

Documentação sobre o projeto de Automação 2024

Acesso de dados em tempo real via Node-Red

Instalação do node-red

O node-red pode ser instalado a partir de <https://nodered.org/docs/getting-started/windows> (para windows) ou <https://nodered.org/docs/getting-started/local> (para linux).

Uma vez instalado o node-red, é preciso executar no terminal de acordo com as instruções para cada ambiente.

Acesso ao node-red

A porta padrão para acessar o ambiente do node-red é 1880, no caso, digite <http://localhost:1880/>. Caso queira acessar o dashboard (após instalar a biblioteca), digite: <http://localhost:1880/ui/>.

Instalação de bibliotecas

Para realizar a instalação de uma biblioteca no node-red basta ir em “manage palette” no menu aberto ao clicar-se no canto superior direito, após isso, clique em “install”, depois, é necessário procurar pelo nome exato da biblioteca.

Informações do broker

Broker MQTT V3.1.1

IP: 143.107.102.8

Port: 1883

Username: garsoft-java

Senha: garsoft-java

Tópico: garsoft/dev/je05/dados

Sem TLS.

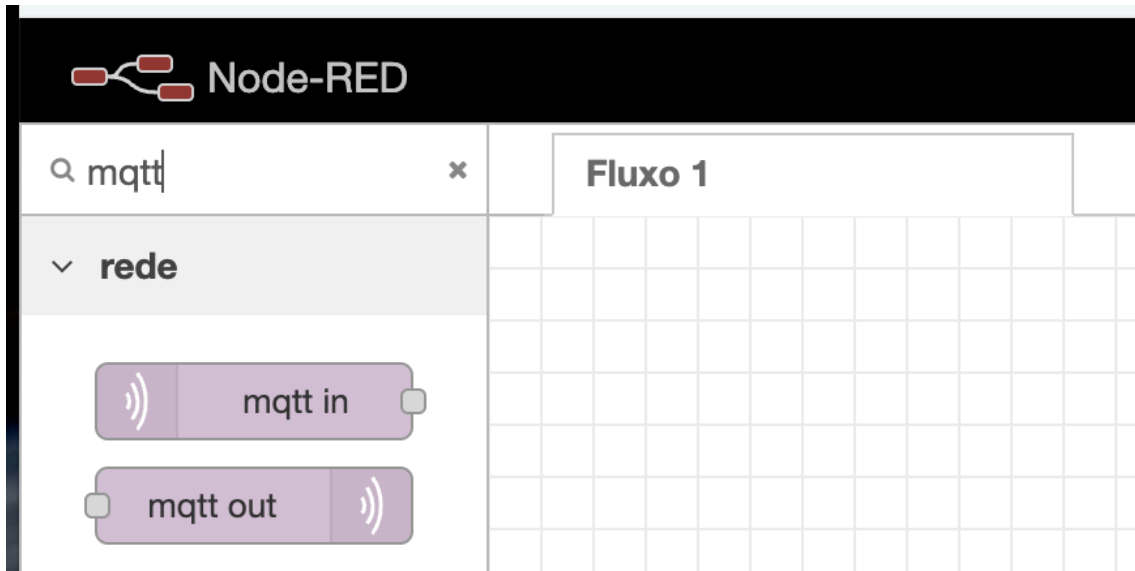
Princípios básicos do node-red

O Node-RED é uma plataforma de código aberto para programação visual de IoT (Internet das Coisas), integração de sistemas e automação de fluxos de trabalho.

O projeto utiliza a biblioteca “node-red-dashboard” para construção do dashboard, as bibliotecas podem ser vistas na aba “manage pallette” ao clicar-se no ícone do canto superior direito.

O primeiro bloco utilizado é o “mqtt in” que estabelece conexão com um broker MQTT, esse bloco está presente na aba “network”:

Bloco “MQTT in” sem configuração.



Fonte: Autor.

Após arrastar o bloco para a área quadriculada, deve-se configurar o broker. Para isso é necessário digitar o endereço IP do servidor e a porta correspondente. Além disso, pode-se controlar sobre quantos e quais tópicos o “client” MQTT receberá. Pode-se configurar também o “Quality of Service” da conexão MQTT e o formato de saída dos dados recebidos do broker.

Configuração do bloco “MQTT in”

Editar mqtt in nó

Deletar

Cancelar

Feito

Propriedades

Servidor

Adicionar novo mqtt-broker...

Ação

Assinar um tópico único

Tópico

Tópico

QoS

2

Saída

auto-deteção(objeto JSON, cadeia de caracteres ou armazenamento temporário anal

Nome

Nome

Editar mqtt in nó > Adicionar novo mqtt-broker configuração de nó

Cancelar

Adicionar

Propriedades

Nome

Labsoftlot

Conexão

Segurança

Mensagens

Servidor

143.107.102.8

Porta

1883

☒ Conectar automaticamente

☐ Usar TLS

Protocolo

MQTT V3.1.1

ID do cliente

Deixe em branco para geração automática

Mantenha-se vivo

60

Sessão

☒ Usar sessão limpa

Editar mqtt in nó > **Adicionar novo mqtt-broker configuração de nó**

Cancelar Adicionar

Propriedades

Nome Labsoftlot

Conexão Segurança Mensagens

Nome de usuário garsoft-java

Senha

Editar mqtt in nó

Deletar Cancelar Feito

Propriedades

Servidor Labsoftlot

Ação Assinar um tópico único

Tópico garsoft/dev/je05/dados

QoS 2

Saída auto-deteção(objeto JSON, cadeia de caracteres ou armazenamento temporário anal)

Nome Dados JE05

Fonte: Autor.

Ademais, há outras configurações a serem feitas com relação ao broker no espaço de edição do item “Server”. Após devidamente configurado e após apertar o botão de “deploy” o status do bloco deve estar como mostrado abaixo:

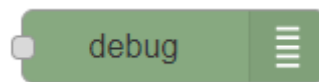
Bloco “MQTT in” configurado.



Fonte: Autor.

Para verificar o formato dos dados recebidos, utiliza-se o bloco de “debug” localizado na aba de “Commons” do node-red.

Bloco “debug”.



Fonte: Autor.

Conecta-se a saída do bloco “mqtt in” com o bloco de “debug”, ademais, configura-se o bloco de “debug” da seguinte forma:

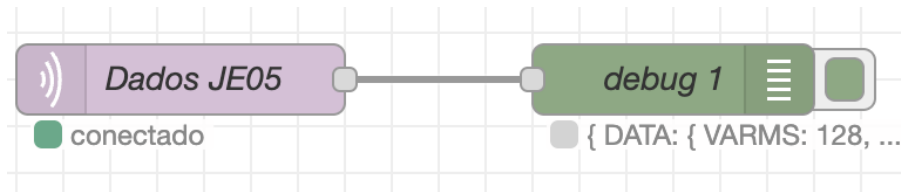
Configurações do bloco “debug”.

The image shows the 'Edit debug node' configuration window in Node-RED. At the top, there are three buttons: 'Deletar', 'Cancelar', and 'Feito'. Below these is a tab labeled 'Propriedades'. The configuration is divided into three sections: 'Saída' (Output) with a dropdown menu set to 'msg. payload'; 'Para' (To) with two checked options, 'janela de depuração' (debug window) and 'estado do nó (32 caracteres)' (node state (32 characters)), and an unchecked option 'console do sistema' (system console); and 'Nome' (Name) with a text input field containing 'debug 1'. There are also icons for settings, documentation, and a preview on the right side of the configuration area.

Fonte: Autor.

O conjunto de blocos deve ser esse:

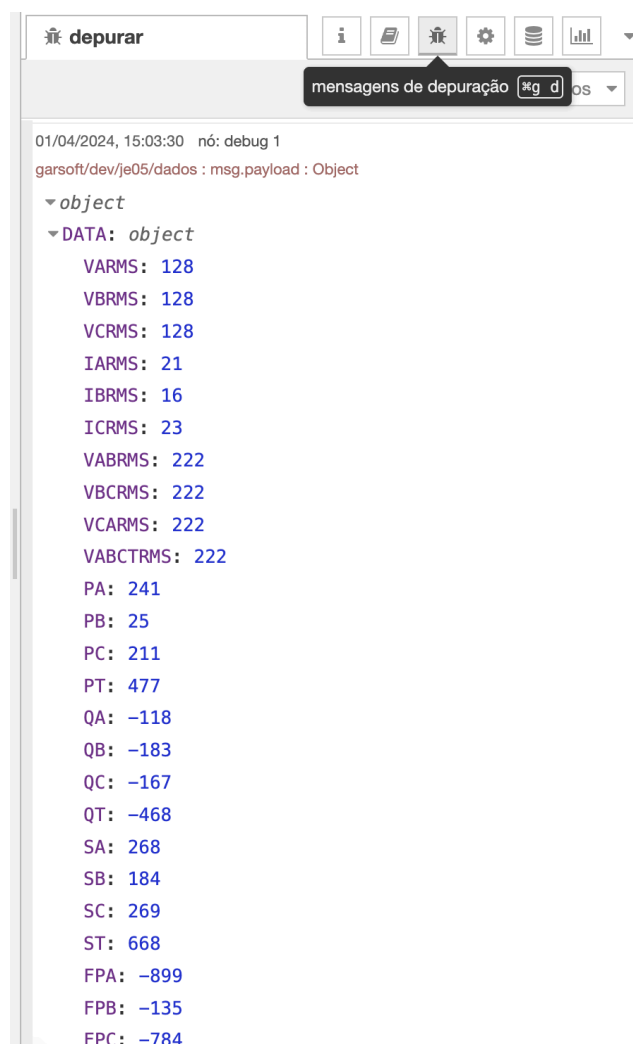
Conjunto de blocos “mqtt in” e “debug”.



Fonte: Autor.

Após o “deploy”, deve-se observar uma mensagem como essa no painel de “debug”:

JSON recebido do broker.

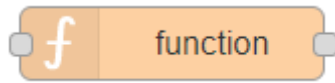


Fonte: Autor.

Pode-se acessar os valores da mensagem enviada pelo broker ao utilizar um bloco “function” localizado na seção de mesmo nome. Os blocos “function” são códigos em Javascript que podem ser executados em três situações diferentes: No começo do

fluxo; no fim do fluxo e ao receber uma mensagem. No caso, utilizar-se-á no modo “ao receber uma mensagem”.

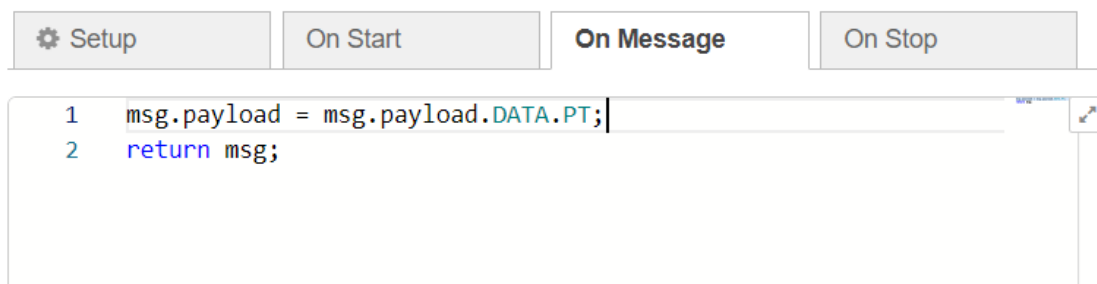
Bloco “function”.



Fonte: Autor.

Utilizar-se-á outro bloco de “debug” com a mesma configuração e o seguinte código dentro do bloco “function”:

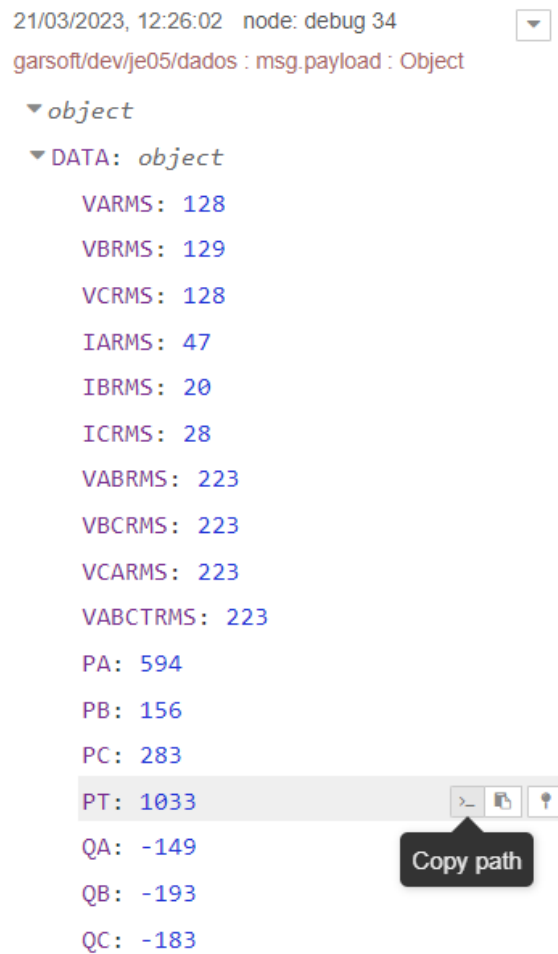
Código para filtrar o valor da potência total.



Fonte: Autor.

Caso haja dificuldade em identificar qual o atributo a ser acessado, é possível copiar sua “hierarquia” ao clicar no botão “copy path”:

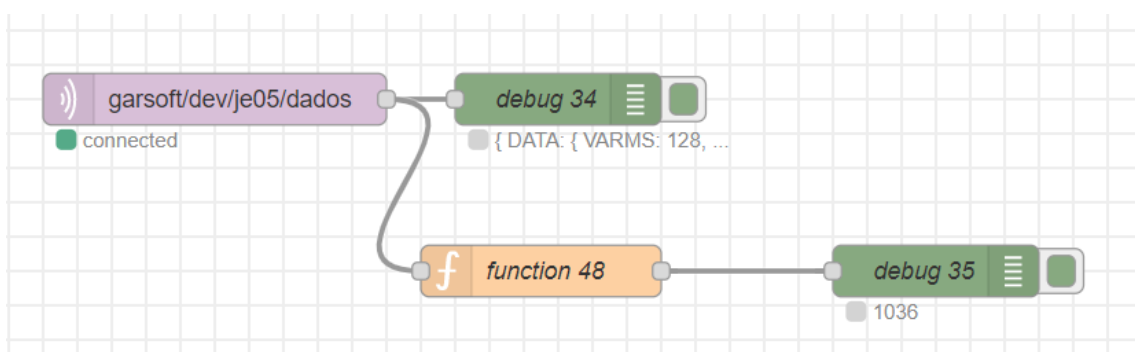
Botão “Copy path”.



Fonte: Autor.

No caso, o “copy path” será: “payload.DATA.PT”.

Conexão feita.

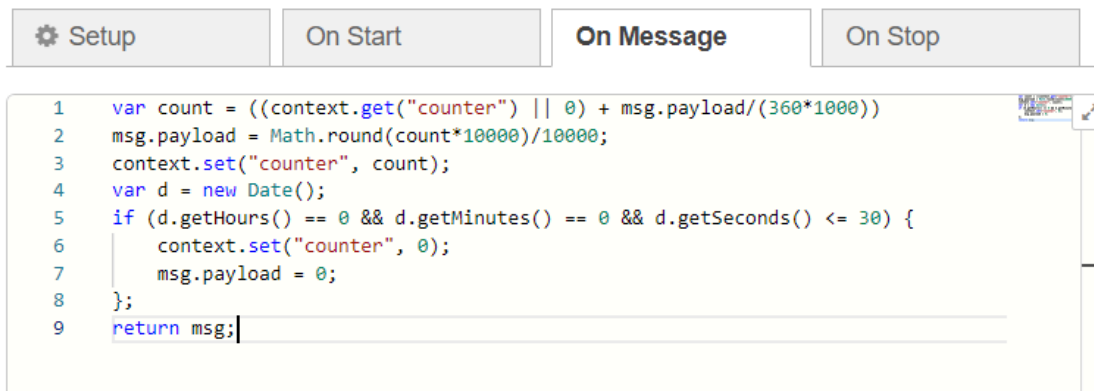


Fonte: Autor.

No caso, a mensagem enviada pelo bloco “function” contém em sua “carga” (payload) apenas um número inteiro. Esse número pode ser enviado para um bloco “chart” na seção de “dashboard” por exemplo.

É possível elaborar códigos mais complexos caso necessário, um exemplo o código que realizar o cálculo da energia consumida em kwh:

Código referente ao consumo diário.



```
1 var count = ((context.get("counter") || 0) + msg.payload/(360*1000))
2 msg.payload = Math.round(count*10000)/10000;
3 context.set("counter", count);
4 var d = new Date();
5 if (d.getHours() == 0 && d.getMinutes() == 0 && d.getSeconds() <= 30) {
6     context.set("counter", 0);
7     msg.payload = 0;
8 };
9 return msg;
```

Fonte: Autor.

Primeiramente, o código recebe do contexto “counter” um valor previamente guardado ou, caso nenhum valor tenha sido colocado antes, zero. Após isso, soma-se com o conteúdo da mensagem recebida dividido por 360*1000, pois, considera-se que a mesma potência tenha sido utilizada por um intervalo de dez segundos, sendo assim, para transformar watts em watts hora, tem-se 10/(60*60), depois, divide-se por 1000 para que seja transformado em kwh. O valor em kwh é armazenado na variável “count”.

Em sequência, o conteúdo da mensagem enviada é dado como um número inteiro de quatro casas decimais. Coloca-se o valor da variável “count” no contexto “counter”. Ademais, cria-se uma variável “d” que contém a data no momento de execução do código. Caso o horário obtido esteja entre 00:00:00 e 00:00:30 inclusive, o valor armazenado no contexto “counter” é zero e o conteúdo da mensagem enviada é zero.

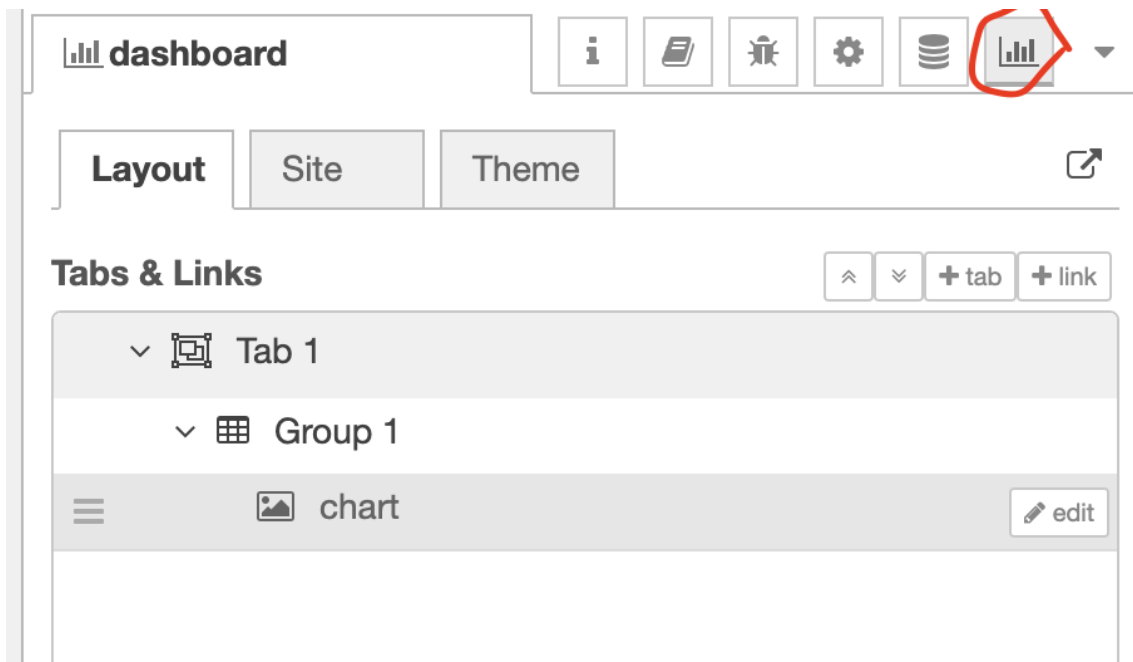
Configuração da Dashboard

A configuração do Dashboard se dá em um menu específico no node-red ao clicar-se na seta do canto superior direito do menu de abas (embaixo do botão de deploy).

Ao clicar em “Dashboard”, é possível realizar configurações referentes aos “grupos” nos quais os elementos visuais como gráficos, textos e seletores são colocados.

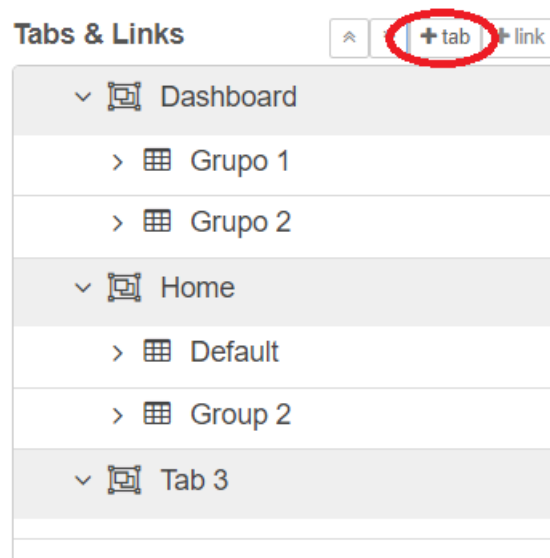
Para criar um dashboard, vá na aba de “dashboard” e crie uma nova “tab”

Menu de informações do node-red.



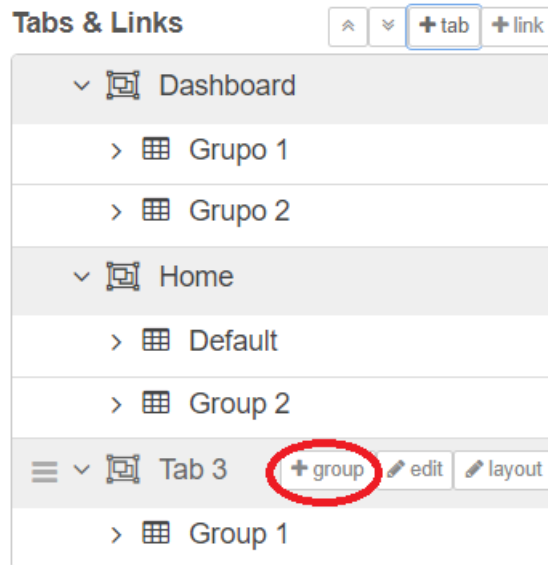
Fonte: Autor.

Botão para criar "tab".



Fonte: Autor.

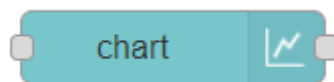
Botão para criar grupo.



Fonte: Autor.

Após criar o grupo, pode-se adicionar elementos a ele. Agora, cria-se um “chart” utilizando o bloco de mesmo nome da biblioteca “dashboard” supracitada.

Bloco “chart”.



Fonte: Autor.

Configurações utilizadas no bloco “chart”.

Editar chart nó

Deletar

Cancelar

Feito

Propriedades

Group

[Tab 1] Group 1

Size

auto

Label

chart

Type

Line chart

☐ enlarge points

X-axis

last

1

hours

OR

1000

points

X-axis Label

HH:mm:ss

☐ as UTC

Y-axis

min

max

Legend

None

Interpolate

linear

Series Colours

Blank label

display this text before valid data arrives

</> Class

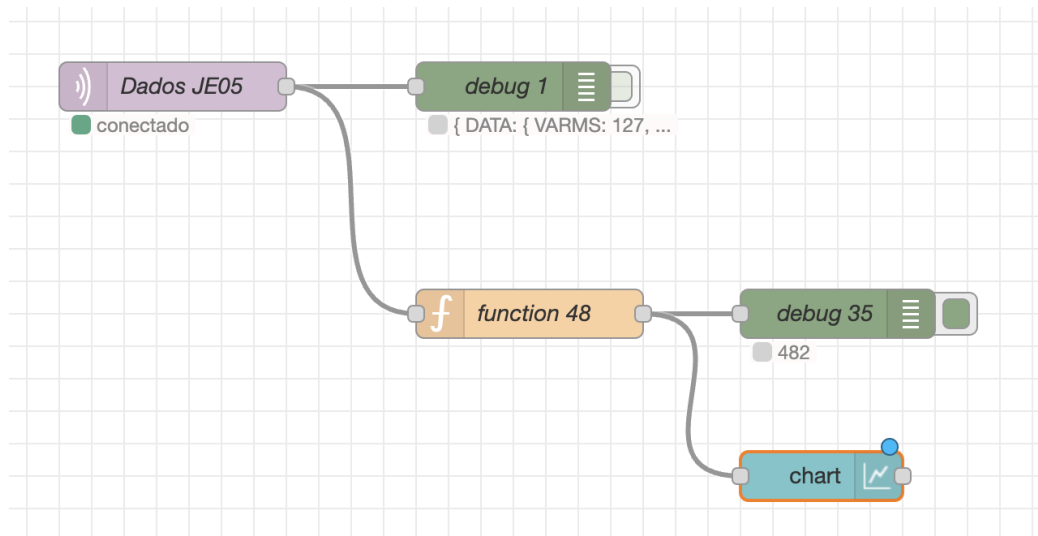
Optional CSS class name(s) for widget

Nome

Nome

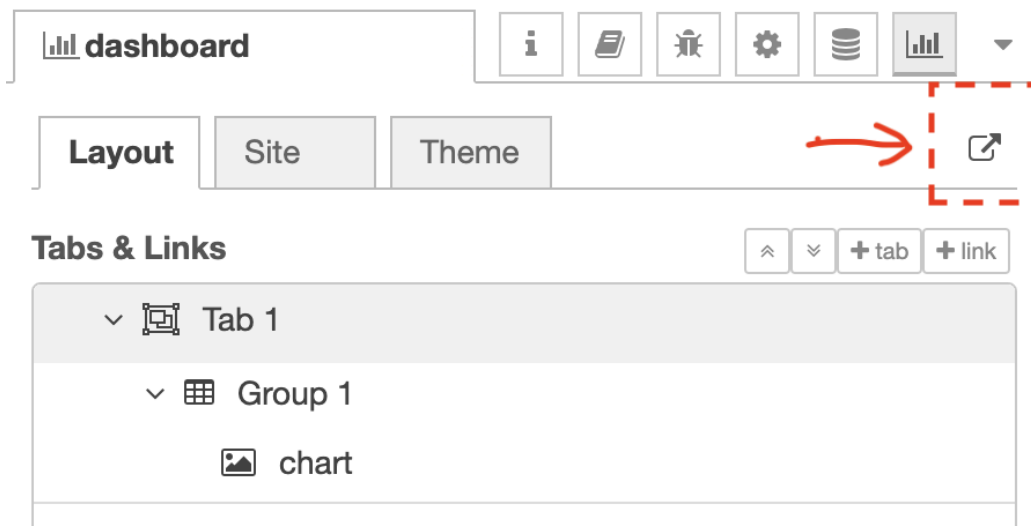
Fonte: Autor.

Para testarmos o chart, consumimos a saída da função 48, de acordo com a figura abaixo:

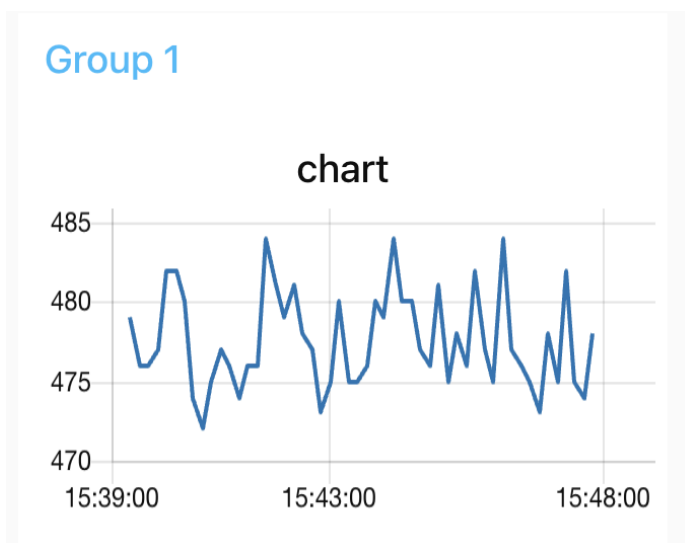


Fonte: Autor.

Depois, ao clicar no ícone de visualização do gráfico:



Podemos ver o Dashboard com “chart” funcionando.



Essa documentação ainda está em construção pelo “feedback” dos alunos da disciplina, caso tenha alguma dúvida ou sugestão envie um email para michelet@usp.br. Há um link com um docs google do documento mais atualizado:

<https://docs.google.com/document/d/1OcDE6uVBBJHRpcAnxR6XHvMaLq7MqbwYUYtP4LcoQE/edit>

Acesso via SQL - Histórico de dados

O histórico das medições pode ser acesso via consultas sql no banco de dados postgres. Os dados de acesso são:

host: 143.107.102.8

portal:5432

database: data-energia

usuário: leitor

senha: garsoftData2024

O esquema de dados é o seguinte:

<table><tr><th colspan="2">demanda</th></tr><tr><td>potencia_aparente_i</td><td>real</td></tr><tr><td>potencia_aparente_t</td><td>real</td></tr><tr><td>potencia_aparente_c</td><td>real</td></tr><tr><td>potencia_aparente_tot</td><td>real</td></tr><tr><td>timestamp</td><td>timestamp</td></tr><tr><td>demandaid</td><td>integer</td></tr></table>	demanda		potencia_aparente_i	real	potencia_aparente_t	real	potencia_aparente_c	real	potencia_aparente_tot	real	timestamp	timestamp	demandaid	integer	<table><tr><th colspan="2">detalhes_a</th></tr><tr><td>potencia_reativa_i</td><td>real</td></tr><tr><td>potencia_ativa_i</td><td>real</td></tr><tr><td>fator_potencia_i</td><td>real</td></tr><tr><td>timestamp</td><td>timestamp</td></tr><tr><td>detalhes_a_ic</td><td>integer</td></tr></table>	detalhes_a		potencia_reativa_i	real	potencia_ativa_i	real	fator_potencia_i	real	timestamp	timestamp	detalhes_a_ic	integer
demanda																											
potencia_aparente_i	real																										
potencia_aparente_t	real																										
potencia_aparente_c	real																										
potencia_aparente_tot	real																										
timestamp	timestamp																										
demandaid	integer																										
detalhes_a																											
potencia_reativa_i	real																										
potencia_ativa_i	real																										
fator_potencia_i	real																										
timestamp	timestamp																										
detalhes_a_ic	integer																										
<table><tr><th colspan="2">consumo</th></tr><tr><td>consumo_ativo_acumulado_</td><td>real</td></tr><tr><td>consumo_ativo_acumulado_</td><td>real</td></tr><tr><td>consumo_ativo_acumulado_</td><td>real</td></tr><tr><td>consumo_ativo_acumulado_tot</td><td>real</td></tr><tr><td>timestamp</td><td>timestamp</td></tr><tr><td>consumoid</td><td>integer</td></tr></table>	consumo		consumo_ativo_acumulado_	real	consumo_ativo_acumulado_	real	consumo_ativo_acumulado_	real	consumo_ativo_acumulado_tot	real	timestamp	timestamp	consumoid	integer	<table><tr><th colspan="2">detalhes_b</th></tr><tr><td>potencia_reativa_l</td><td>real</td></tr><tr><td>potencia_ativa_l</td><td>real</td></tr><tr><td>fator_potencia_t</td><td>real</td></tr><tr><td>timestamp</td><td>timestamp</td></tr><tr><td>detalhes_b_id</td><td>integer</td></tr></table>	detalhes_b		potencia_reativa_l	real	potencia_ativa_l	real	fator_potencia_t	real	timestamp	timestamp	detalhes_b_id	integer
consumo																											
consumo_ativo_acumulado_	real																										
consumo_ativo_acumulado_	real																										
consumo_ativo_acumulado_	real																										
consumo_ativo_acumulado_tot	real																										
timestamp	timestamp																										
consumoid	integer																										
detalhes_b																											
potencia_reativa_l	real																										
potencia_ativa_l	real																										
fator_potencia_t	real																										
timestamp	timestamp																										
detalhes_b_id	integer																										
	<table><tr><th colspan="2">detalhes_c</th></tr><tr><td>potencia_reativa_c</td><td>real</td></tr><tr><td>potencia_ativa_c</td><td>real</td></tr><tr><td>fator_potencia_c</td><td>real</td></tr><tr><td>timestamp</td><td>timestamp</td></tr><tr><td>detalhes_c_ic</td><td>integer</td></tr></table>	detalhes_c		potencia_reativa_c	real	potencia_ativa_c	real	fator_potencia_c	real	timestamp	timestamp	detalhes_c_ic	integer														
detalhes_c																											
potencia_reativa_c	real																										
potencia_ativa_c	real																										
fator_potencia_c	real																										
timestamp	timestamp																										
detalhes_c_ic	integer																										