

Node-RED

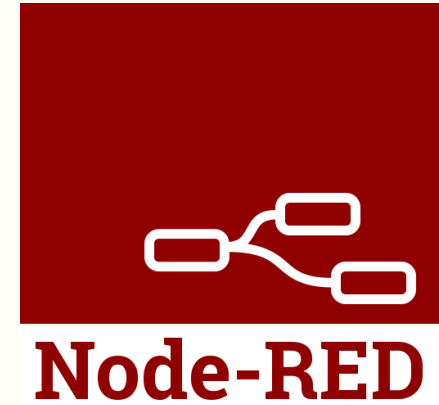


O que é o Node-RED?

Ferramenta de **fluxo** para programação visual.

desenvolvida pela IBM;

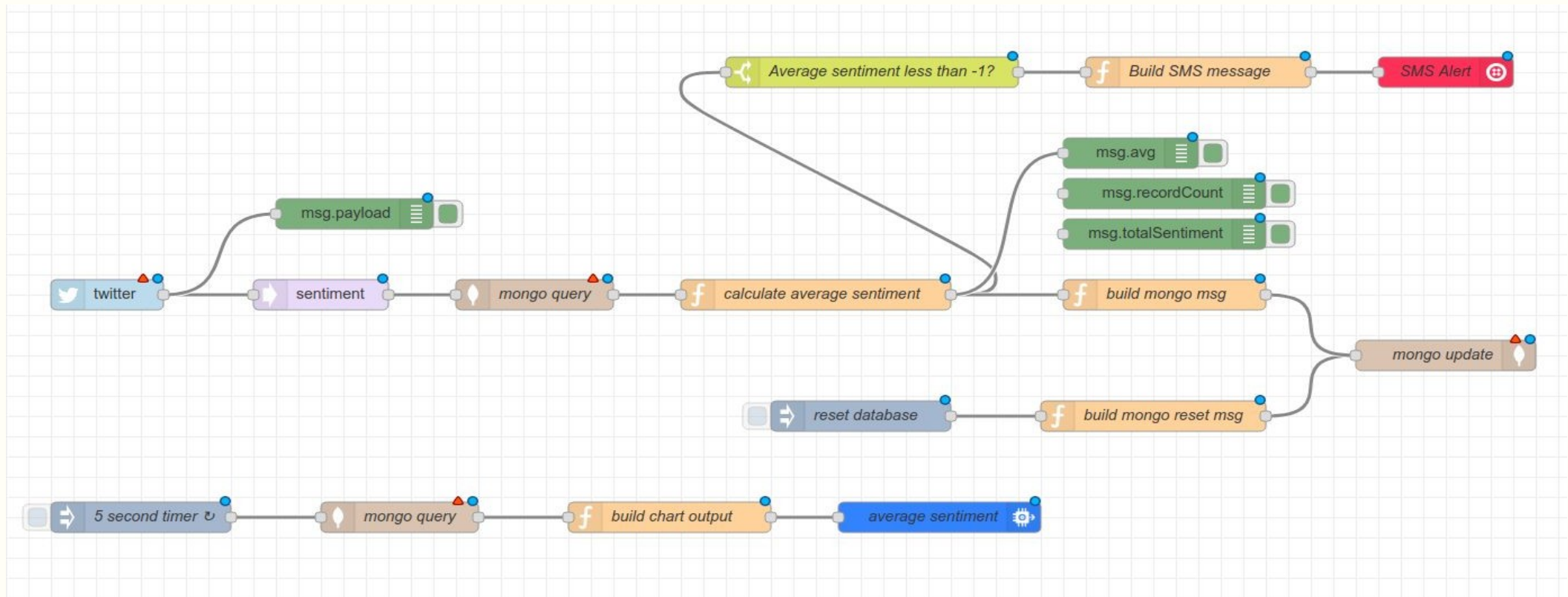
conectar dispositivos de hardware, APIs e serviços online como parte da Internet das Coisas (IoT).



Principais Características

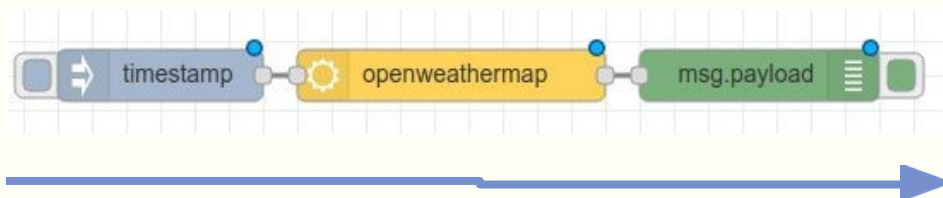
- Instalação simples (`sudo npm install -g --unsafe-perm node-red`)
- Vasta documentação
 - Tutoriais: <https://cookbook.nodered.org/>
 - Guia de usuário: <https://nodered.org/docs/user-guide/>
 - Tutorial para criação de nós: <https://nodered.org/docs/creating-nodes/>
 - Documentação da API: <https://nodered.org/docs/api/>
 - Tutorial para desenvolvedores colaboradores: <https://nodered.org/docs/developing/>

Fluxos



Fluxos

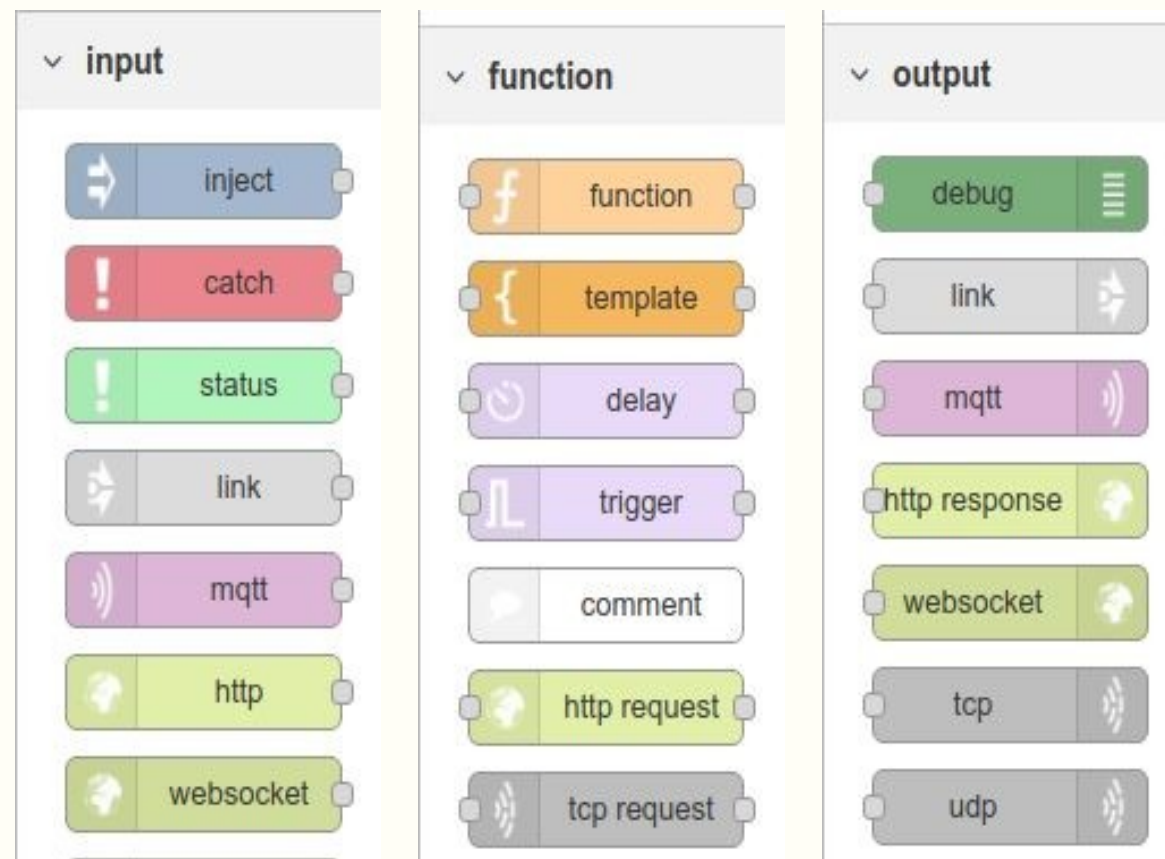
- No Node-RED, **programas** são fluxos compostos por uma coleção de nós conectados para trocar **mensagens**;
- Cada nó tem um propósito bem definido;
- Tecnicamente, um fluxo consiste em uma lista de objetos **JavaScript** que descrevem os nós e suas configurações.



```
[
  {
    "id": "31e10b93.612544",
    "type": "openweathermap",
    "z": "7dc53b7c-69de-45a4-b600-a41b4440979b",
    "wtype": "current",
    "city": "Goiabeiras",
    "country": "BR",
    "language": "pt",
```

NÓS

- 3 tipos básicos
- **Nós de entrada** - gera mensagens para nós de recebimento de dados.
- **Nós de processamento** - nós que processam dados de alguma forma, emitindo mensagens novas ou modificadas.
- **Nós de saída** - consome mensagens, por exemplo, para enviar dados para um serviço externo.
- 2 categorias adicionais
- **Nós de credenciais**
- **Nós criados por usuários**



Nós de Função

- Os nós de função são o “canivete suíço” do Node-RED;
- Podem ser utilizados quando não há um nó dedicado especificamente à tarefa que se deseja executar;
- São escritos em Javascript por meio de um editor de código disponível no Node-RED.



A interface "Edit function node" permite configurar um nó de função. No topo, há botões para "Delete", "Cancel" e "Done". Abaixo, a seção "node properties" contém:

- Name:** Um campo de texto com o valor "Converts to uppercase".
- Function:** Um editor de código com o seguinte conteúdo:

```
1 msg.payload = msg.payload.toUpperCase();  
2 return msg;
```
- Outputs:** Um seletor com o valor "1".

Na base, há uma dica amarela que diz: "See the Info tab for help writing functions."

Interface Node-RED

The screenshot displays the Node-RED web interface. On the left, the 'Paleta de nós' (Node Palette) is visible, containing various input nodes like inject, catch, status, link, mqtt, http, websocket, tcp, and udp. The central 'Editor de fluxo' (Flow Editor) shows two active flows: 'Monitoramento de pacientes' and 'Glucose Simulator'. Each flow consists of a simulator node connected to a 'json' node, then a 'Schema converter' node, and finally a function node. On the right, the 'Menu de' (Menu) bar includes a 'Deploy' button and a hamburger menu icon. Below the menu bar is the 'Informações e debug' (Information and Debug) sidebar, which displays project information (Project: Teste, Flow ID: "7dc53b7c-69de-45a4-b600-a41b4440979b", Name: Monitoramento de pacientes crônicos, Status: Enabled) and a flow description section.

Node-RED

Menu de

filter nodes

input

inject

catch

status

link

mqtt

http

websocket

tcp

udp

Paleta de nós

Monitoramento de pacientes

Blood pressure Simulator

json

Schema converter

Blood Pressure C

Glucose Simulator

json

Schema converter

Glucose Class

Editor de fluxo

Deploy

info

Information

Project

Teste

Flow

"7dc53b7c-69de-45a4-b600-a41b4440979b"

Name

Monitoramento de pacientes crônicos

Status

Enabled

Flow Description

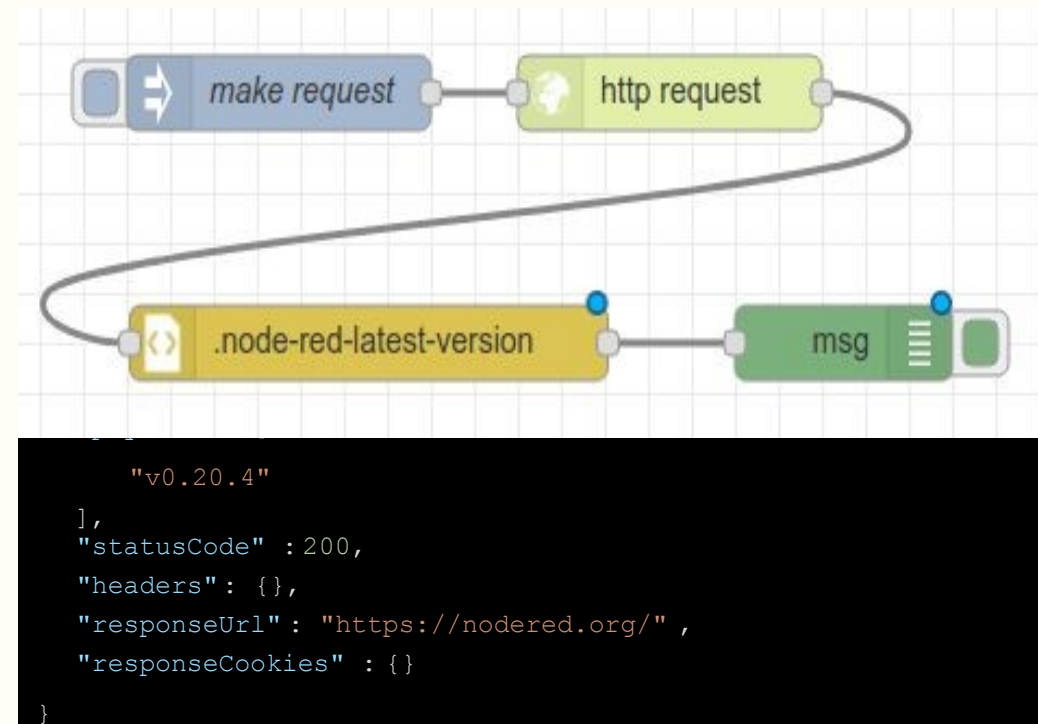
None

You can confirm your changes in the node edit tray with `ctrl-enter` or cancel them with `ctrl-escape`

Informações e debug

Mensagens

- As mensagens transmitidas entre nós no Node-RED são, por convenção, objetos JavaScript chamados **msg**, consistindo em um conjunto de propriedades nomeadas;
- Geralmente contêm uma propriedade **msg.payload** com a carga útil (*payload*) da mensagem;
- Os nós podem anexar outras propriedades a uma mensagem, que pode ser usada para transportar outras informações para o próximo nó no fluxo.

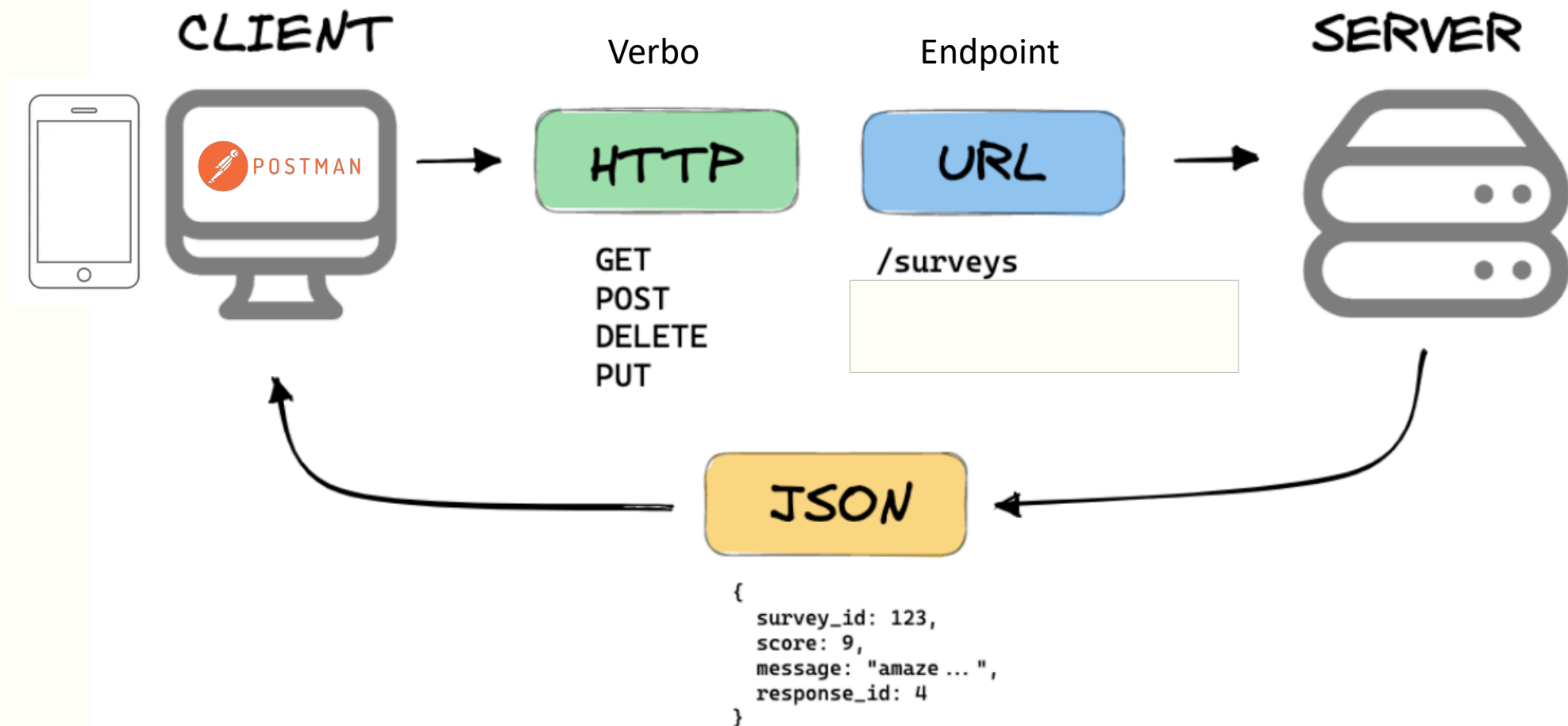


JSON

- JSON -> JavaScript Object Notation;
- Utilizado para representar um objeto JavaScript como String;
- É comumente usado por APIs da web para retornar dados;
- Elementos básicos do JSON:
 - { e } - delimita um objeto;
 - [e] - delimita um array;
 - : - separa chaves (atributos) de valores;
 - , - separa os atributos chave/valor.

JSON

- Os tipos de dados básicos do JSON são:
 - **string** - separados por aspas (duplas ou simples). Ex. "Brasil" ou 'Brasil';
 - **número** - sem aspas e pode ser **inteiro** ou **real**. Ex. 1 (inteiro) ou 23.454 (real);
 - **booleano** - tipo lógico normal, pode assumir valores **true** ou **false**;
 - **nulo** - valor para representar nulo. Ex. { "nome" : null }
 - **object**: É um conjunto de pares nome/valor;
 - **array**: utilizados para elementos ordenados.





/hackatruck



@hackatruck_makerspace



HACKATRUCK MAKERSPACE



www.hackatruck.com.br