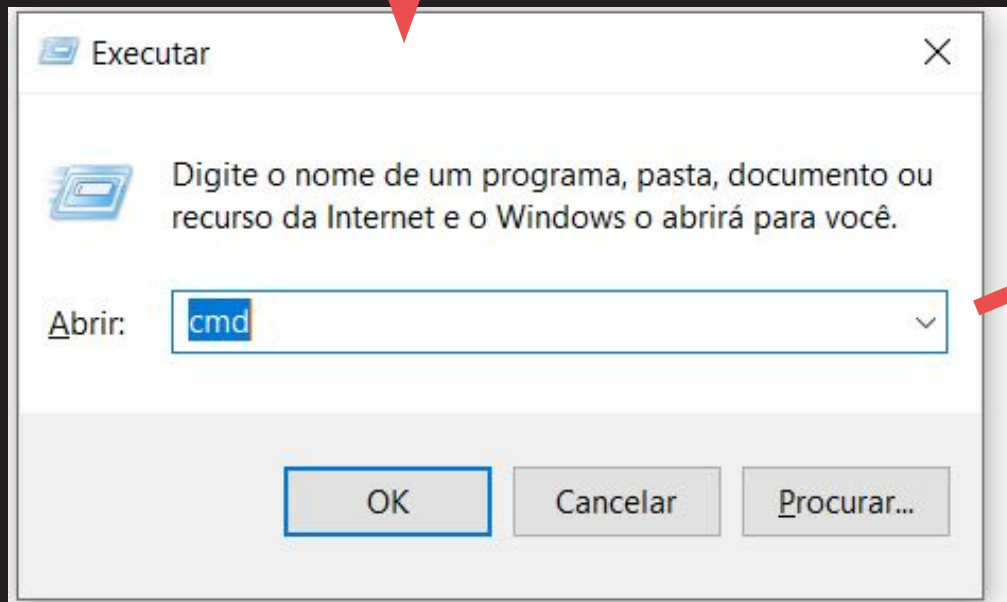
Copiar

Renomear este computador

## Especificações do Windows

# 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 executando, o Node-RED roda 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

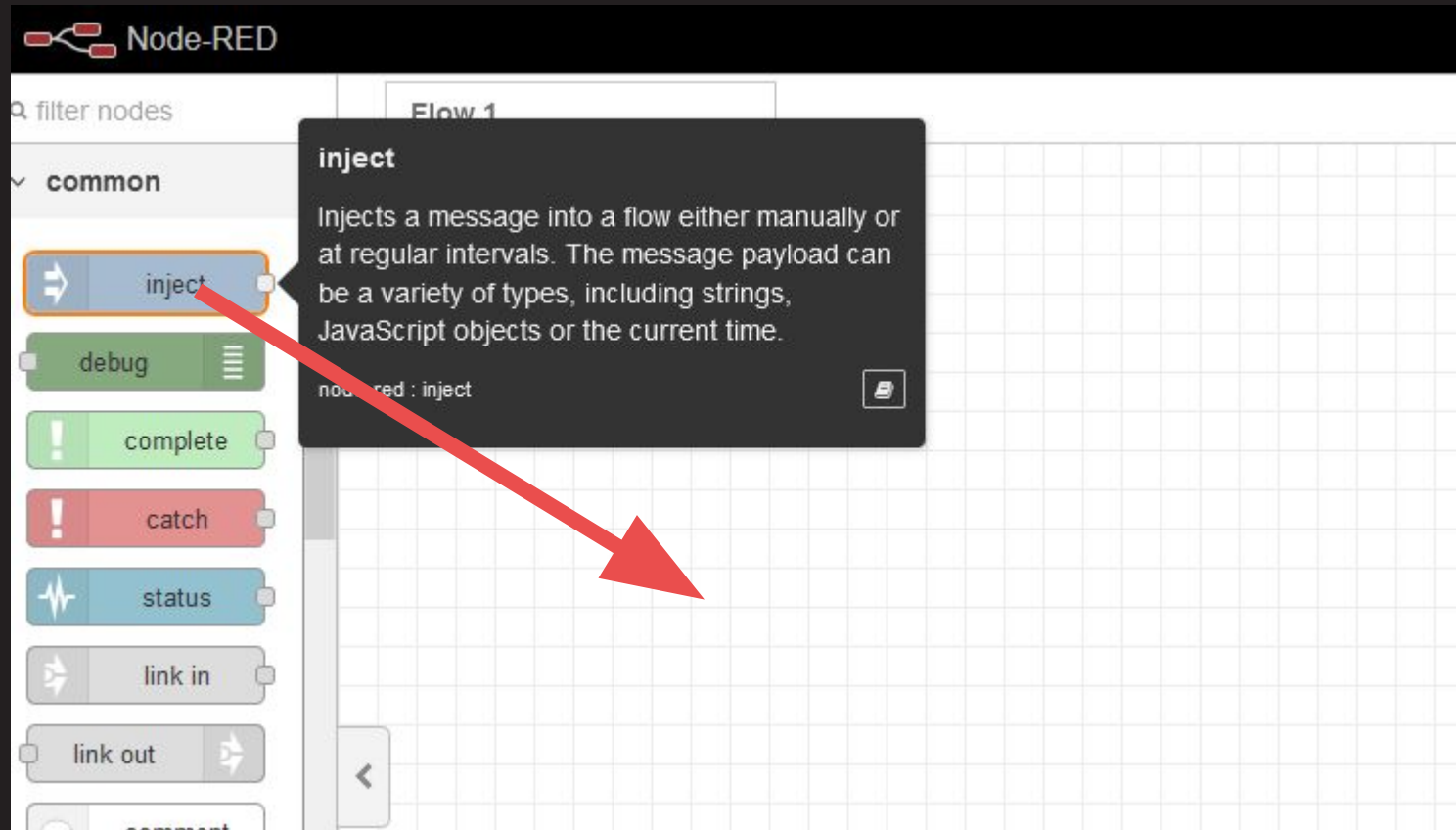
# Hello World!

O primeiro fluxo com Node-RED



# Primeiro fluxo com Node-RED

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



# Primeiro fluxo com Node-RED

Duplo clique no nó:

Selecione string

The screenshot shows the Node-RED interface with the 'Edit inject node' dialog open. The dialog has a 'Properties' tab and a 'Name' field. The 'msg.payload' field is highlighted with a red box and contains the text 'timestamp'. A red arrow points from the text 'Selecione string' to this field. The 'msg.topic' field is set to 'a\_z'. The 'Inject once after' checkbox is unchecked, and the 'Repeat' dropdown is set to 'none'. The main workspace shows a 'timestamp' node in the flow.

Node-RED

Flow 1

filter nodes

common

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

function

- function
- switch
- change
- range
- template

timestamp

Edit inject node

Delete Cancel Done

Properties

Name

msg.payload = timestamp

msg.topic = a\_z

+ add

☐ Inject once after 0.1 seconds, then

Repeat none

info

Search flows

Flows

- Flow 1
- Subflows
- Global Configuration Nodes

timestamp

Node "7fb78cd5.7750b4"

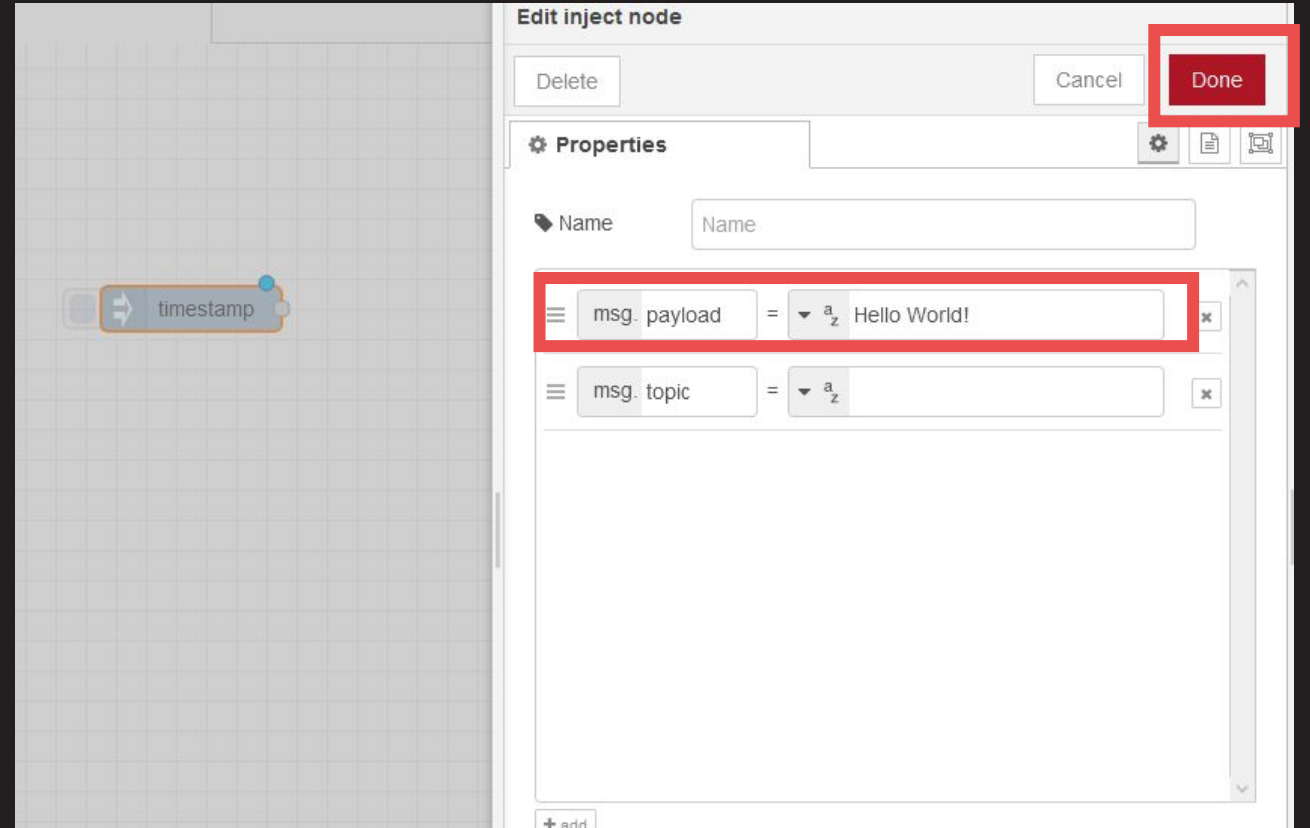
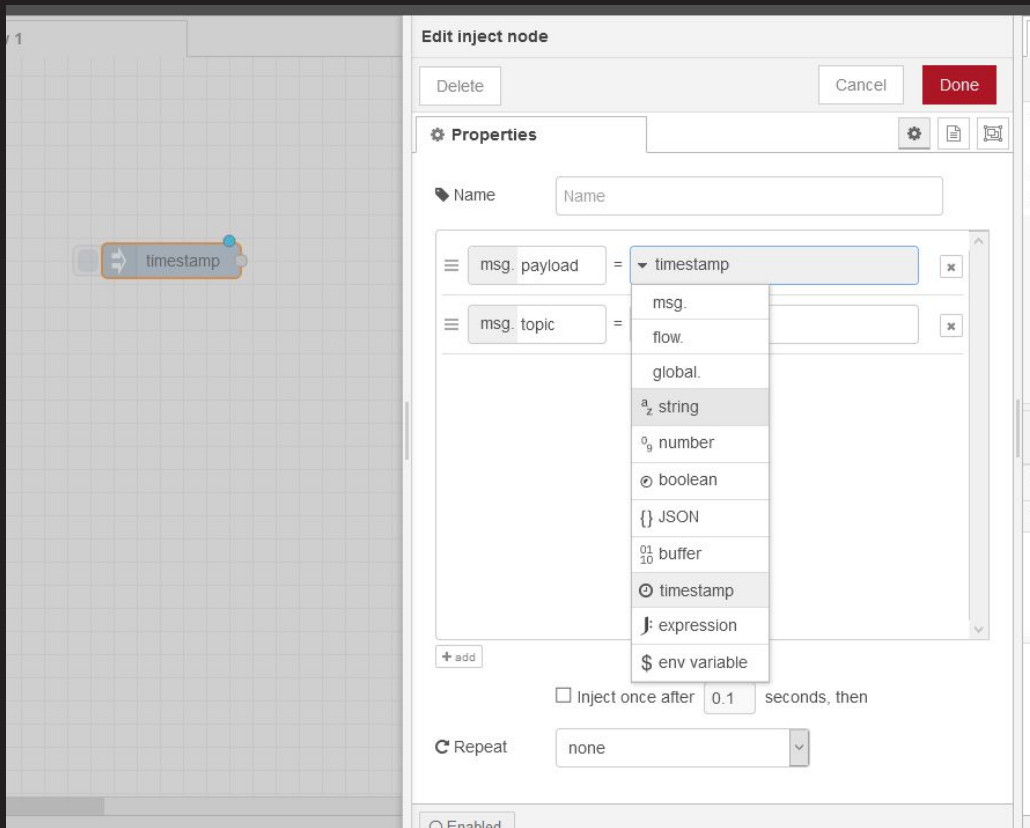
Type inject

show more

Pressing enter will edit the first node in the current selection

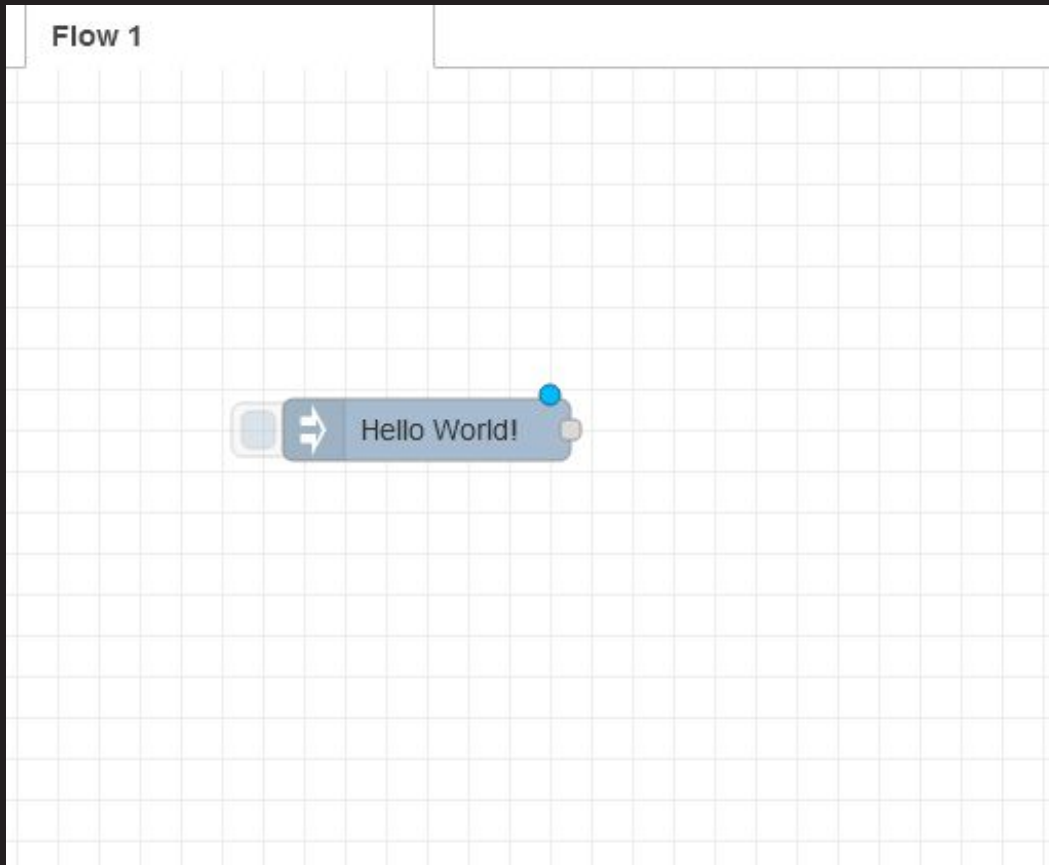
# Primeiro fluxo com Node-RED

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



# Primeiro fluxo com Node-RED

O resultado será:



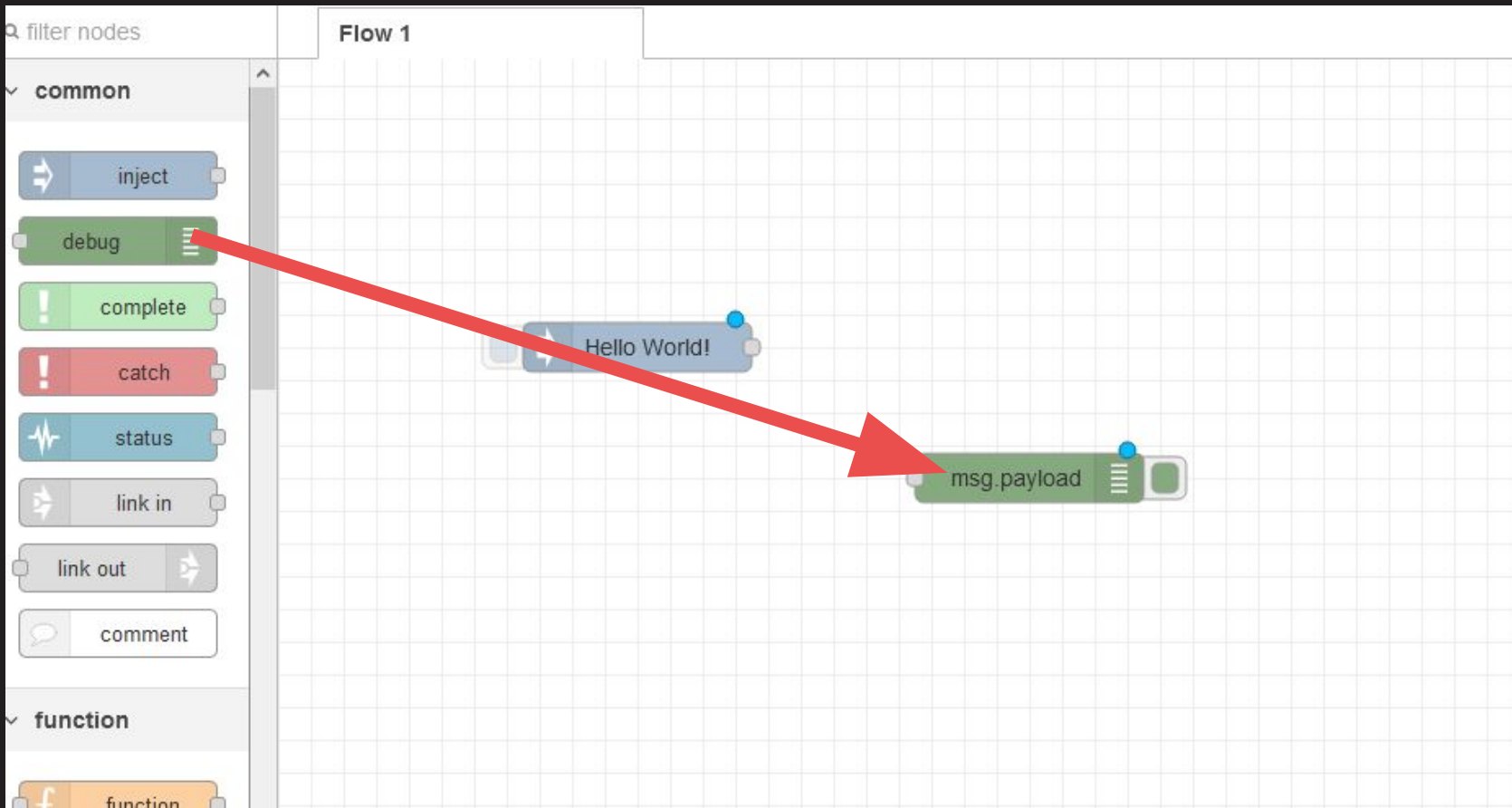
Este nós está “injetando” uma mensagem, do tipo string, com o valor “Hello World”;

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 que contêm um conteúdo (payload). Neste caso, nossa payload é uma string, ou seja, um pedaço de texto.

# Primeiro fluxo com Node-RED

Precisamos de um destino para a mensagem de Hello World.  
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 que ele funcionasse. “Debugar” é ver e tirar bugs do seu código.

# Primeiro fluxo com Node-RED

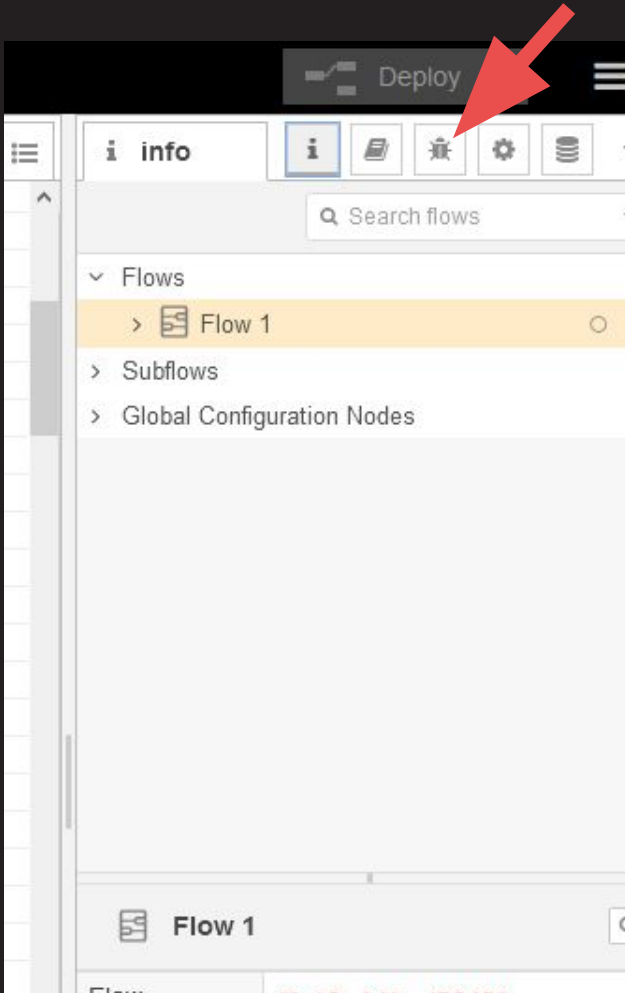
Agora conecte os dós nós.

E clique em deploy.

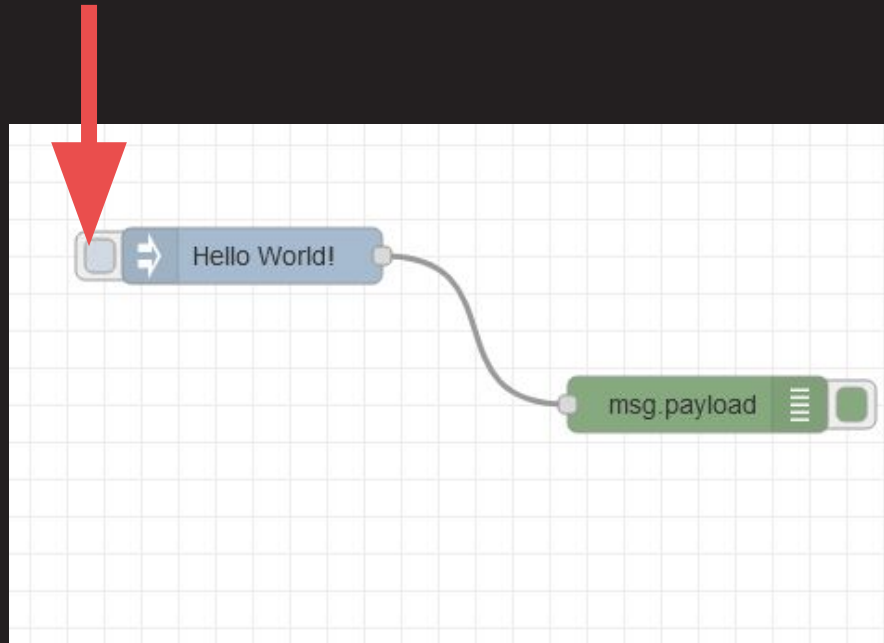
The screenshot shows the Node-RED web interface. In the center workspace, a flow named 'Flow 1' is visible. It consists of two nodes connected by a wire: a blue 'Hello World!' output node and a green 'msg.payload' input node. A red arrow points to the connection line between these two nodes. In the top right corner, there is a red 'Deploy' button with a refresh icon. Another red arrow points to this button. On the right side, there is a sidebar with a search bar and a list of flows, including 'Flow 1'.

# Primeiro fluxo com Node-RED

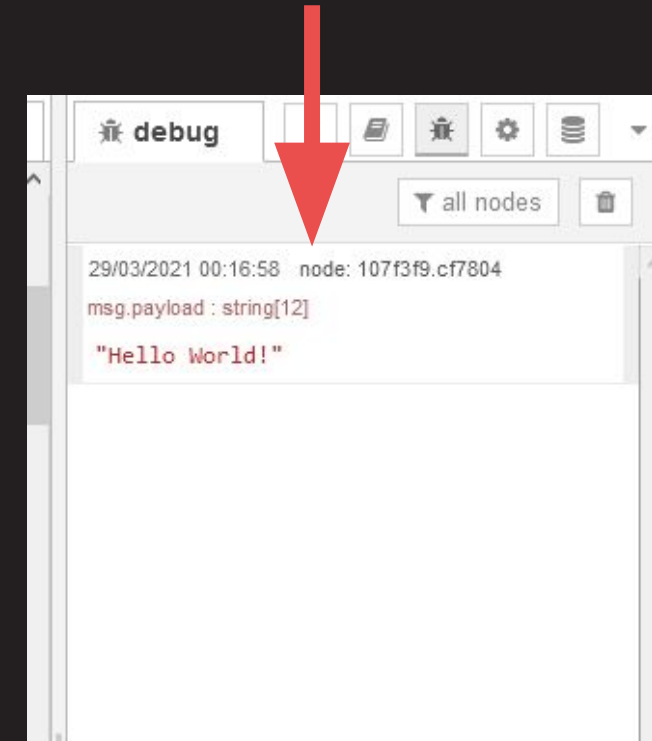
Abra o menu de debug



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



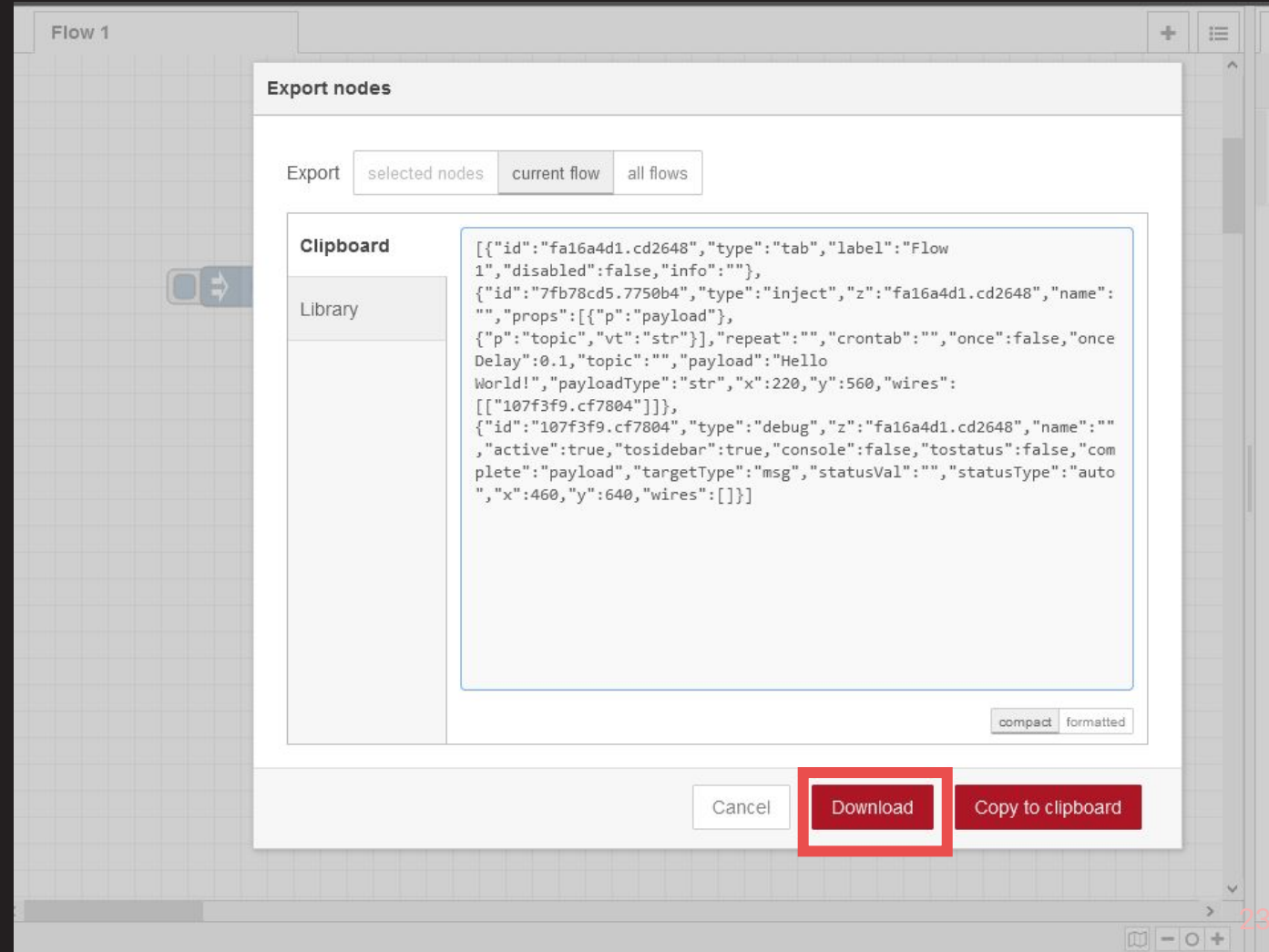
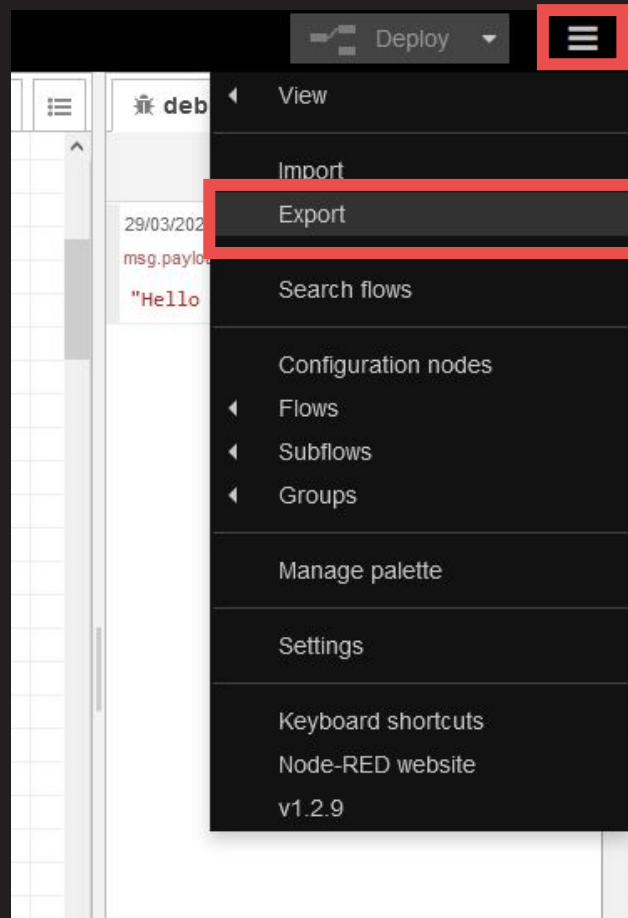
Resultado





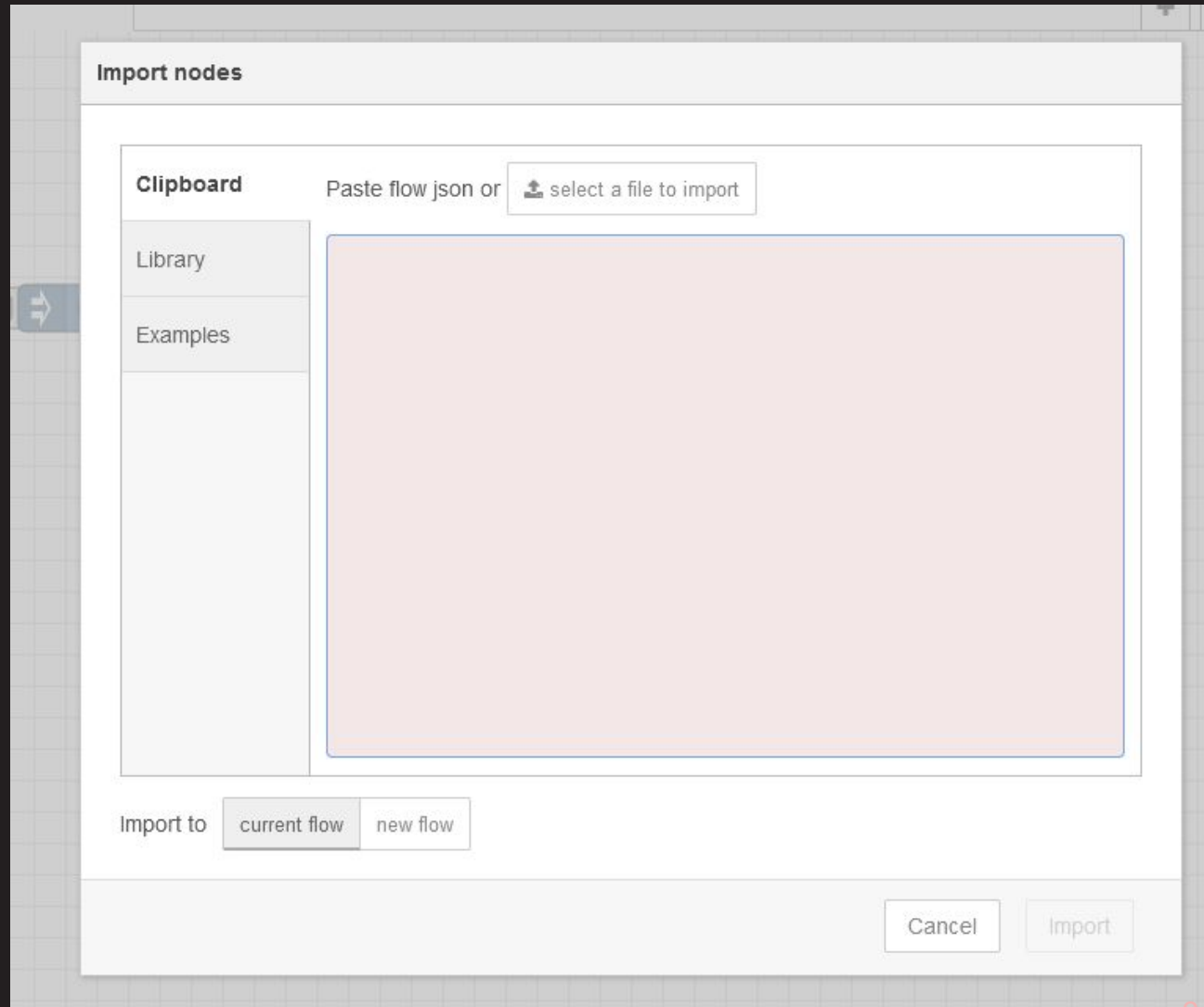
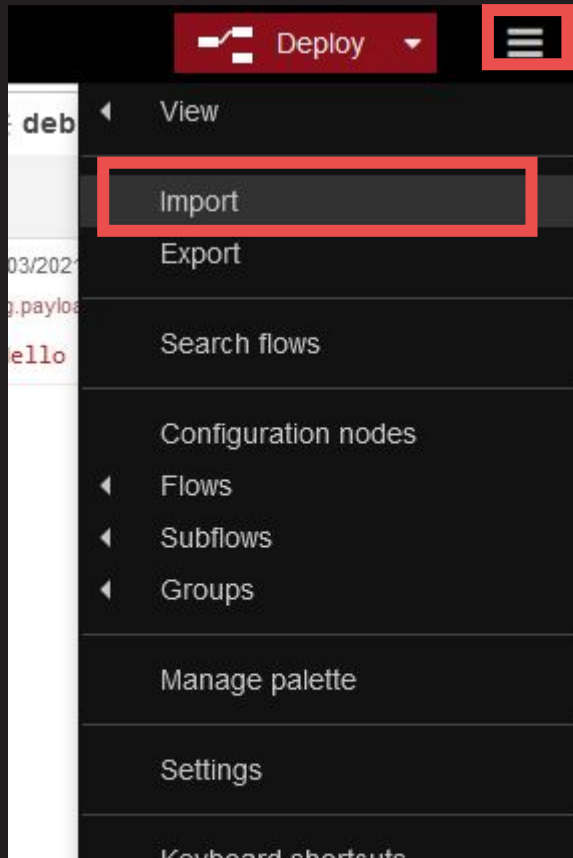
# Primeiro fluxo com Node-RED

Salvando um fluxo:



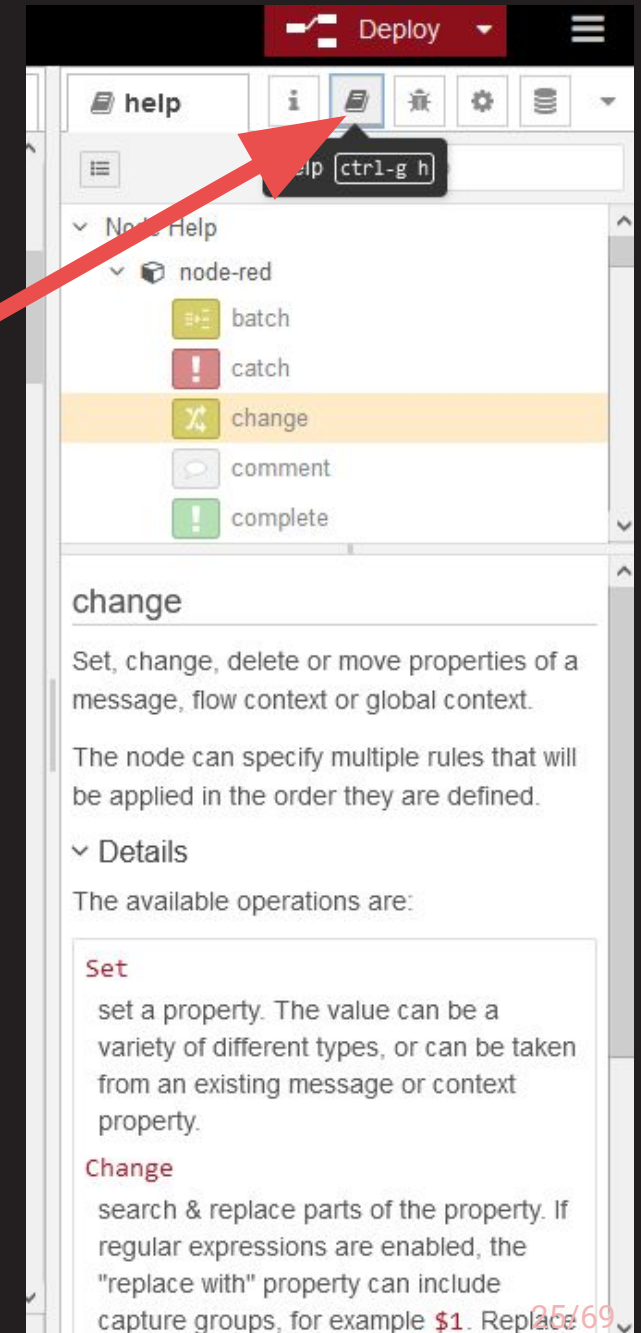
# Primeiro fluxo com Node-RED

Carregando um fluxo:



# Primeiro fluxo com Node-RED

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

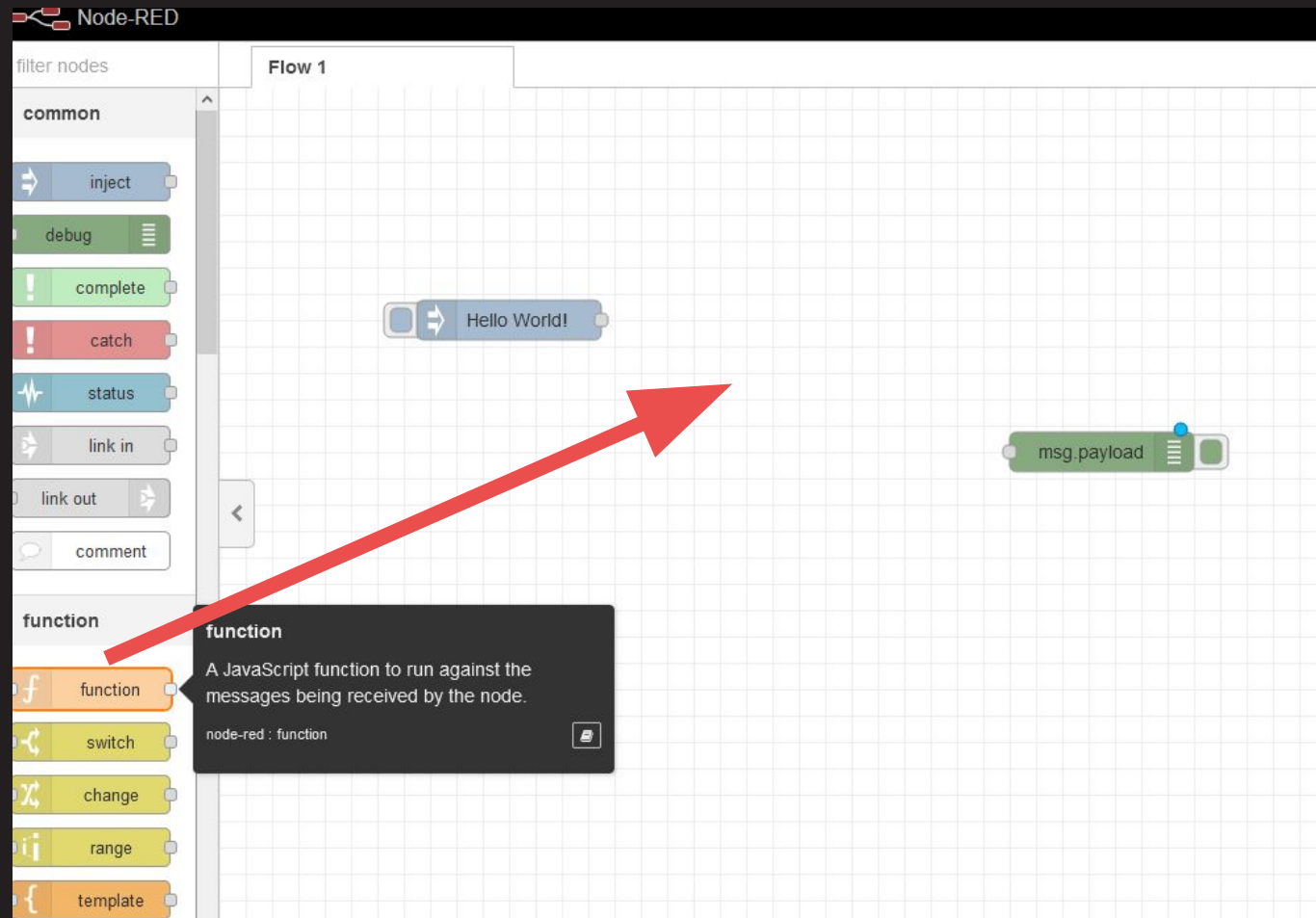


# Hello World 2.0

Manipulando a mensagem de Hello World

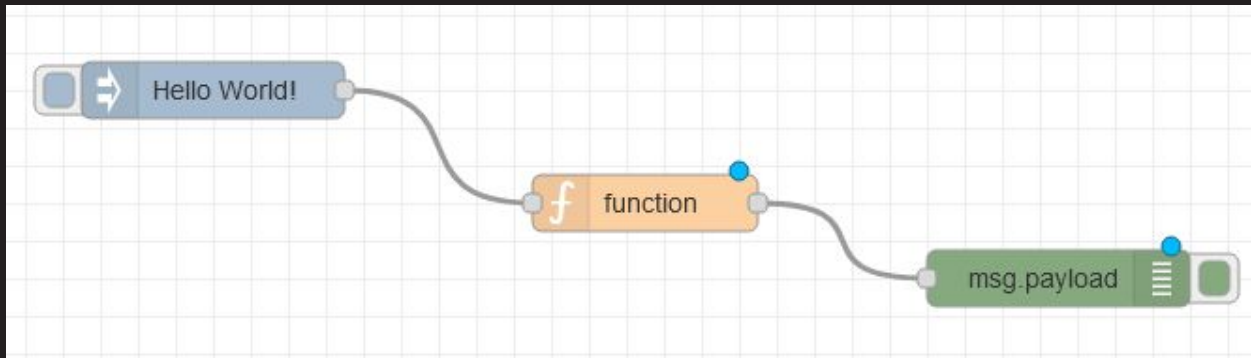
# Hello World 2.0

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

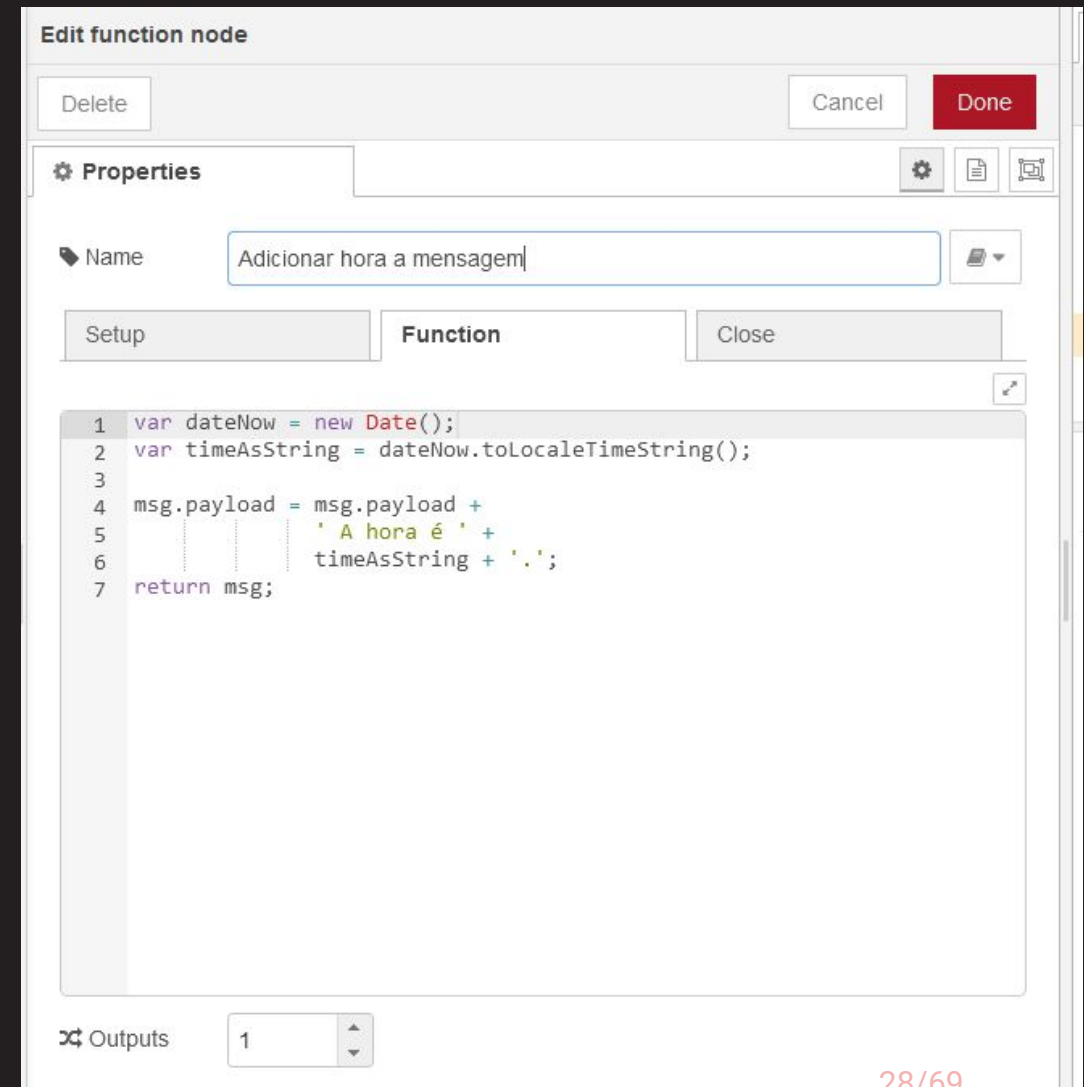


# Hello World 2.0

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.



# Hello World 2.0

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

The screenshot shows the Node-RED web interface. On the left, a workspace titled 'Flow 1' contains a flow with three nodes connected in sequence: a blue 'Hello World!' node, an orange function node labeled 'Adicionar hora a mensagem', and a green 'msg.payload' node. The 'Hello World!' node has a small square button with a right-pointing arrow. On the right, the 'debug' menu is open, showing a list of log entries. The first entry is '29/03/2021 00:16:58 node: 107f3f9.cf7804 msg.payload : string[12] \"Hello World!\"'. The second entry is '29/03/2021 00:38:04 node: 107f3f9.cf7804 msg.payload : string[30] \"Hello World! A hora é 0:38:03.\"'.



# Hello World 2.0

Vamos fazer ele repetir a injeção de mensagem , selecionando interval no campo Repeat, e configurando para 2 segundos.

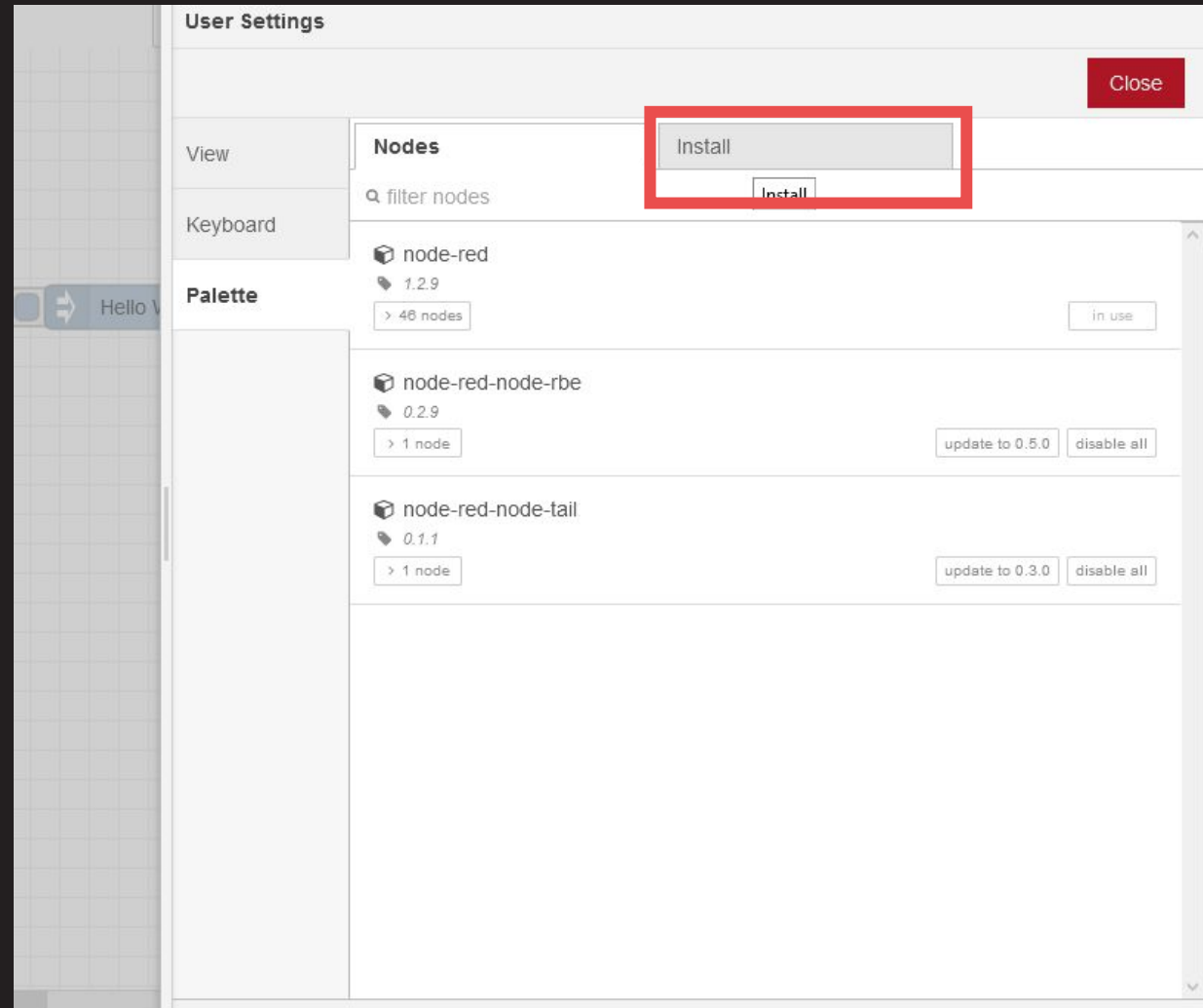
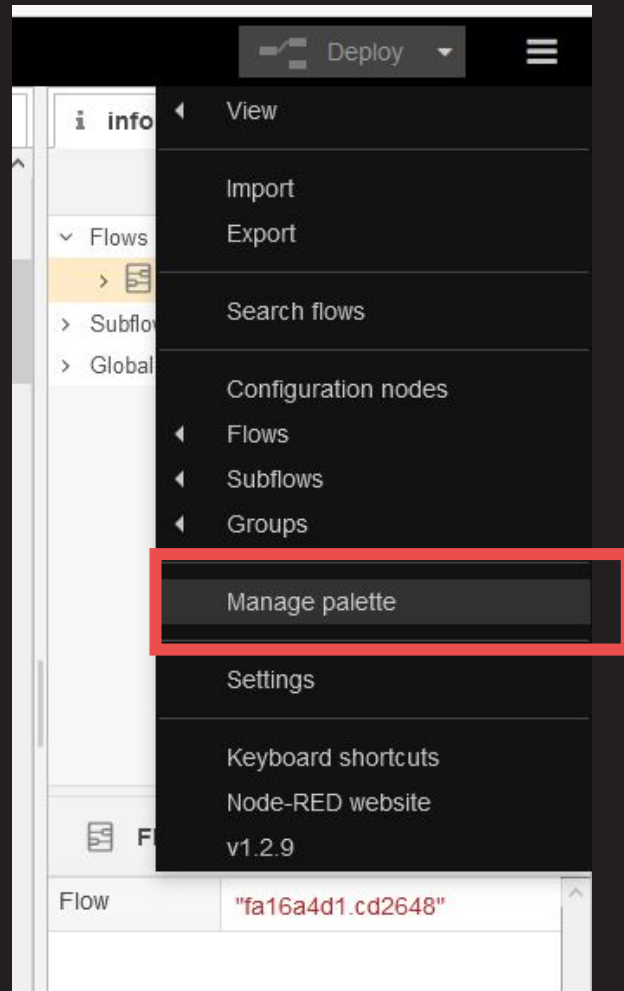
O que acontece agora?

The screenshot shows the 'Edit inject node' dialog box. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' tab with a 'Name' field. The main area contains a list of message properties: 'msg: payload' with value 'Hello World!' and 'msg: topic' with value 'a\_z'. At the bottom, there is a checkbox for 'Inject once after 0.1 seconds, then' and a 'Repeat' section. The 'Repeat' section has a dropdown set to 'interval' and a field set to 'every 2 seconds'.

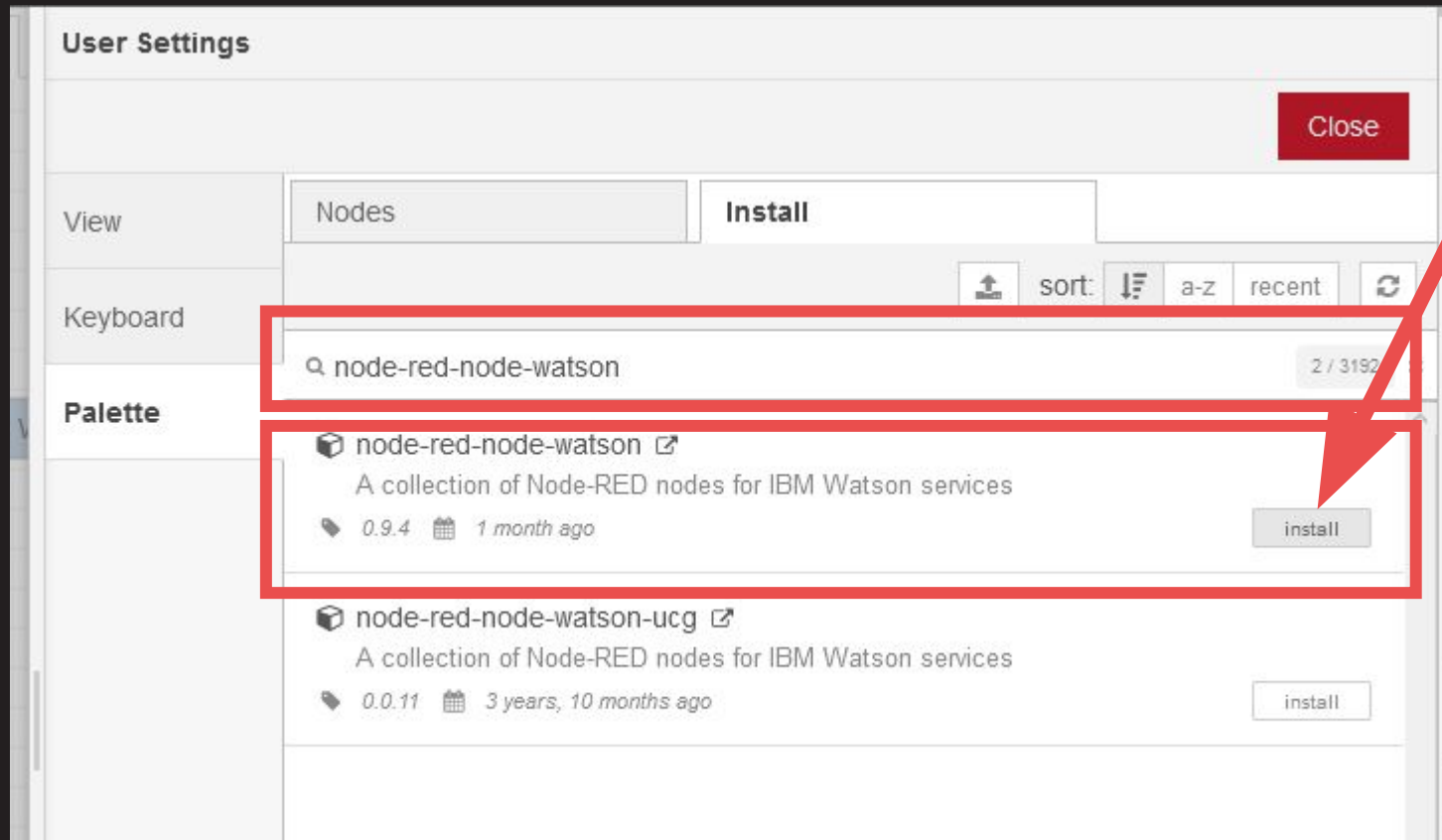
# Nós especiais e bibliotecas

Instalando a biblioteca de desenvolvimento da IBM Cloud e do Telegram

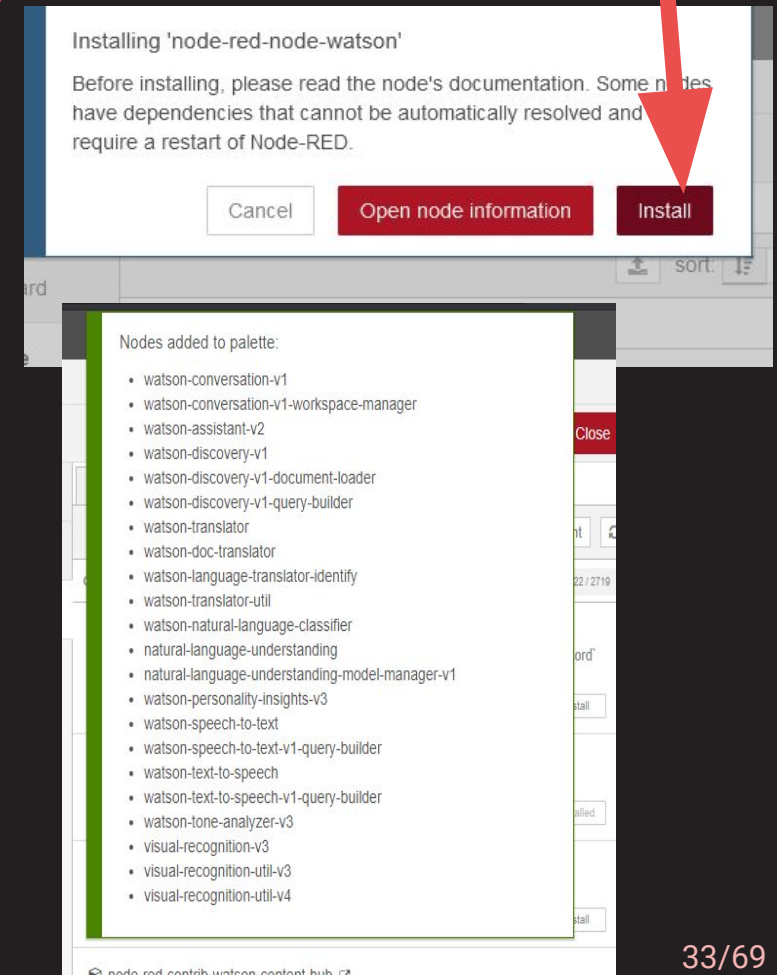
# Instalando a bibliotecas



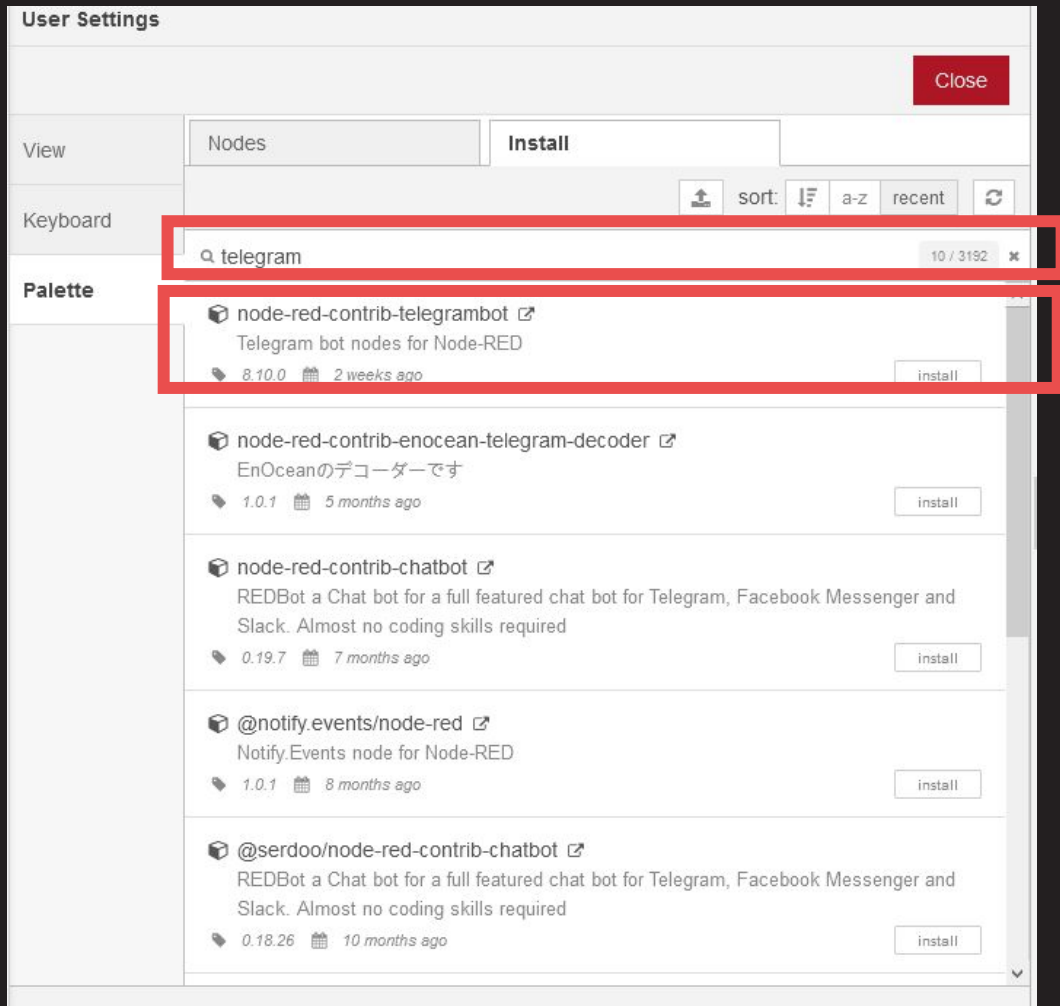
# Instalando a biblioteca do IBM Watson



## Instale o pacote



# Instalando a biblioteca do Telegram



## Nodes added to palette:

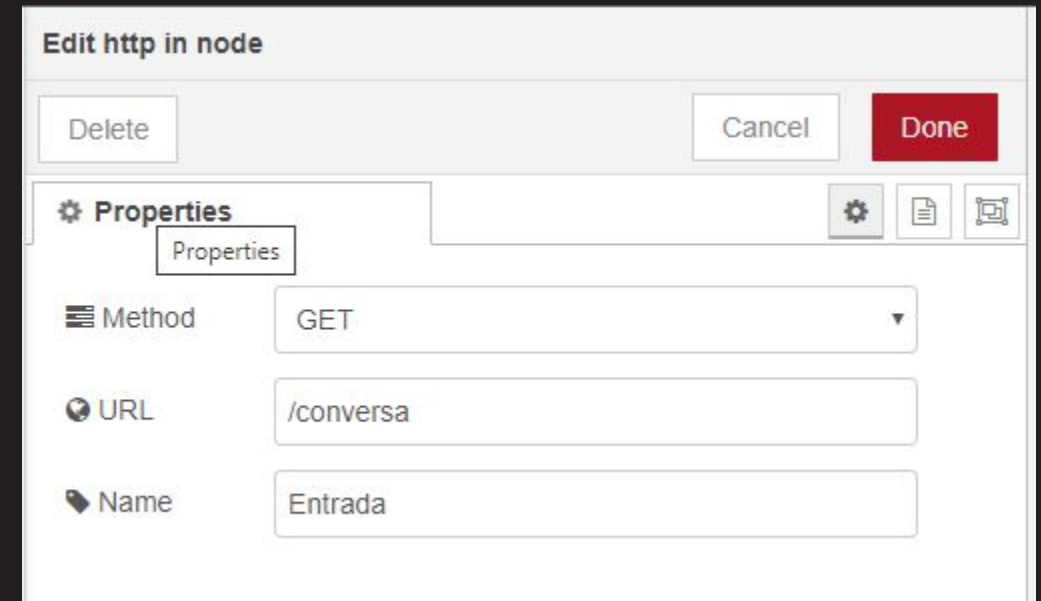
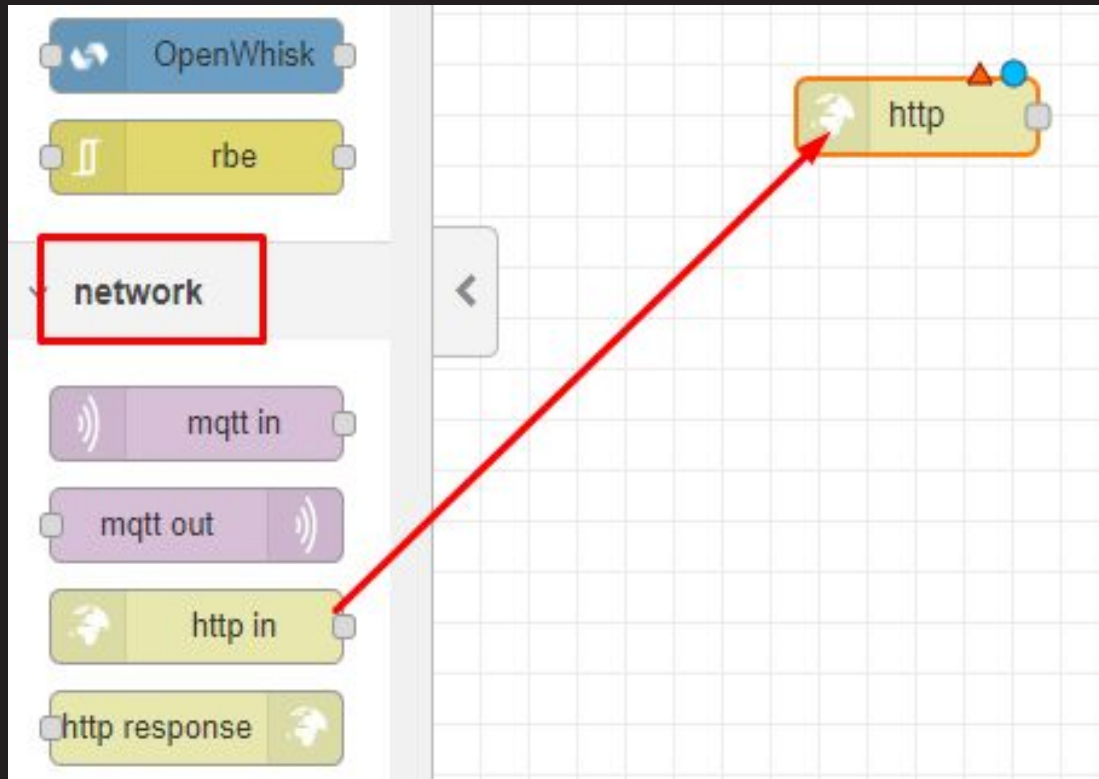
- telegram bot
- telegram receiver
- telegram command
- telegram event
- telegram sender
- telegram reply

# Conectando o Bot

Fazendo um fluxo para enviar e receber mensagens do Watson Assistant

# Configurando conexão com o Assistant de Vendas

Adicione um nó de **http in**. Depois dê um duplo clique e preencha as propriedades do nó como abaixo:

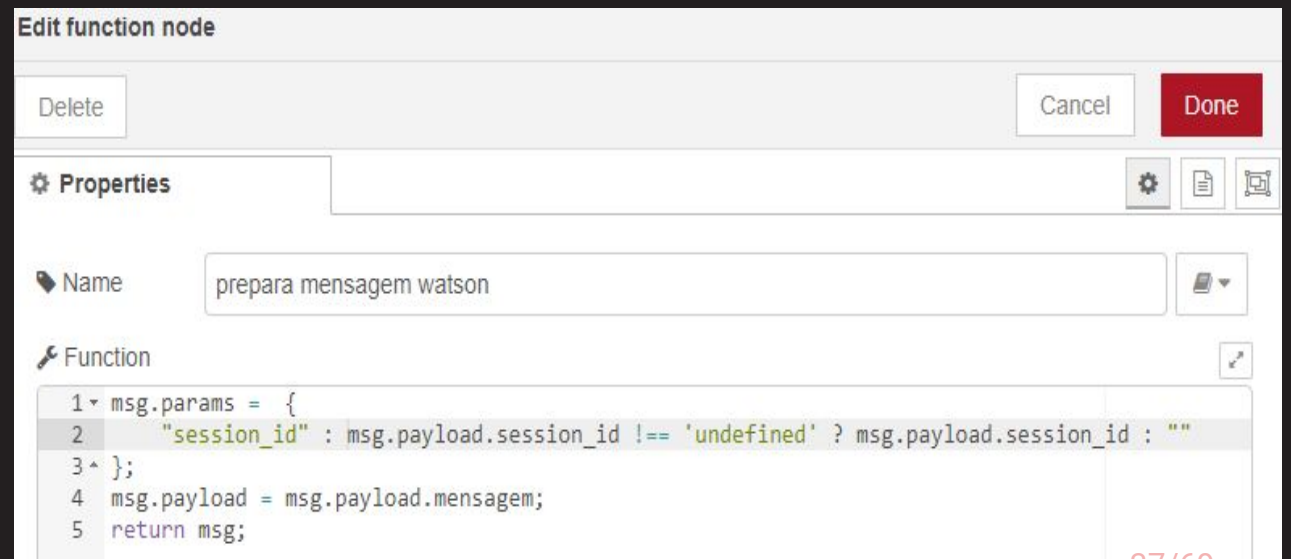




# Configurando conexão com o Assistant de Vendas

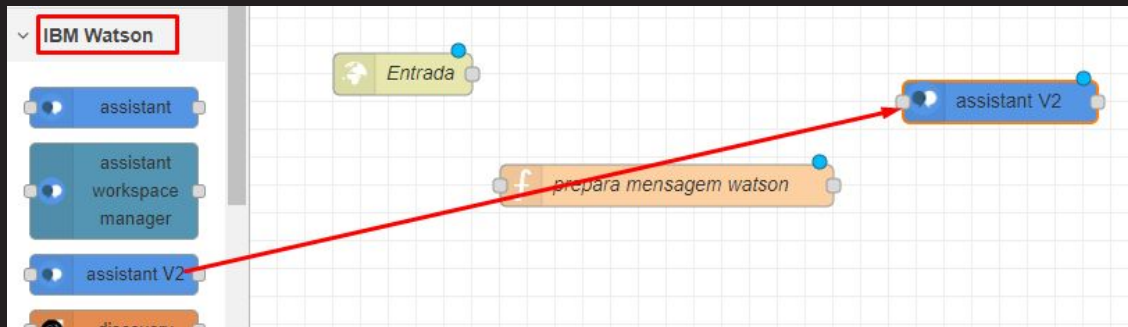
Adicione um nó de **function** com nome de Prepara Mensagem para o Watson. Nas propriedades, digite o seguinte código:

```
msg.params = {  
  "session_id" : msg.payload.session_id !== 'undefined' ? msg.payload.session_id : ""  
};  
msg.payload = msg.payload.mensagem;  
return msg;
```



# Configurando conexão com o Assistant de Vendas

Arraste e solte um nó de Assistant v2.



Precisamos preencher o campo **API Key** e **Assistant ID**. Estes valores são do seu serviço. Você precisará abrir a IBM Cloud para pegá-los.

Edit assistant V2 node

Delete Cancel Done

**Properties**

Name

Username

Password

API Key

Service Endpoint

Assistant ID

Timeout Period

☐ Switch on Debug

☐ Restart Dialog

☒ Return Context

☐ Return Alternate Intents

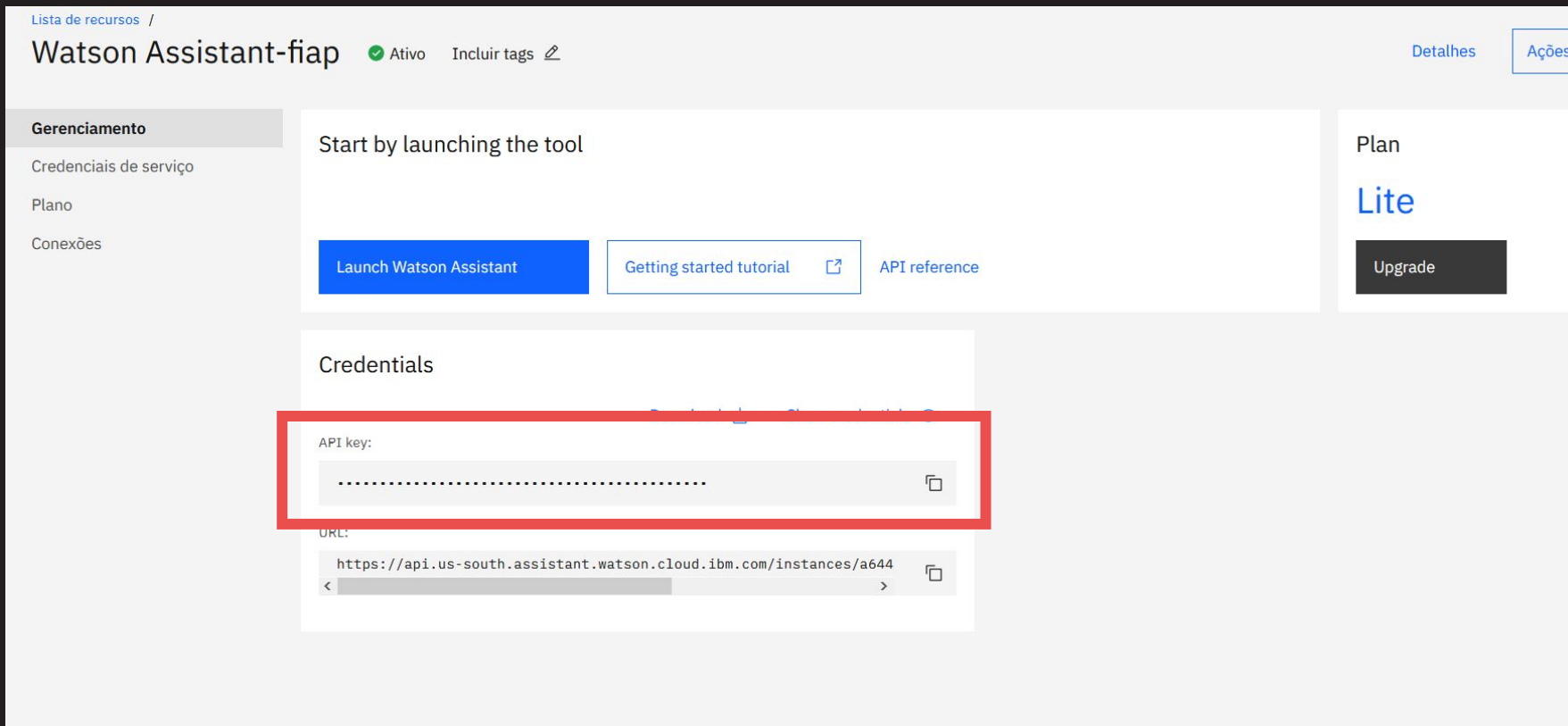
☒ Multiple Sessions

☐ Opt Out Request Logging

**Note:** When using multiple sessions, and `msg.params.session_id` is not set then a new session id is generated. See info box for details.

# Configurando conexão com o Assistant de Vendas

A API Key fica na página de serviço do Watson Assistant. Este número é a senha para acessar todos os seus bots. Guarde-o bem!



The screenshot shows the Watson Assistant service page for 'Watson Assistant-fiap'. The page is in Portuguese and includes a sidebar with navigation options: 'Gerenciamento', 'Credenciais de serviço', 'Plano', and 'Conexões'. The main content area has a 'Start by launching the tool' section with buttons for 'Launch Watson Assistant', 'Getting started tutorial', and 'API reference'. Below this is a 'Credentials' section with a red box highlighting the 'API key' field, which contains a masked value (dots) and a copy icon. The 'URL' field below it contains the URL 'https://api.us-south.assistant.watson.cloud.ibm.com/instances/a644' and also has a copy icon. On the right side, there is a 'Plan' section showing 'Lite' and an 'Upgrade' button.

Lista de recursos /

Watson Assistant-fiap Ativo Incluir tags

Detalhes Ações...

Gerenciamento

Credenciais de serviço

Plano

Conexões

Start by launching the tool

Launch Watson Assistant

Getting started tutorial

API reference

Plan

Lite

Upgrade

Credentials

API key:

.....

URL:

https://api.us-south.assistant.watson.cloud.ibm.com/instances/a644

# Configurando conexão com o Assistant de Vendas

Uma vez dentro do WA, clique nos três pontinhos no Assistant e selecione Settings:

## Assistants

An assistant helps your customers complete tasks and get information faster. It may clarify requests, search for answers from a knowledge base, and can also direct your customer to a human if needed.

Create assistant

### Vendas

Ajuda os clientes com eventuais dúvidas e indicando os melho...

#### Skills (1)

Ajuda

#### Integrations (1)



Rename

Settings

Delete

# Configurando conexão com o Assistant de Vendas

O ID do Assistant pode ser copiado agora:

Assistant settings

Vendas

API details

Webhooks

Inactivity timeout

API details

Assistant details

Assistant name:

Vendas

Assistant ID:

8e7878c6-d0d2-41c6-97ea-544c8fdf0170

Assistant URL:

https://api.us-south.assistant.watson.cloud.ibm.com/instances/a6440d84-f598-4

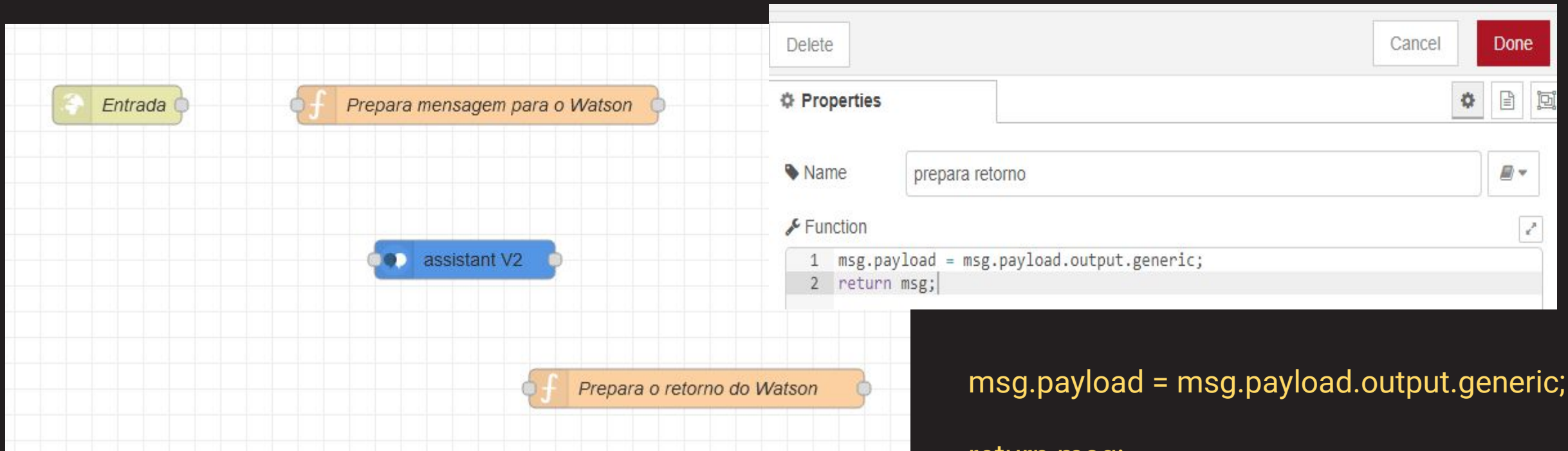
Service credentials

Credentials name:

Auto-generated service credentials

# Configurando conexão com o Assistant de Vendas

Adicione mais um nó de **function** para tratar os dados de retorno.



The screenshot displays the IBM Cloud Functions interface. On the left, a workflow is visible on a grid background, consisting of four nodes: a yellow 'Entrada' node, an orange 'Prepara mensagem para o Watson' node, a blue 'assistant V2' node, and another orange 'Prepara o retorno do Watson' node. On the right, a configuration panel for a function named 'prepara retorno' is shown. The 'Function' tab is active, displaying the following code:

```
1 msg.payload = msg.payload.output.generic;  
2 return msg;
```

Below the code, the same two lines are repeated in a larger font for emphasis:

```
msg.payload = msg.payload.output.generic;  
return msg;
```

# Configurando conexão com o Assistant de Vendas

Adicione um nó de http response e configure-o como na imagem:

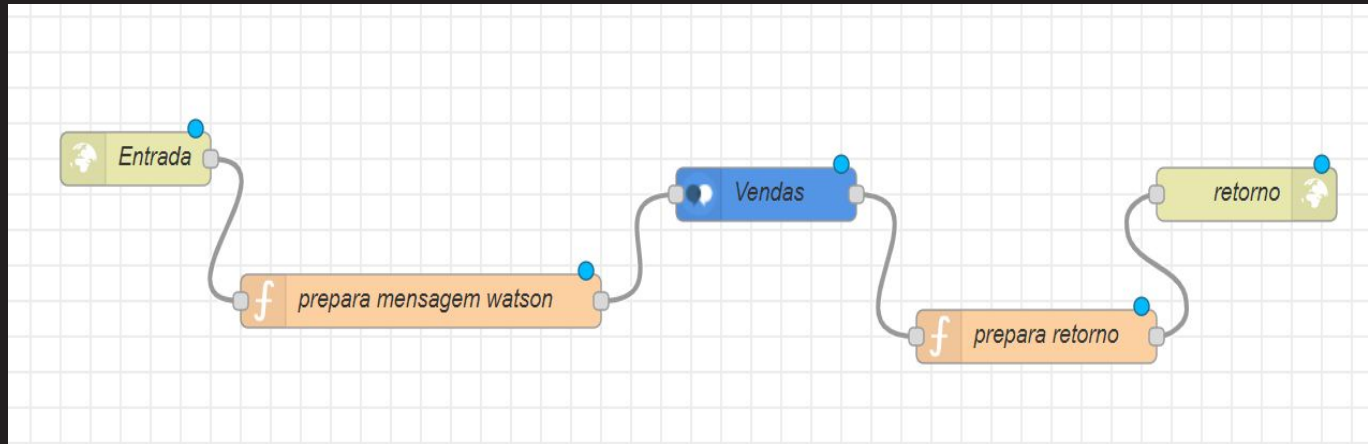
The image shows a Node-RED workspace with a palette on the left containing various nodes. A red arrow points from the 'http response' node in the palette to an 'http' node in the workspace. The 'http' node is connected to a 'Vendas' node and two function nodes labeled 'prepara mensagem watson' and 'prepara retorno'. A properties dialog is open for the 'http' node, showing the following configuration:

- Name:** retorno
- Status code:** msg.statusCode
- Headers:** Access-Control-Allow-Origin: \*

The 'Access-Control-Allow-Origin' header and its value '\*' are highlighted with red boxes in the original image.

# Configurando conexão com o Assistant de Vendas

Temos um integração pronta!



Para testar localmente, digite no navegador o IP local, seguido da porta do Node-RED / o endereço do nó de entrada.





# Próximos Passos

O que veremos na próxima aula

# Na próxima aula...

- Criando um bot no Telegram;
- Criando o fluxo Node-RED de integração com o Telegram;
- Subindo o fluxo de integração para Nuvem da IBM.

**Copyright © 2022**

**Slides do Prof. Henrique Ferreira, com adaptações dos  
slides dos Prof. Marcelo Grave e Andrey Masiero - FIAP**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).