



Material de Apoio módulo 6

Node-RED é uma plataforma de código aberto que oferece uma maneira visual de conectar dispositivos e sistemas, criando fluxos de automação e integração. Ele é amplamente usado na Internet das Coisas (IoT) e em projetos de automação, permitindo que desenvolvedores e entusiastas criem rapidamente aplicações e automações sem a necessidade de codificar tudo do zero.

O Node-RED é usado em diversas aplicações, incluindo:

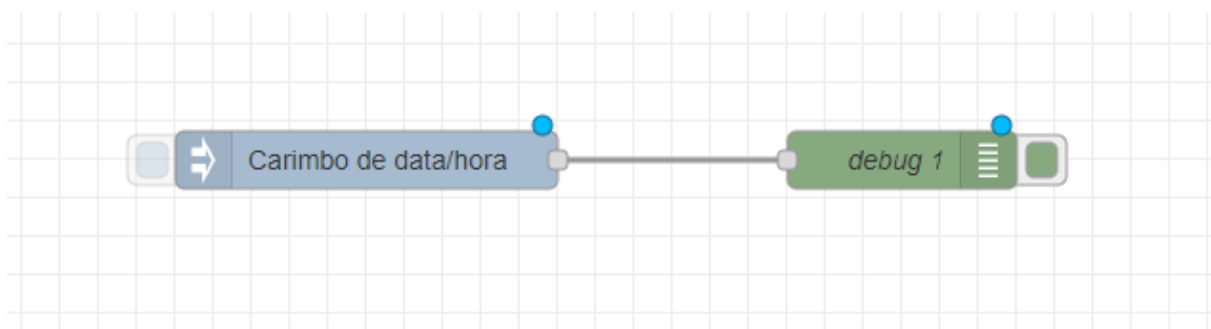
1. **Automatização Residencial Inteligente:** Controle de iluminação, termostatos, dispositivos IoT e sistemas de segurança em casa.
2. **Monitoramento e Controle Industrial:** Monitoramento em tempo real de máquinas e processos industriais, integração de sistemas SCADA e automação de processos.
3. **IoT e Sensores:** Coleta e processamento de dados de sensores IoT para análise em tempo real.
4. **Integração de APIs e Serviços Web:** Conexão de serviços web e aplicativos para automatizar fluxos de trabalho e integrações.
5. **Prototipagem Rápida:** Desenvolvimento rápido de protótipos e provas de conceito para projetos IoT e automação.

Funcionamento



Depois de instalado, execute o com o comando `node-red` no *powershell*. Acesse seu projeto por meio da URL `localhost:1880`

Seu projeto é chamado de fluxo. Dentro do fluxo, é possível adicionar nós que executam comandos. O fluxo precisa de uma entrada de dados que pode ser simulada através do nó **inject**, os dados podem ser visualizados no console do node-red pelo nó **debug**.



No exemplo acima, ao clicar no node inject será gerado uma leitura de timestamp que será enviada ao debug. O debug vai mostrar estes dados no console do node-red.

O debug é muito util para monitorar as mensagens que estão transitando entre os nodes. Ele acessa o objeto **msg** e mostra tudo o que está sendo gerado em determinado ponto.

O objeto msg

Objetos em JavaScript são extremamente versáteis. Se trata de coleções de pares chave-valor, onde cada chave (também chama de propriedade) está associada a um valor.

```
var pessoa = {  
  nome: "João",  
  idade: 30,  
  profissao: "Desenvolvedor"  
};
```

Neste exemplo, `pessoa` é um objeto que possui três propriedades: `nome`, `idade` e `profissao`, cada uma com seu próprio valor. Você pode acessar as propriedades de um objeto usando a notação de ponto ou a notação de colchetes:

```
console.log(pessoa.nome); // Saída: "João"
console.log(pessoa['idade']); // Saída: 30
```

Além disso, você pode adicionar, modificar e remover propriedades de um objeto dinamicamente:

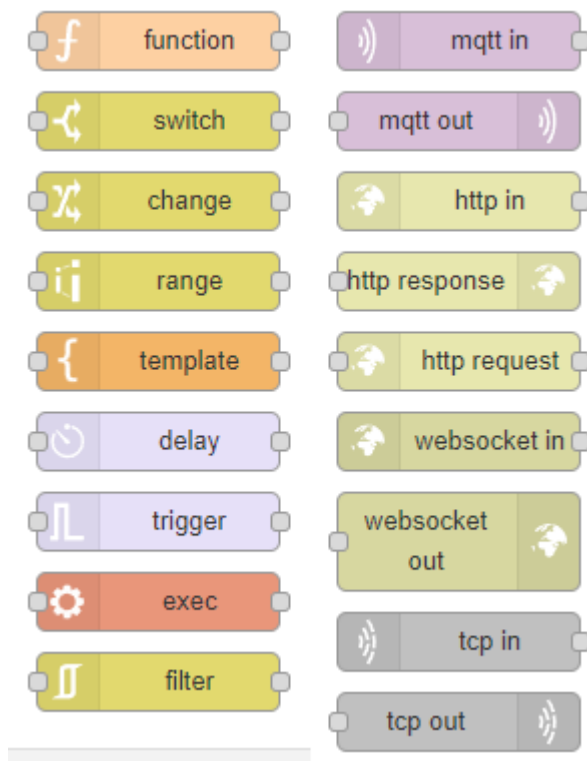
```
pessoa.cidade = "São Paulo"; // Adiciona uma nova propriedade
pessoa.profissao = "Designer"; // Modifica o valor de uma propriedade
delete pessoa.idade; // Remove a propriedade 'idade'
```

No node-red, existe o objeto chamado `msg` que contém a propriedade `payload` que carrega a maioria dos dados. Mas você pode criar novas propriedades por meio de um nó function, por exemplo:

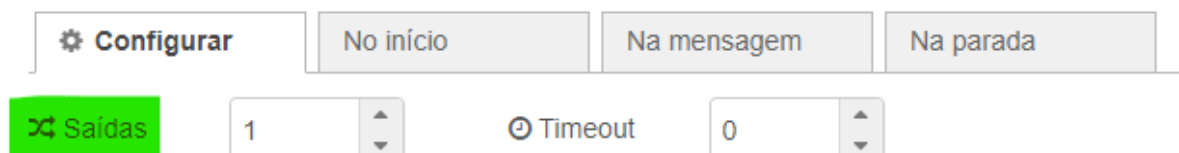
```
1  /* Por padrão, msg contem as propriedades:
2     _msgid
3     payload
4     topic
5  */
6
7  // Voce pode adicionar mais propriedades a ele
8  msg.nome_do_sensor = "ESP32";
9  msg.timestamp = 1698409591836;
10
11  return msg;
12
```

Node function

Perceba que alguns nodes contém dois canais de conexão (entrada e saída) como os nodes function, switch e change, e outros possuem apenas um, como mqtt in e mqtt out.



Porém, o node function é capaz de gerar várias saídas. Para isto, vá na aba configurar do node e defina a quantidade de saídas desejadas



E no código, retorne um array:

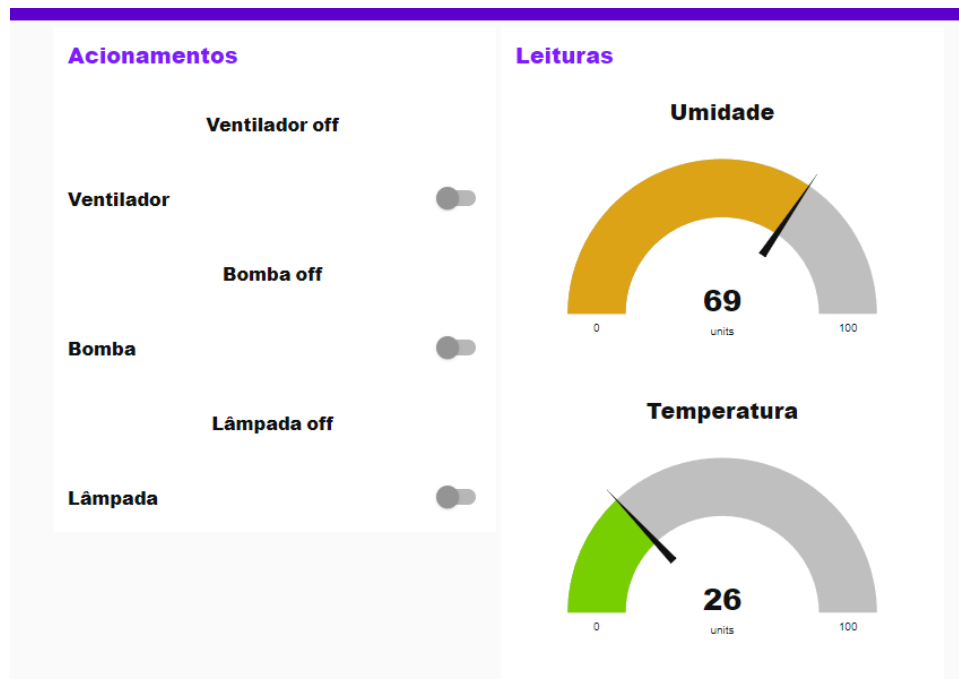
```

11  const msg2 = {
12    _msgid: 5,
13    payload: 255,
14    topic: "leituraAdc"
15  }
16
17  // Exemplo de uma função com 3 saídas
18  return [msg, msg2, null]
19

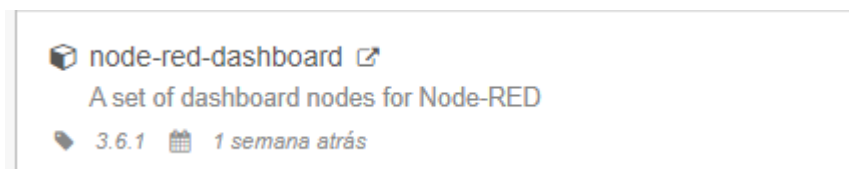
```

Dashboards

Com este plugin, é possível criar interfaces front-end amigáveis ao usuário com pouquíssimo trabalho.



Instale o dashboard na aba *manage pallets*.

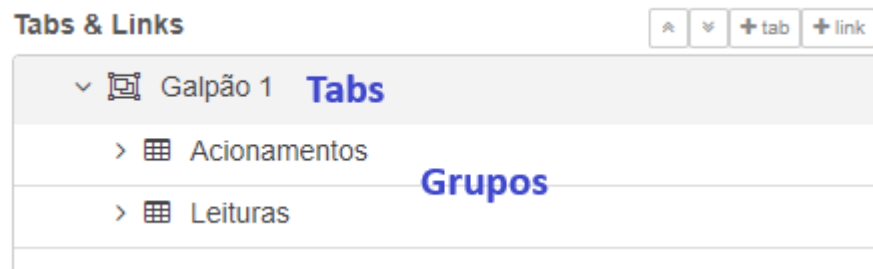


Com o dashboard instalado, são disponibilizados vários elementos de interfaces gráficas comuns ao universo IoT: botões, caixas de texto, gráficos, medidores, formulários, entre outros.

Ao adicionar eles em sua interface, eles funcionarão como blocos. Mas antes é necessário configurar as tabs e os grupos.

- Tabs → São as páginas de sua aplicação. Caso tenha necessidade de criar interfaces diferentes para uma mesma aplicação, crie outra tab. Elas ficarão separadas por um menu no canto esquerdo.
- Grupos → São os conjuntos de elementos na interface. É dentro do grupo que você irá adicionar os botões, gráficos, etc. Estes elementos visuais serão dispostos um abaixo do

outro. Se você decidir criar outro grupo, este grupo ficará ao lado do primeiro.



Comando gerar senha

```
node-red admin hash-pw
```

Copie o hash gerado para settings.js, defina um usuário.

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
    password: "$2b$08$1bxoqPVKZky2eZoIlhxLH0RR2.P/b7epHljkkfUgbWf/2Qny3As0a",  
    permissions: "*"   
  }]  
},
```