

# AI & CHATBOT

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

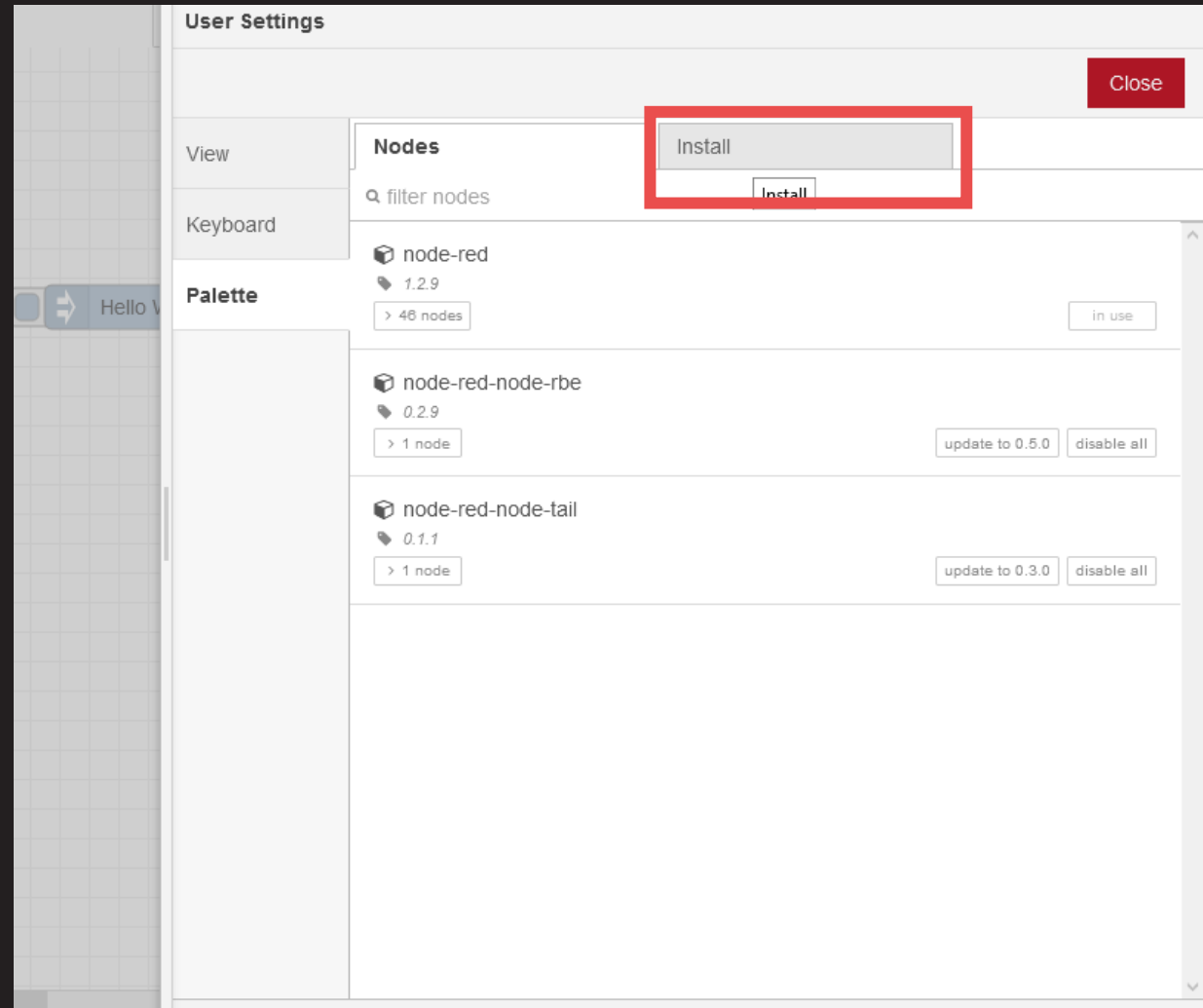
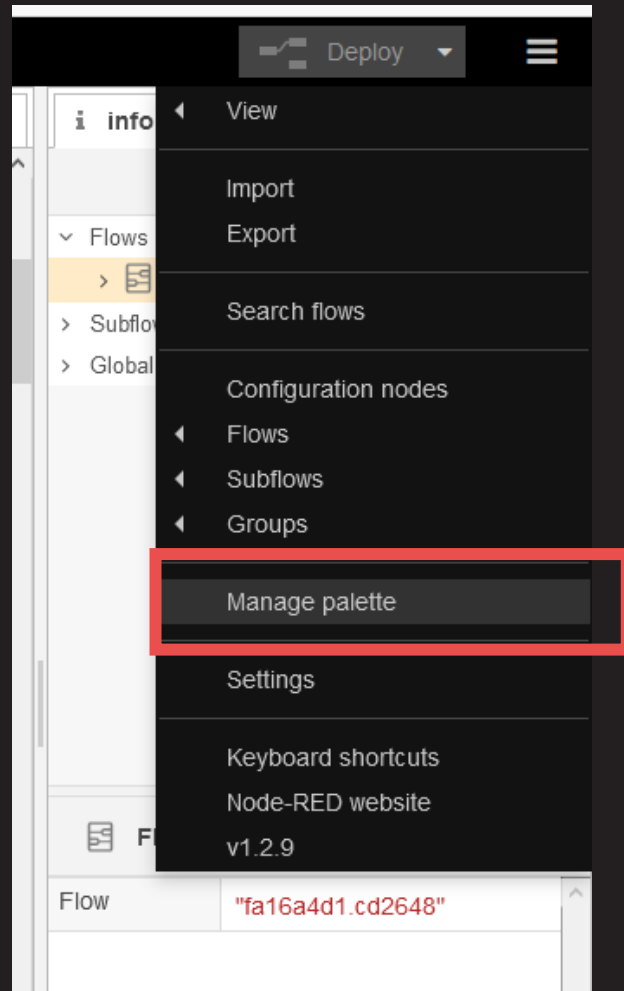
Prof. Daniel Sória

FIAP  
GRADUAÇÃO

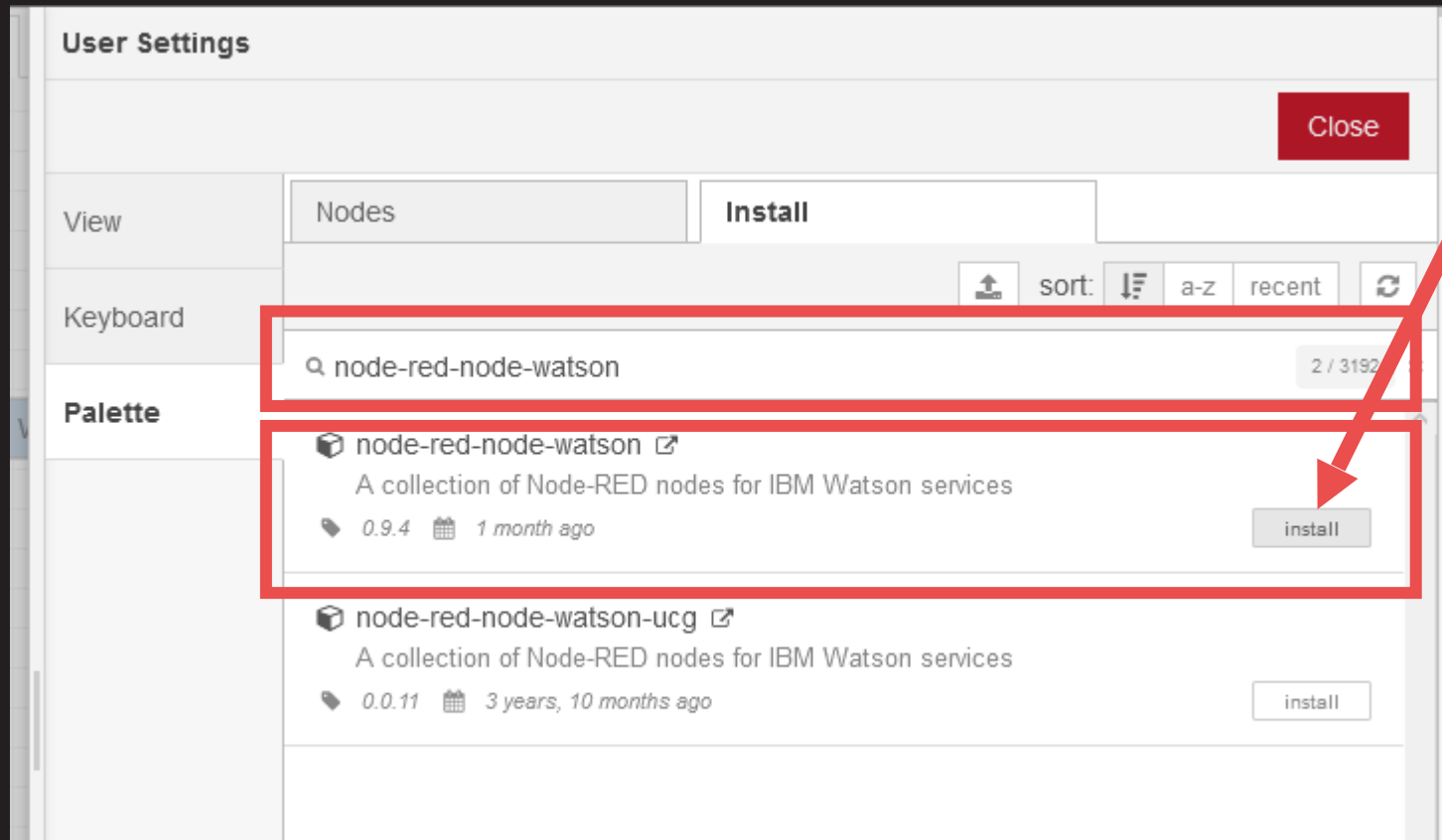
# Nós especiais e bibliotecas

Instalando a biblioteca de desenvolvimento da IBM Cloud e do Telegram

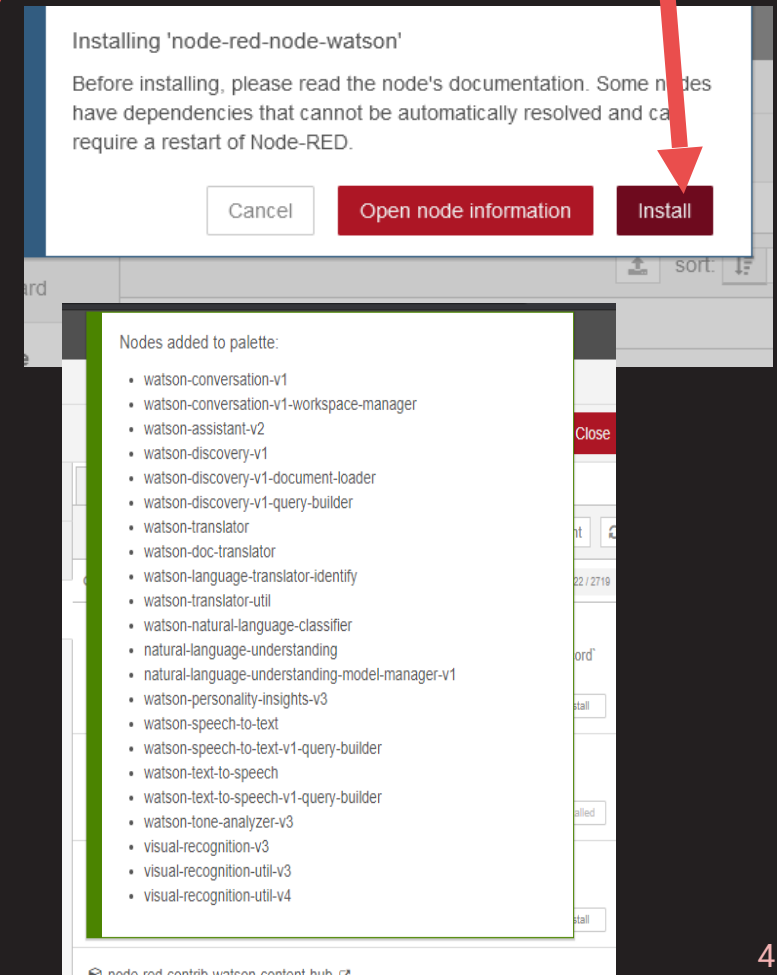
# Instalando a bibliotecas



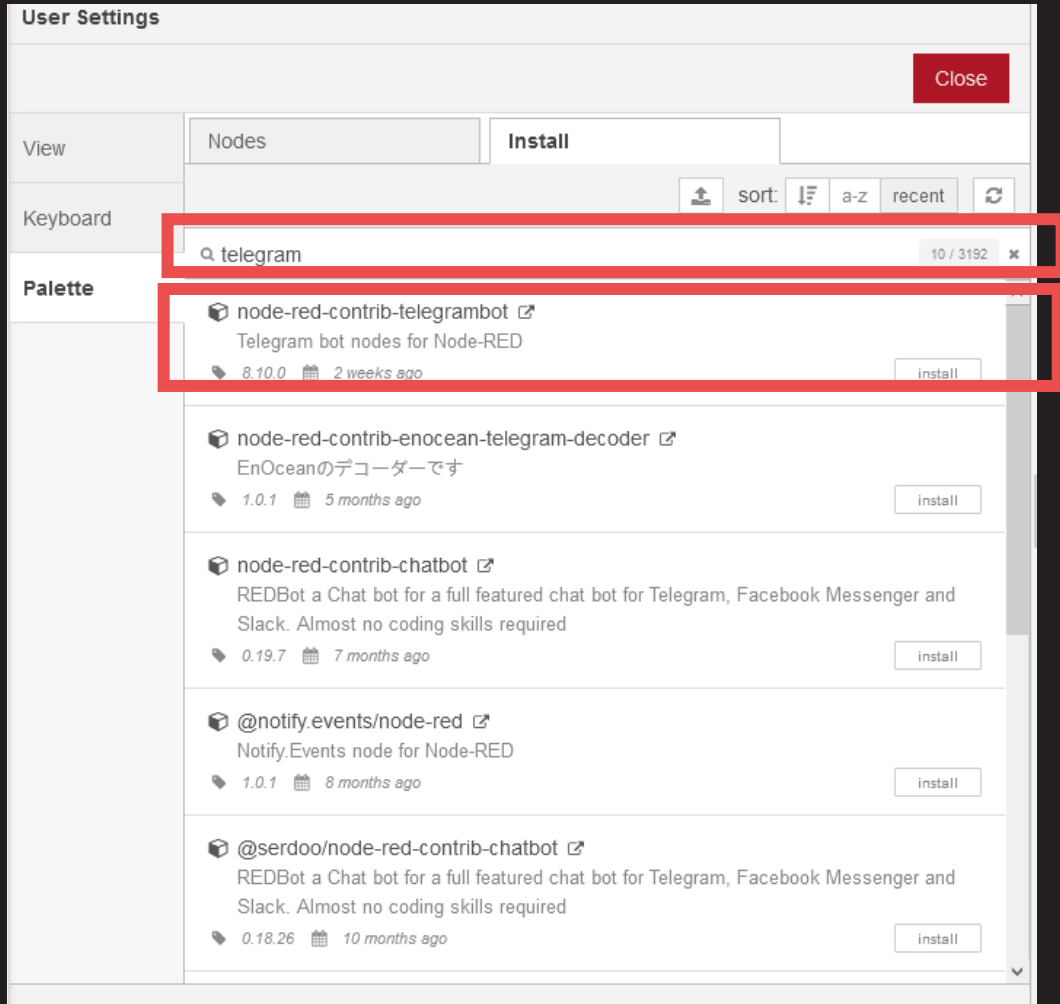
# Instalando a biblioteca do IBM Watson



## Instale o pacote



# Instalando a biblioteca do Telegram



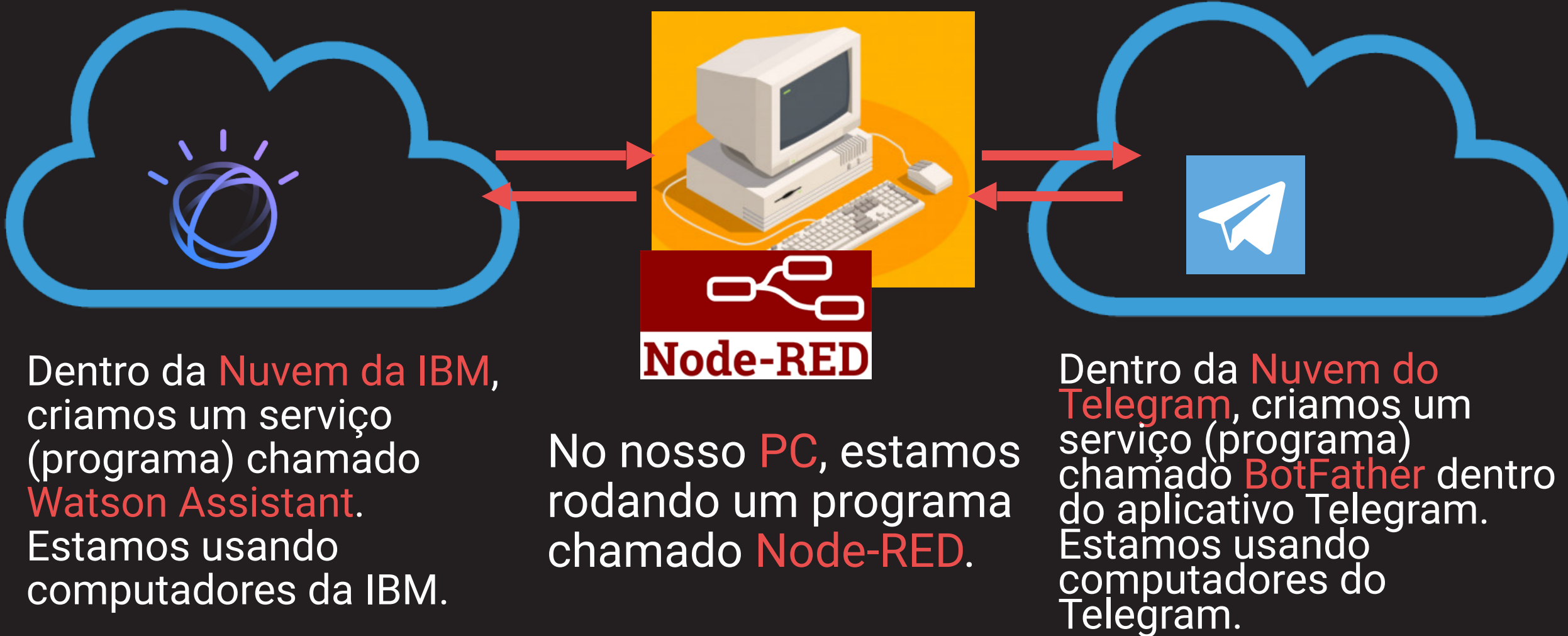
## Nodes added to palette:

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

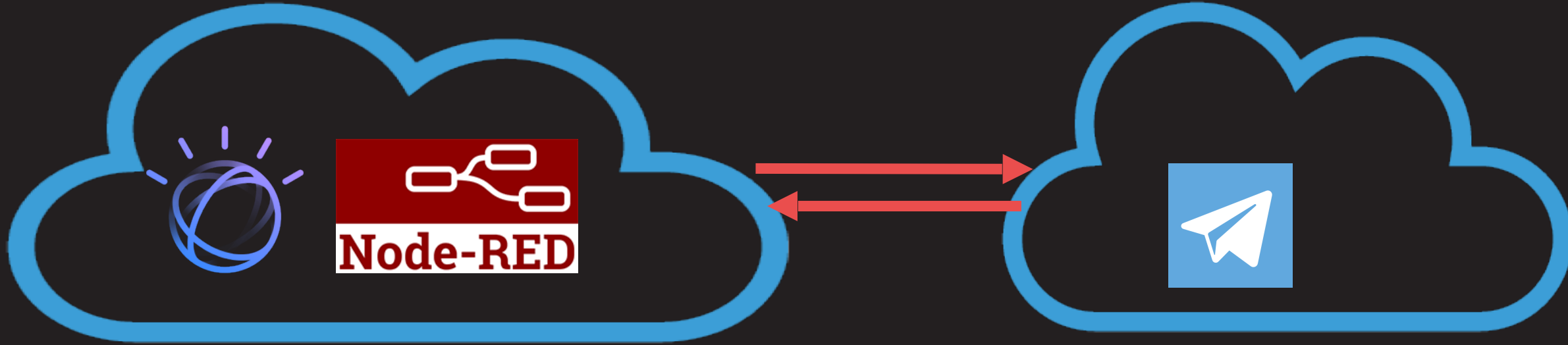
# Integração de Serviços

Entendendo como as coisas se conectam na internet

# Integração de Serviços – Computação Distribuída



# Integração de Serviços – Computação Distribuída



Dentro da **Nuvem da IBM**, podemos criar mais de um serviço: o **Watson Assistant** e o **Node-Red**. Estamos usando computadores da IBM.

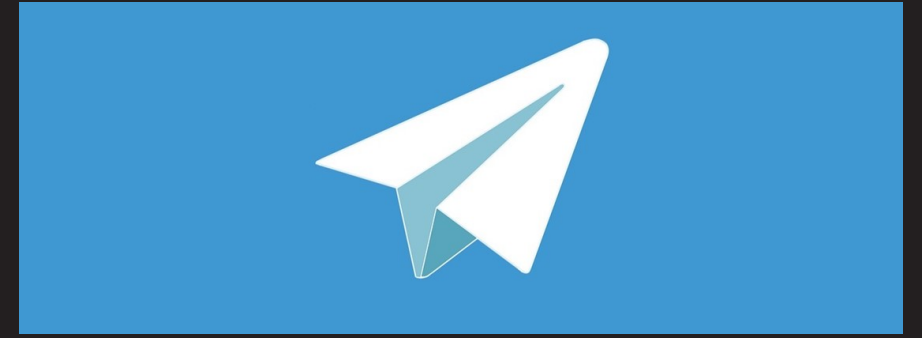
Dentro da **Nuvem do Telegram**, criamos um serviço (programa) chamado **BotFather** dentro do aplicativo Telegram. Estamos usando computadores do Telegram.



# Telegram

Conectando o bot a um serviço de mensagens

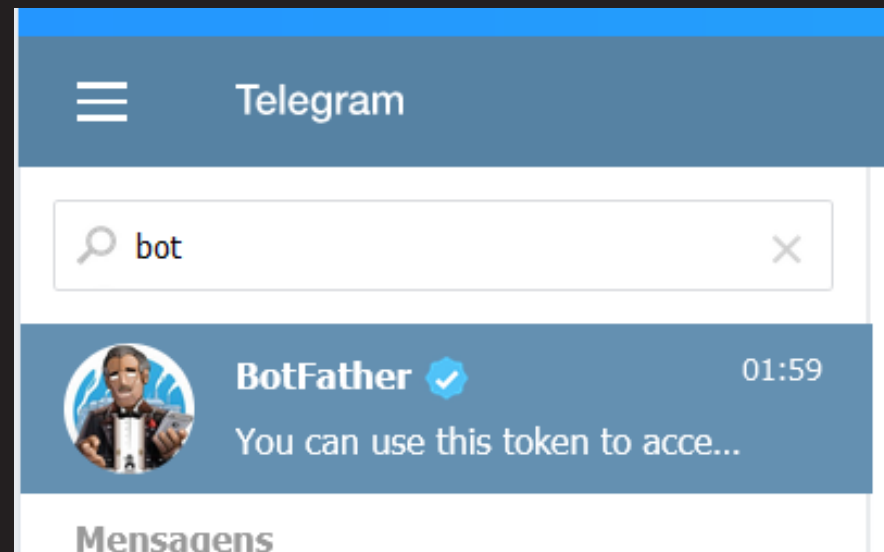
# Telegram



- Nesta primeira etapa vamos apenas habilitar o serviço de criação de bots do Telegram;
- Escolhemos o Telegram pois não é necessária uma conta empresa na plataforma para poder criar bots;
- Primeiro passo: instale o aplicativo do Telegram no seu celular;
- Segundo passo: acesso o Telegram Web no seu navegador;

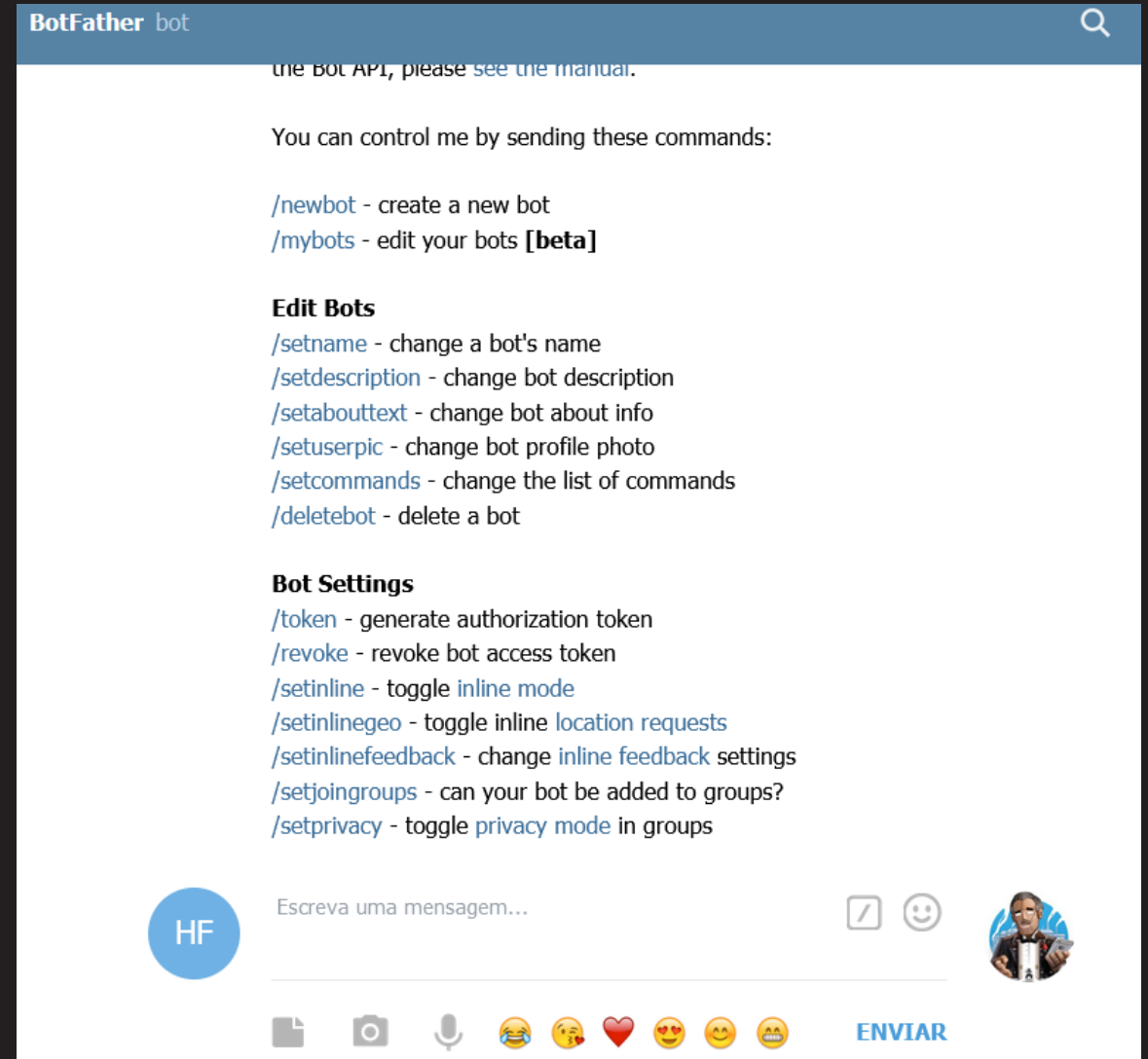
# Telegram - Botfather

- Na barra de busca, digite BotFather.
- Selecione o bonequinho maneiro do BotFather imitando o bom Don Corleone em O Poderoso Chefão.



# Telegram - Botfather

- O BotFather irá soltar uma mensagem com uma série de comandos para se criar e editar um bot



# Telegram - Botfather

- Digite **/newbot** para criar um bot;
- Dê um nome para seu bot
- Agora escolha um username para ele. Atenção, usernames são públicos, então você precisa criar um username único.



**Henrique**  
/newbot



**BotFather**

Alright, a new bot. How are we going to call it? Please choose a name for your bot.



**Henrique**  
Vendas



**BotFather**

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris\_bot.



**Henrique**  
vendas\_bot

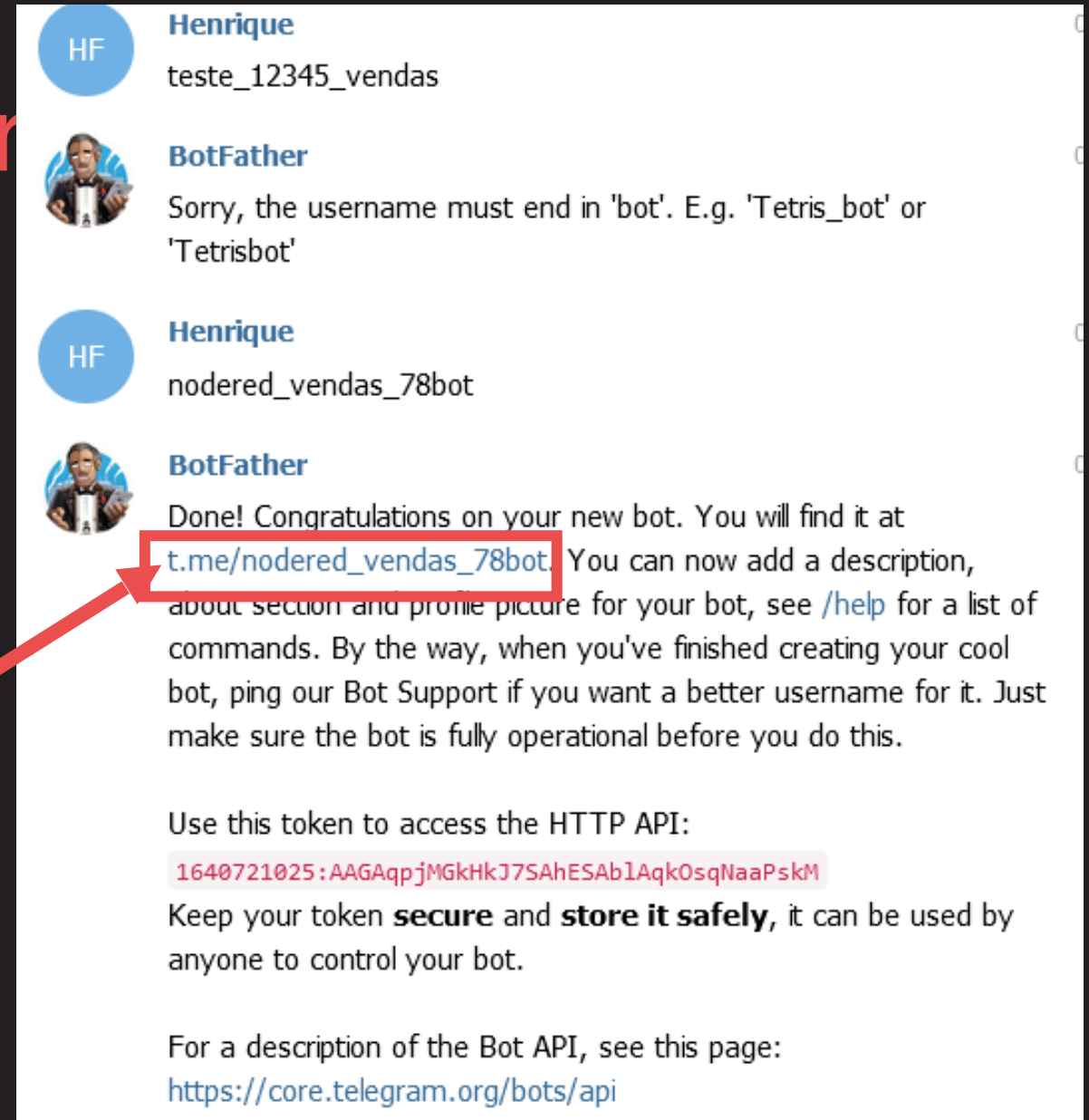


**BotFather**

Sorry, this username is already taken. Please try something different.

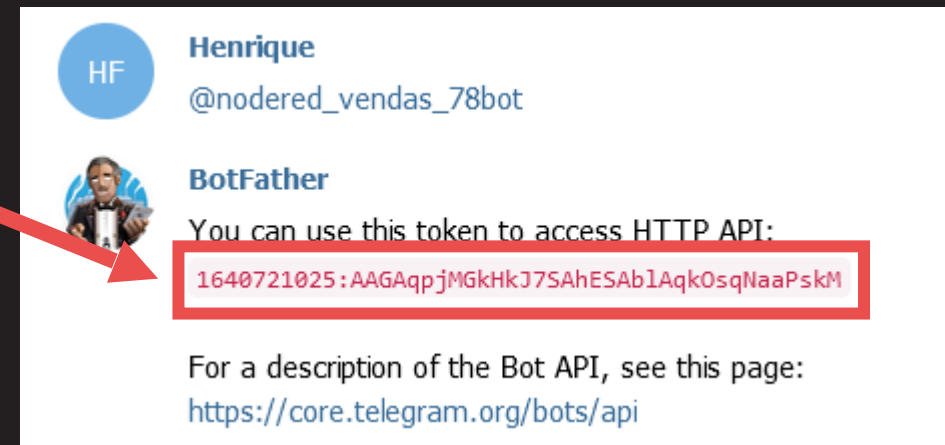
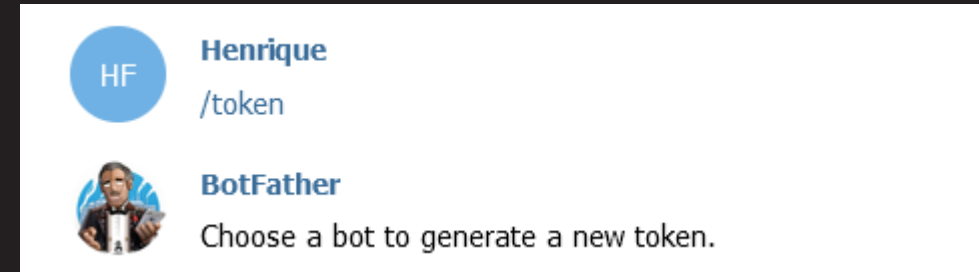
# Telegram - Botfather

- Não esqueça de escrever bot na parte final do username.
- O BotFather irá soltar uma mensagem de criação bem sucedida.
- Este é o link para iniciar uma conversa com o seu bot;
- Anote o número do HTTP API para acessar o seu bot.



# Telegram - Botfather

- Digite **/token** para verificar o token do seu bot no telegrama;
- Em seguida coloque **@username** do seu bot para verificar o HTTP API sempre que necessário;
- Esse número será usado no Node-RED para criar uma conexão entre o Telegram e o Watson Assistant;



# Testando o bot com Node-RED

Fluxo de demonstração para testar a conexão com o bot e o Telegram



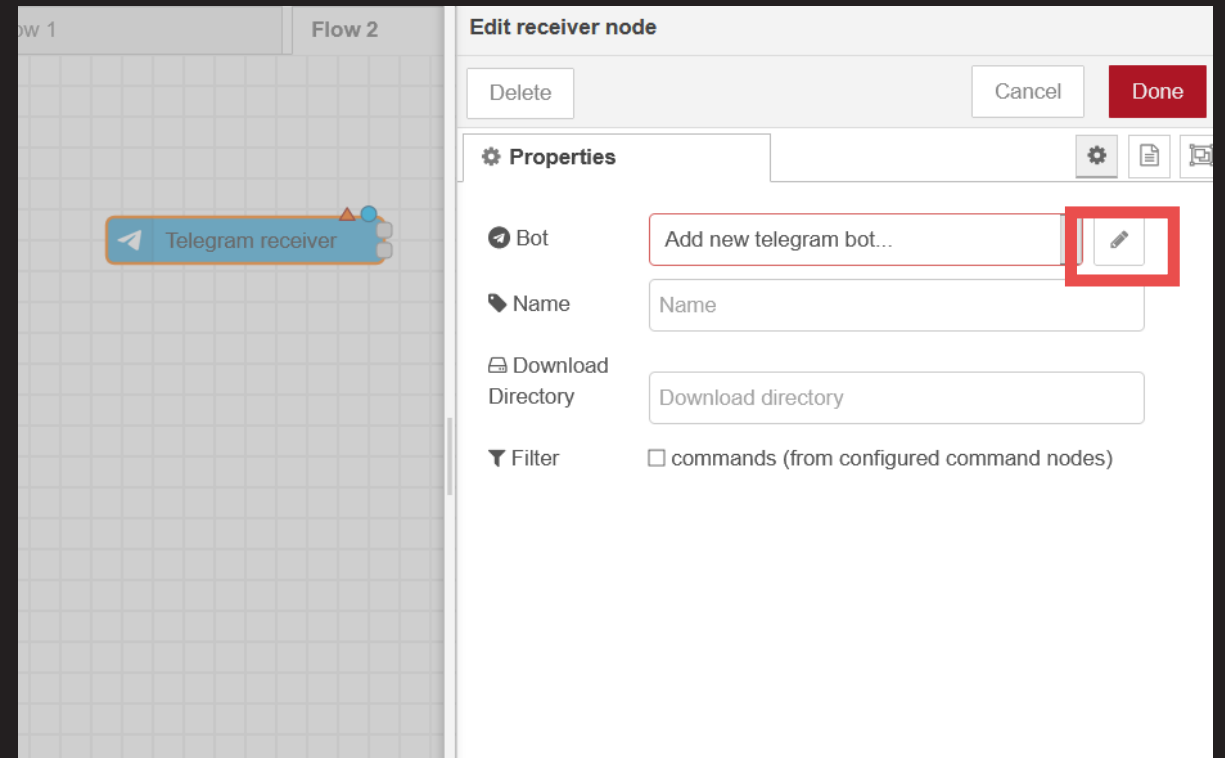
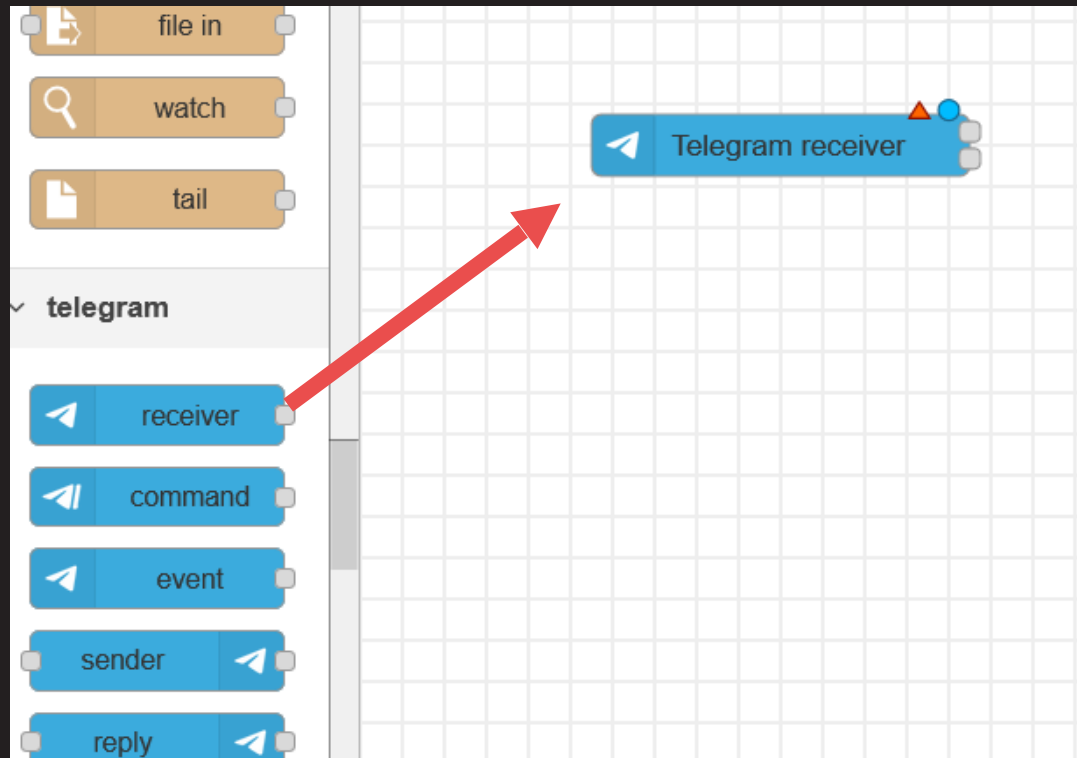
# Abrindo o Node-RED

- Abra o terminal e suba o servidor Node-RED na sua máquina;
- Para o ambiente de desenvolvimento,, coloque o IP local e porta no seu navegador;

```
D:\>node-red
4 Apr 12:32:37 - [info]
Welcome to Node-RED
=====
4 Apr 12:32:37 - [info] Node-RED version: v1.2.9
4 Apr 12:32:37 - [info] Node.js version: v12.13.0
4 Apr 12:32:37 - [info] Windows_NT 10.0.19042 x64 LE
4 Apr 12:32:39 - [info] Loading palette nodes
4 Apr 12:32:43 - [info] Settings file : C:\Users\Ferreira\.node-red\settings.js
4 Apr 12:32:43 - [info] Context store : 'default' [module=memory]
4 Apr 12:32:43 - [info] User directory : C:\Users\Ferreira\.node-red
4 Apr 12:32:43 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Apr 12:32:43 - [info] Flows file : C:\Users\Ferreira\.node-red\flows_DESKTOP-E3VFKJ5.json
4 Apr 12:32:43 - [info] Server now running at http://127.0.0.1:1880/
4 Apr 12:32:43 - [warn]
```

# Telegram + Node-RED

Vamos adicionar um nó de recebimento do telegram (**telegram receiver**); Duplo clique no nó para alterar as propriedades, e então, clique no ícone do lápis;



# Telegram + Node-RED

Precisamos colocar o nome do Bot e o **Token** para se conectar com o bot que criamos através do BotFather. Uma vez preenchido, clique em **Update**.

Edit receiver node > Edit telegram bot node

Delete Cancel Update

**Properties**

**Bot-Name** (Name of bot to connect to)

**Token** (Enter the bot token from botfather here)

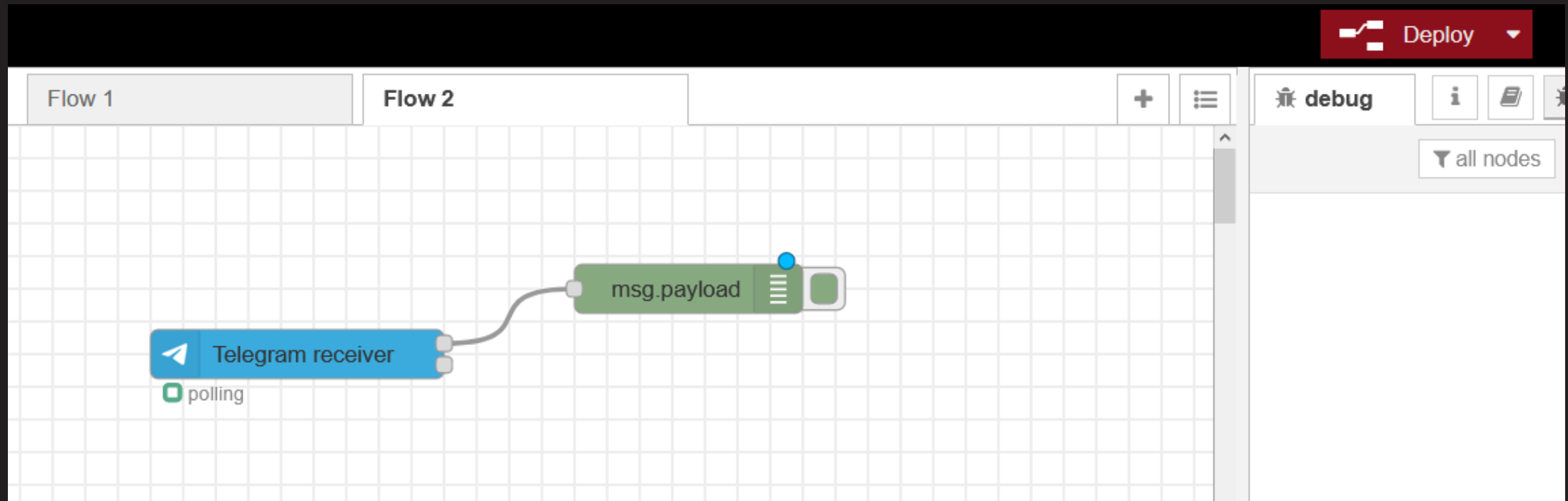
**Tip:** If you don't have a token yet, you can create a new one here: [@BotFather](#).

**Users** (Optional list of authorized user names e.g.: hugo,sepp,egon)

**ChatIds** (Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)

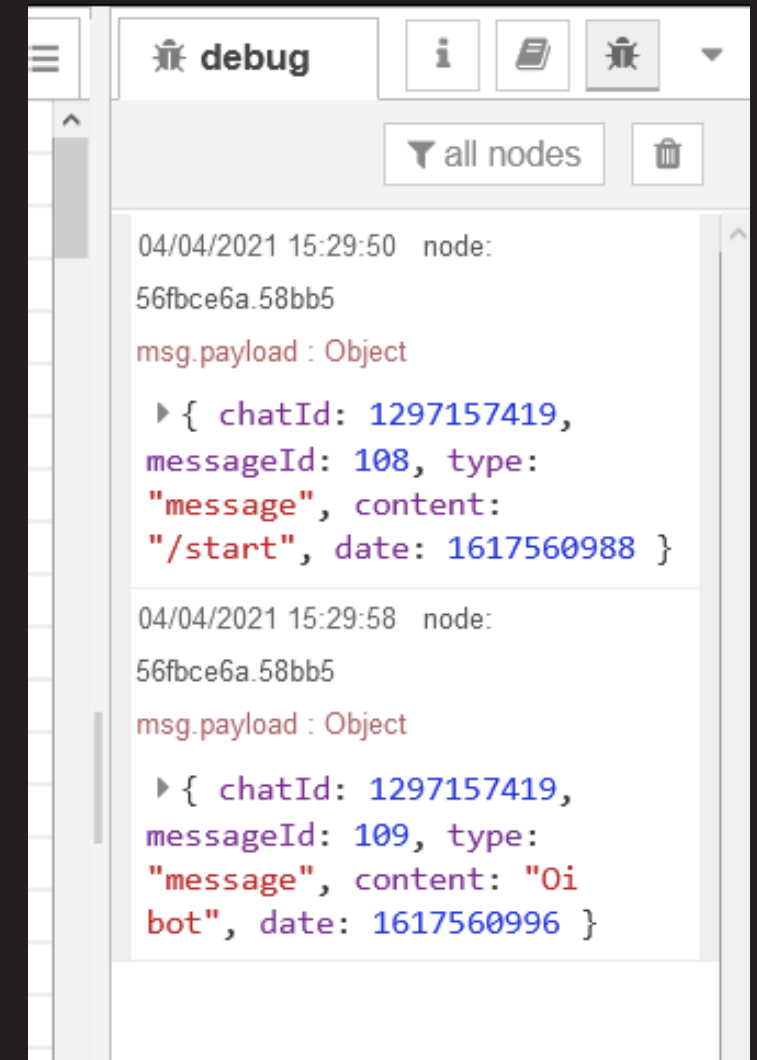
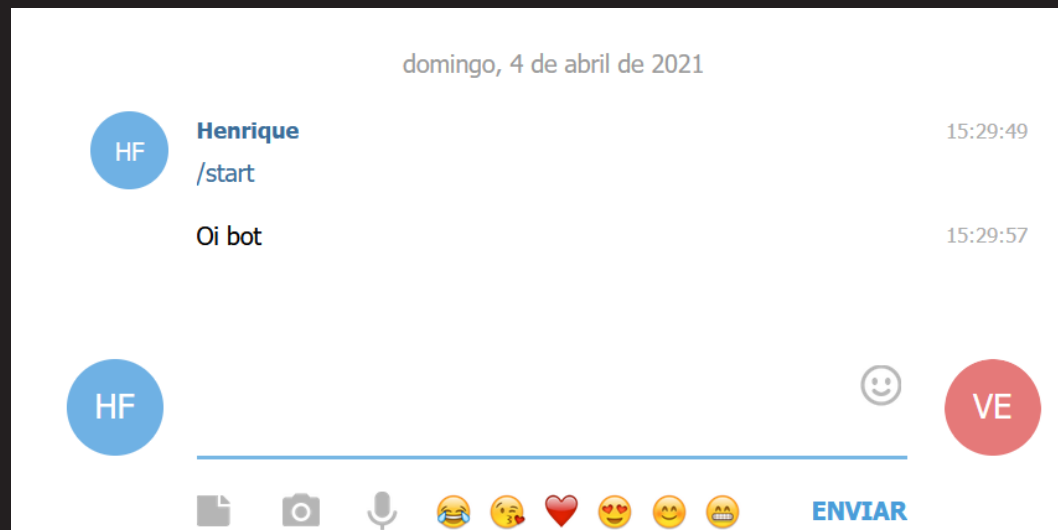
# Telegram + Node-RED

Vamos testar se estamos recebendo as mensagens do Telegram no Node-RED. Adicionamos um nó de Debug e damos deploy



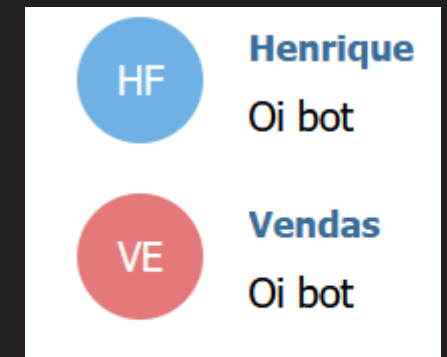
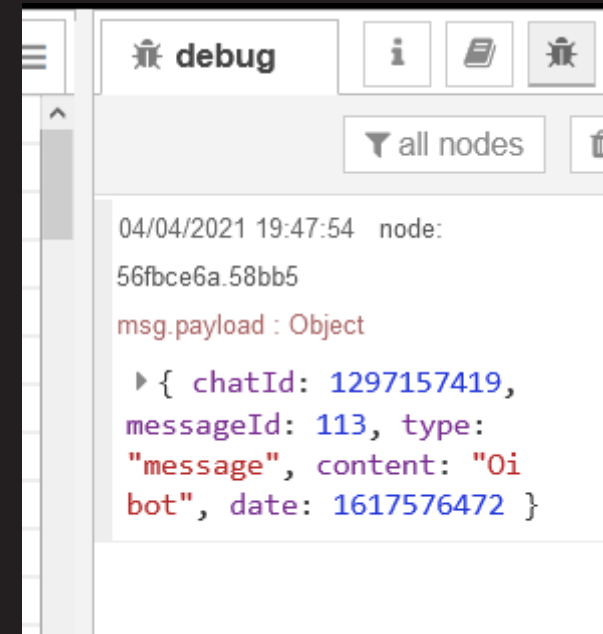
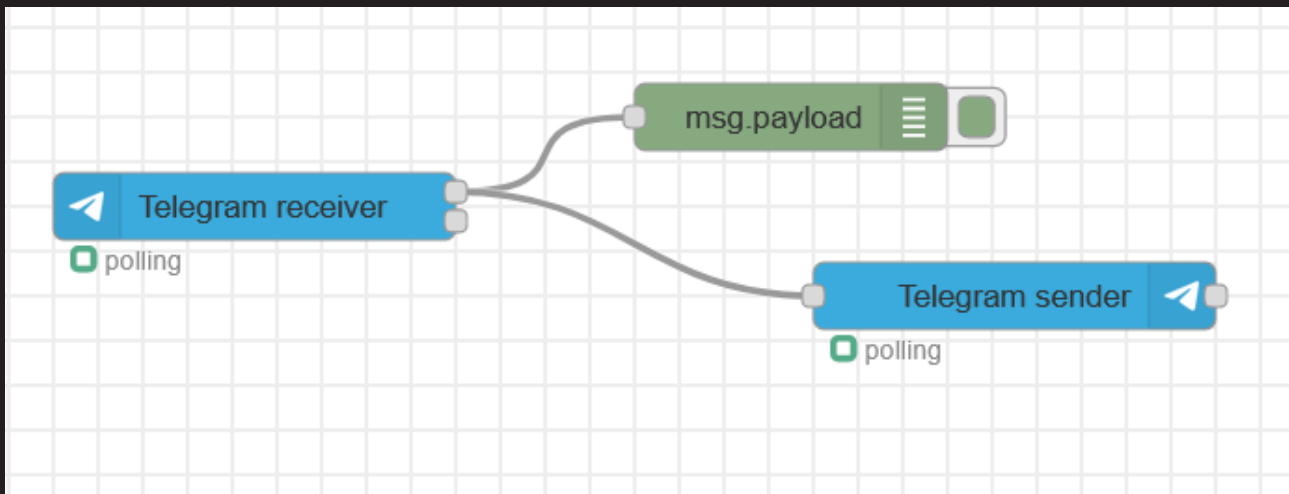
# Telegram + Node-RED

No Telegram Web (ou no seu celular), digite alguma mensagem para o bot. Não esqueça de conversar com o seu bot e não com o BotFather, clicando no link fornecido. Observe o menu de debug no Node-RED.



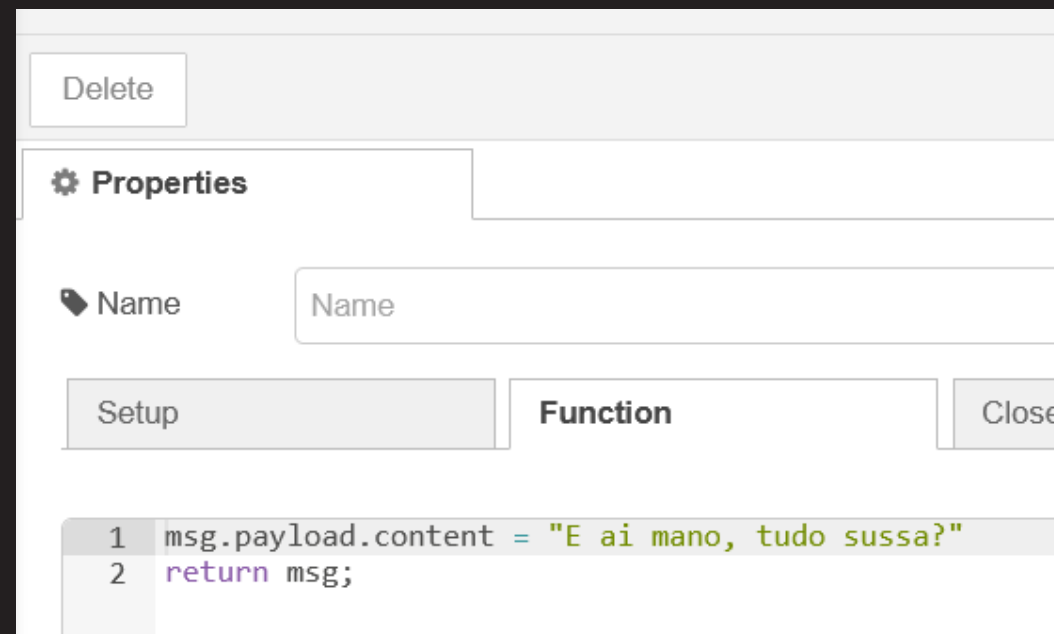
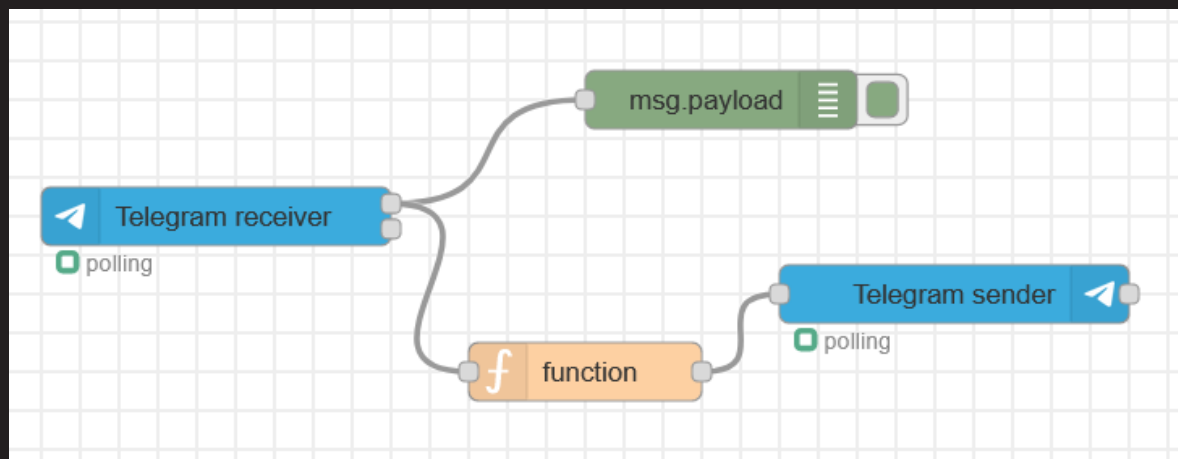
# Telegram + Node-RED

Vamos fazer o bot do Telegram repetir para nós o que dizemos para ele. Para isso, basta usar o **Telegram Sender** na frente do **Telegram Receiver** (não esqueça de configurar o **Token** no nó sender):



# Telegram + Node-RED

Podemos processar a mensagem recebida com um nó de function e depois enviá-la para o sender:



Resultado:

HF	<b>Henrique</b> oi bot
VE	<b>Vendas</b> E ai mano, tudo sussa?

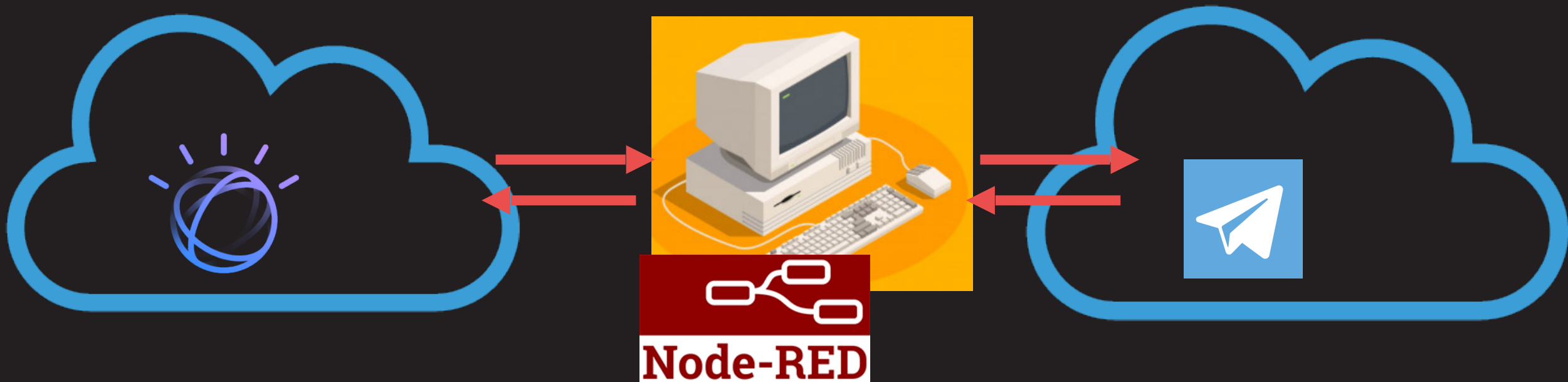
**Pergunta:** o que o bot responde para outra pergunta? Como podemos deixá-lo mais inteligente?

# Integrando nosso bot

Conectando o IBM Watson Assistant ao Telegram



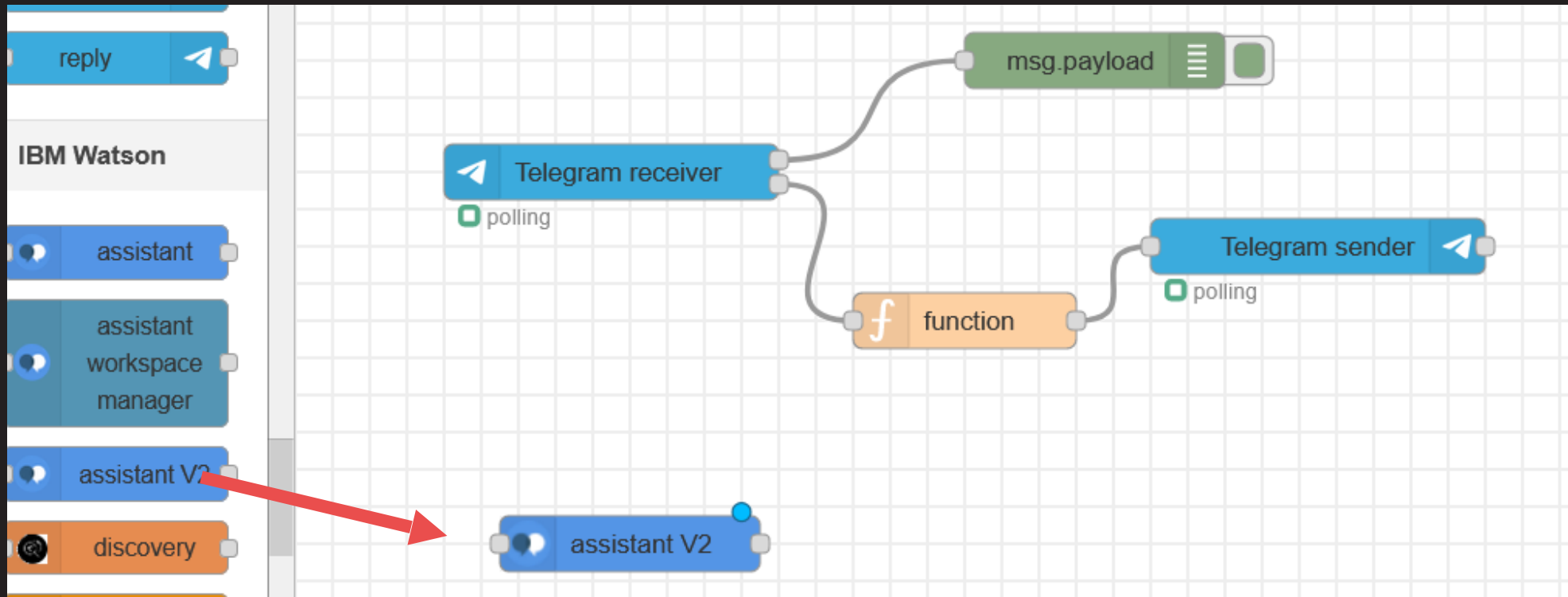
# Vamos usar o Watson Assistant para resolver o processamento inteligente da mensagem



- Vamos começar programando nossa integração localmente em nossas máquinas;
- O servidor Node-RED no nosso computador servirá de orquestrador;

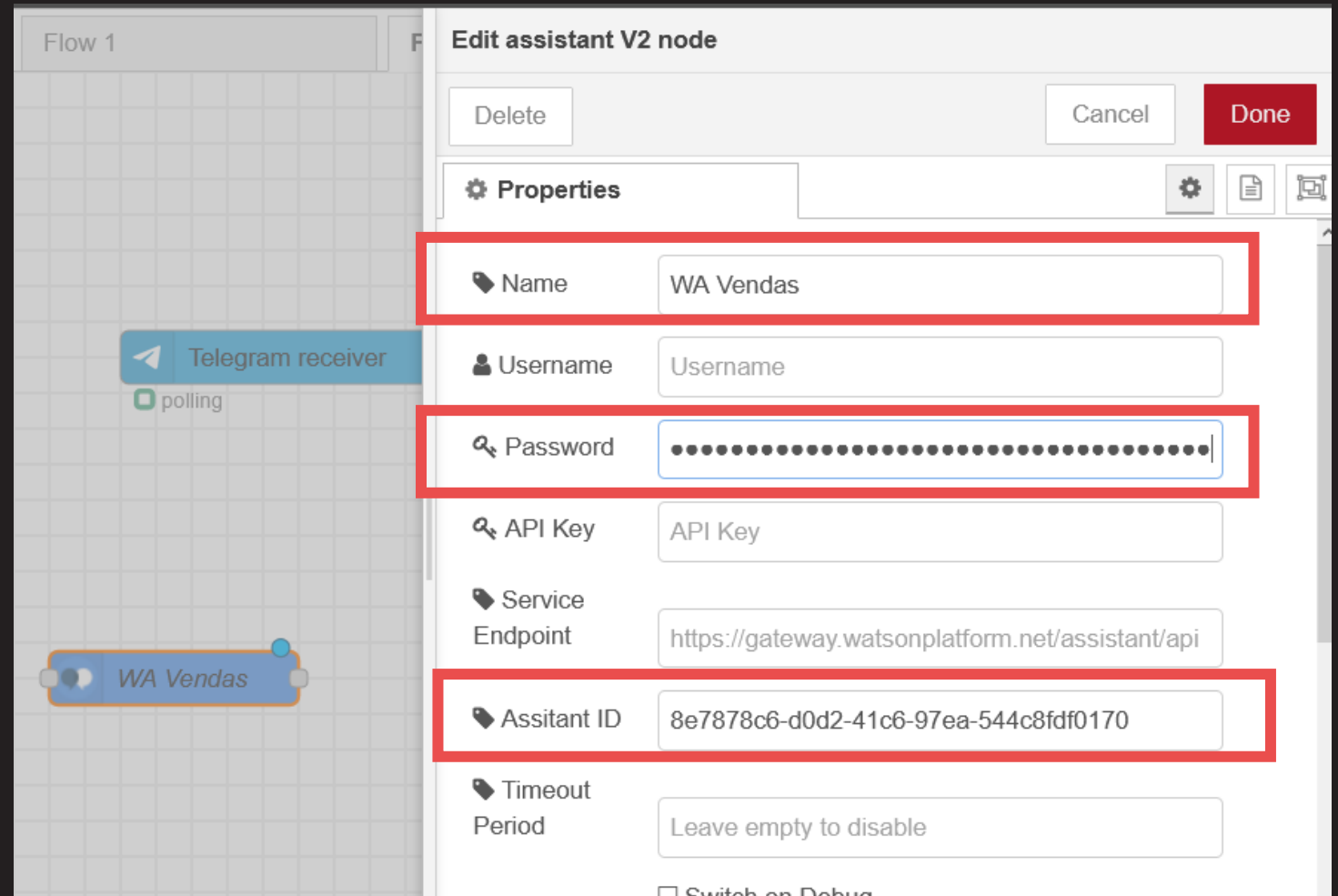
# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Vamos trocar o nosso nó de **function** pelo nó do **assistant V2**:



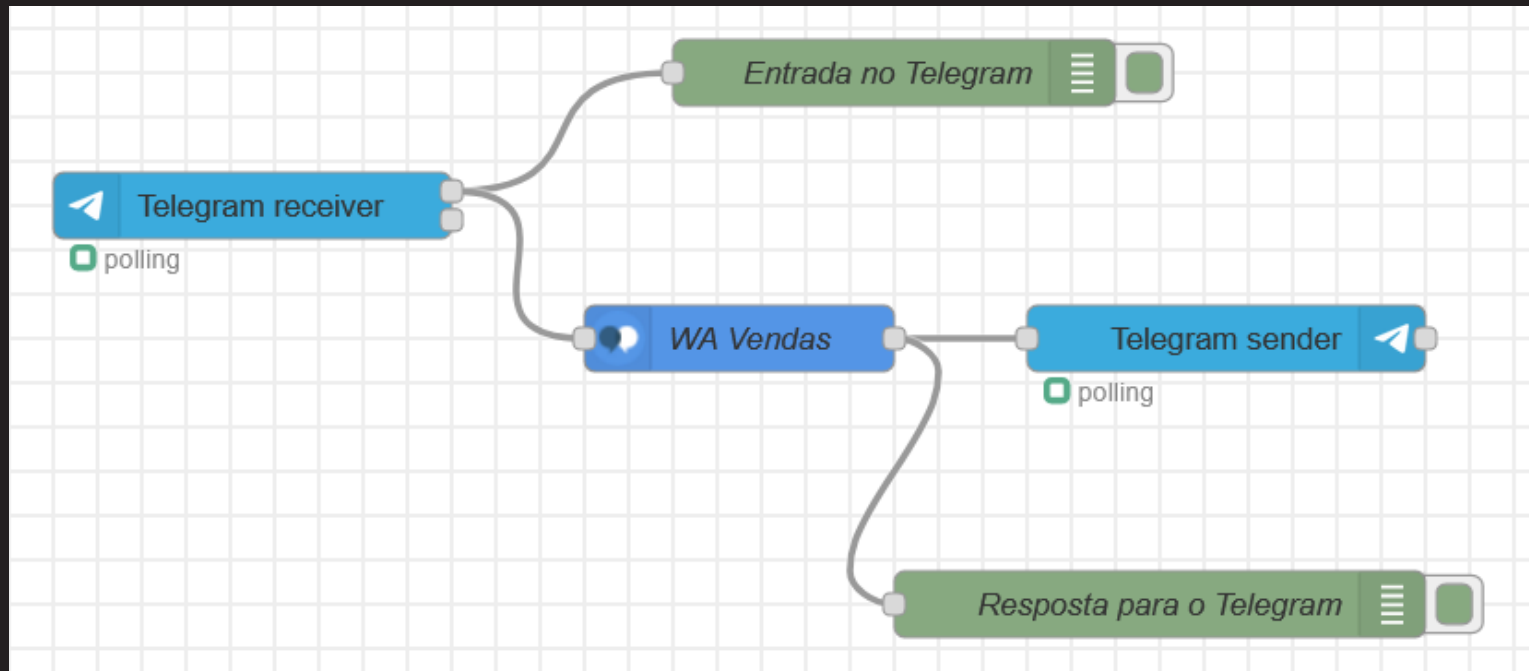
# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Como na aula anterior, precisamos colocar as credências do nosso serviço na nuvem da IBM dentro do nó;



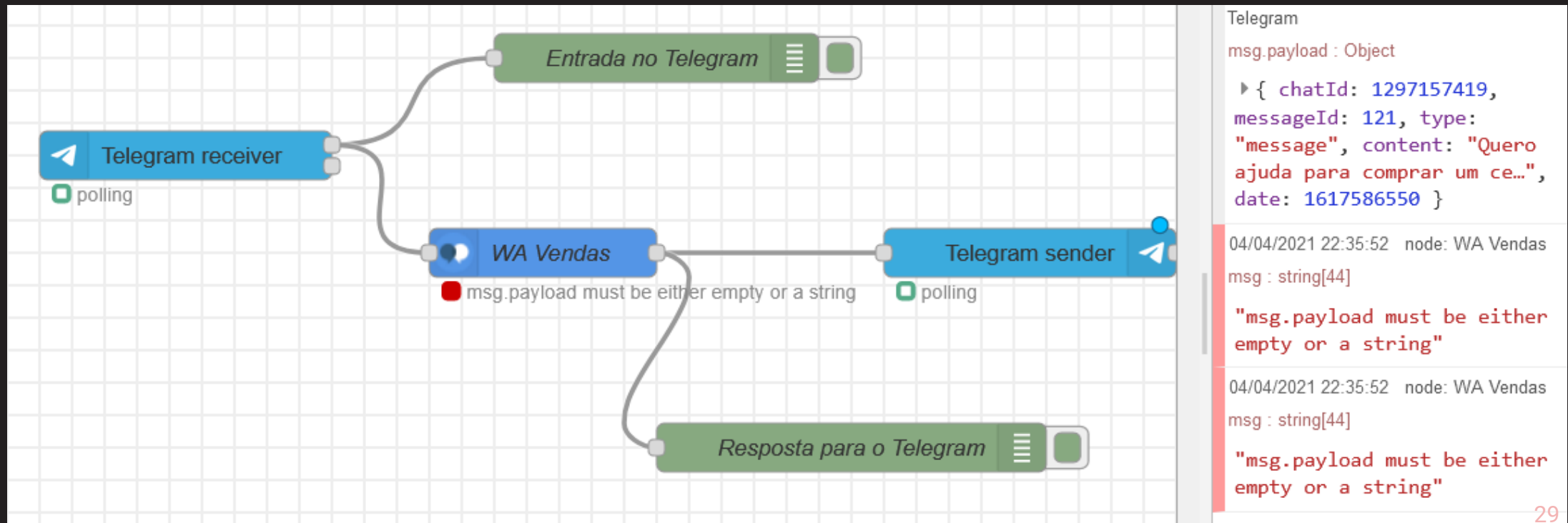
# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Adicionamos um nó de debug na saída do Watson Assistant;
- Teste o bot enviando uma mensagem pelo Telegram; O que aconteceu?



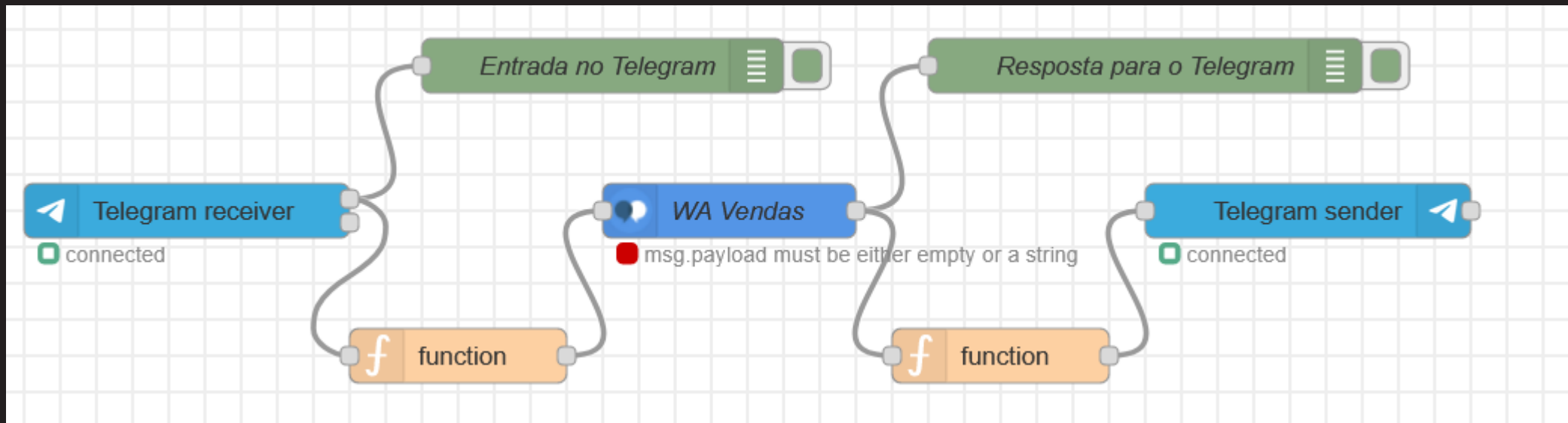
# Fazendo o Watson Assistant para processar as mensagens do Telegram

- A mensagem não está corretamente preparada; Devemos sempre nos atentar sobre o **padrão de mensagem que cada serviço/aplicativo gera e consome**;



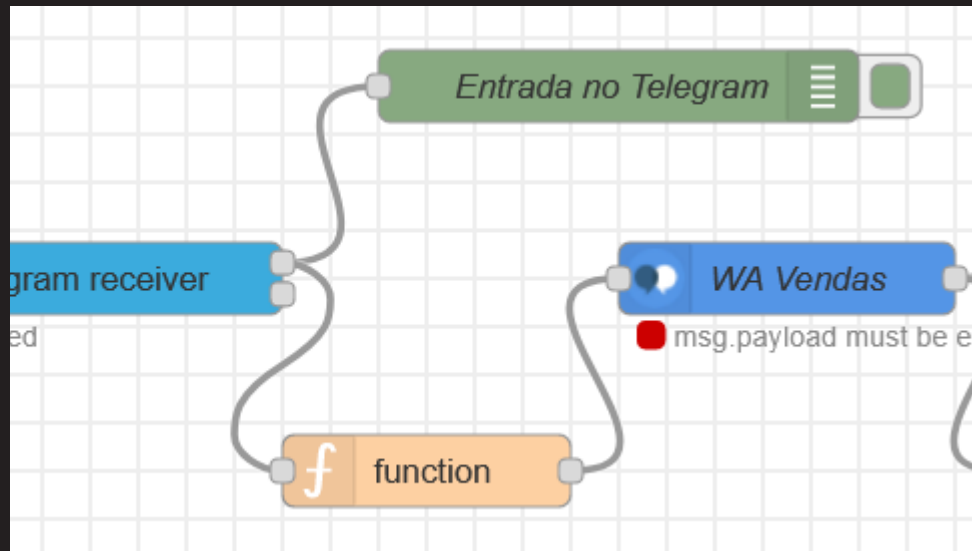
# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Vamos fazer como fizemos na primeira aula, adicionando dois nós de function, antes e depois do nó do WA:



# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Agora vamos inserir o código de alteração da mensagem em cada nó de function; Primeiro o de Entrada



Edit function node

Delete Cancel Done

Properties

Name Proc Entrada

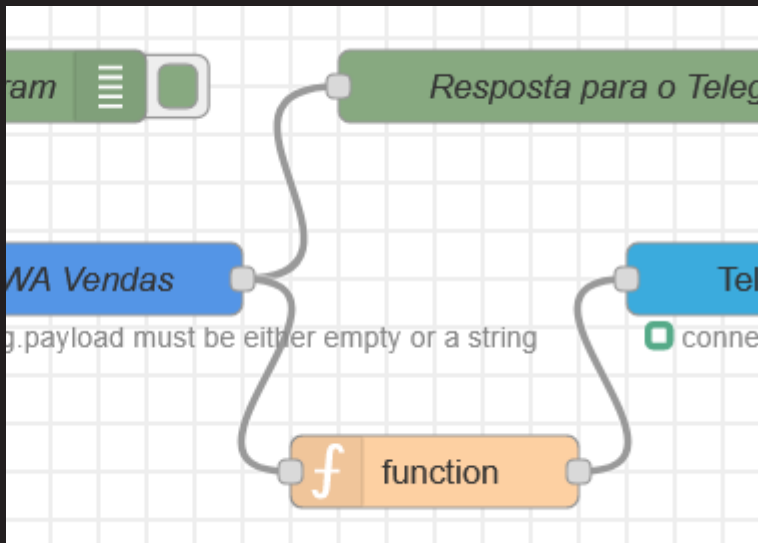
Setup Function Close

```
1 msg.params = {
2   "session_id" : msg.payload.session_id !== 'undefined' ? msg.payload
3 };
4 msg.chatId = msg.payload.chatId;
5 msg.payload = msg.payload.content;
6 return msg;
```

Outputs 1

# Fazendo o Watson Assistant para processar as mensagens do Telegram

- Agora vamos inserir o código de alteração da mensagem em cada nó de function; Agora o de Saída (retorno)



Edit function node

Delete Cancel Done

⚙ Properties

Name Proc Retorno

Setup Function Close

```
1 msg.payload = {
2   chatId : msg.chatId,
3   type : "message",
4   content : msg.payload.output.generic[0].text
5 };
6 return msg;
```



# Resultado



**Henrique**

Quero ajuda para comprar um celular



**Vendas**

Legal vou te ajudar. Irei fazer algumas perguntas para selecionar o melhor produto para você, ok?

O que acontece se for respondido positivamente a pergunta do bot?

# Resultado

HF Henrique  
sim

VE Vendas  
Eu não entendi. Você pode tentar reformular a frase.

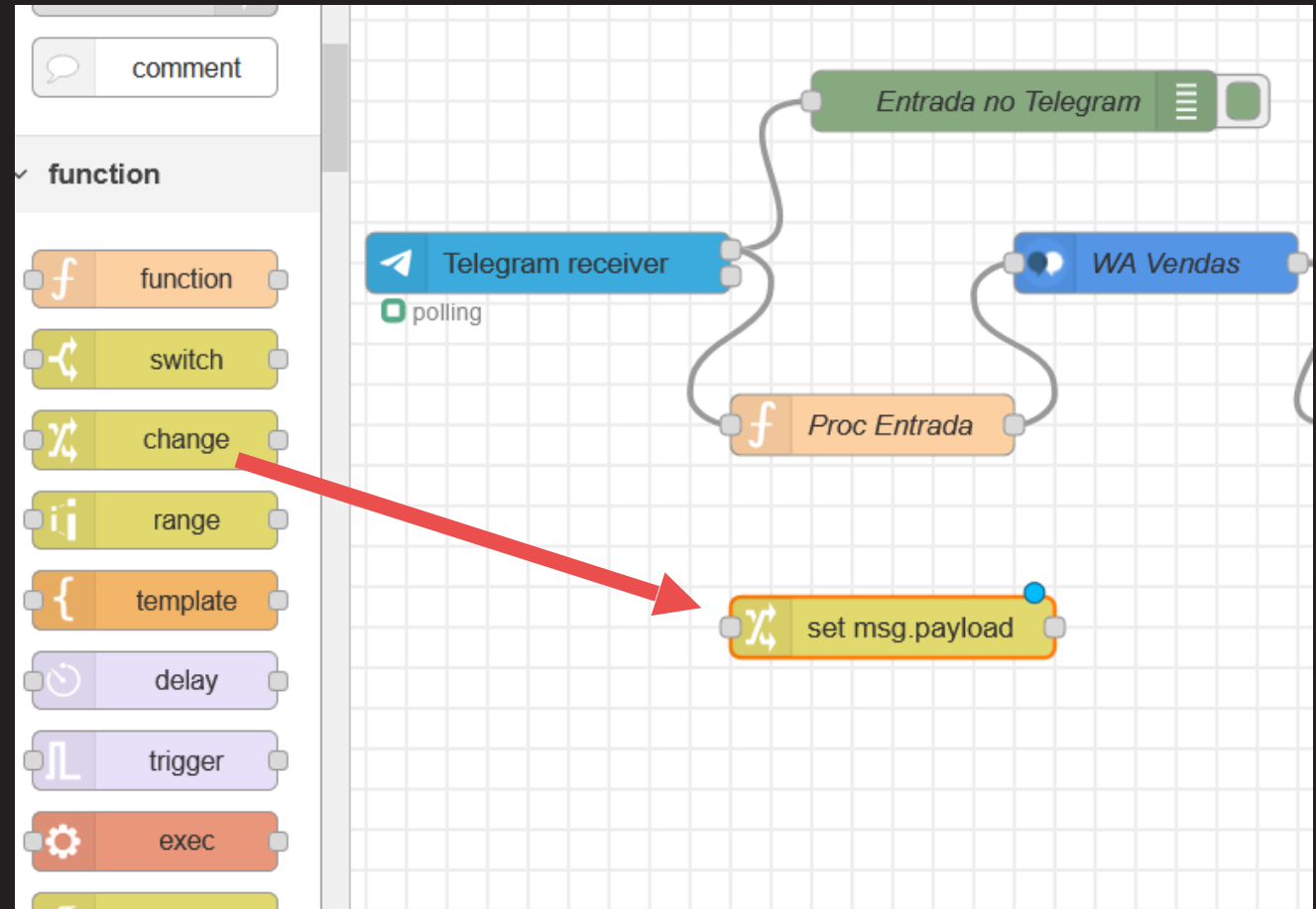
Isso acontece pois o **bot não sabe que está falando com a mesma pessoa**. Como nosso bot não está configurado para identificar um simples “sim” no início da árvore de diálogo, então ele irá cair no nó de **Em outros casos** (condição de `anything_else`)

# Integrando nosso bot 2

Mantendo o contexto de identificação de usuário

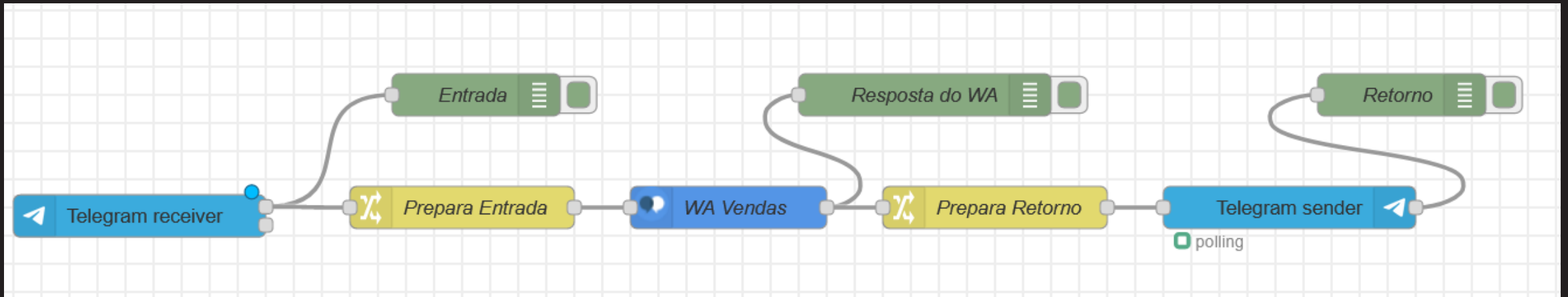
# Fluxo de integração: mantendo o ID do usuário

- Vamos seguir a seguinte abordagem: usar o ID do Telegram como ID do Watson Assistant. Para isso vamos usar um nó de function especial, o **change**;
- Vamos trocar os dois nós de function por novos dois nós de change;



# Fluxo de integração: mantendo o ID do usuário

- O fluxo ficar assim:



# Fluxo de integração: mantendo o ID do usuário

- Duple clique no primeiro nó;
- Vamos adicionar as seguintes regras de configuração:

**Edit change node**

Delete Cancel Done

**Properties**

Name: Prepara Entrada

**Rules**

- Set msg.params.session\_id to msg.payload.chatId
- Set msg.chatId to msg.payload.chatId
- Set msg.payload to msg.payload.content

+ add

# Fluxo de integração: mantendo o ID do usuário

- Duple clique no segundo nó;
- Vamos adicionar as seguintes regras de configuração:

**Edit change node**

Delete Cancel Done

**Properties**

Name: Prepara Retorno

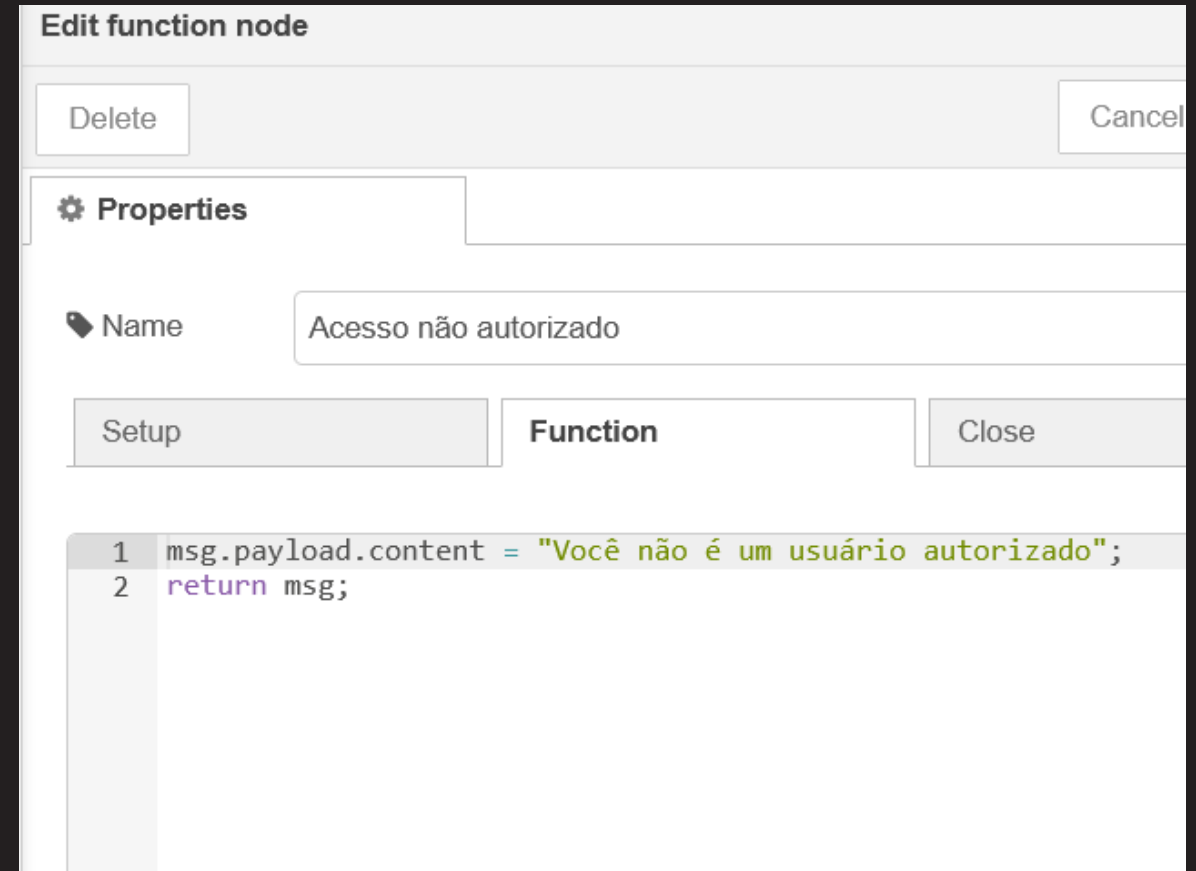
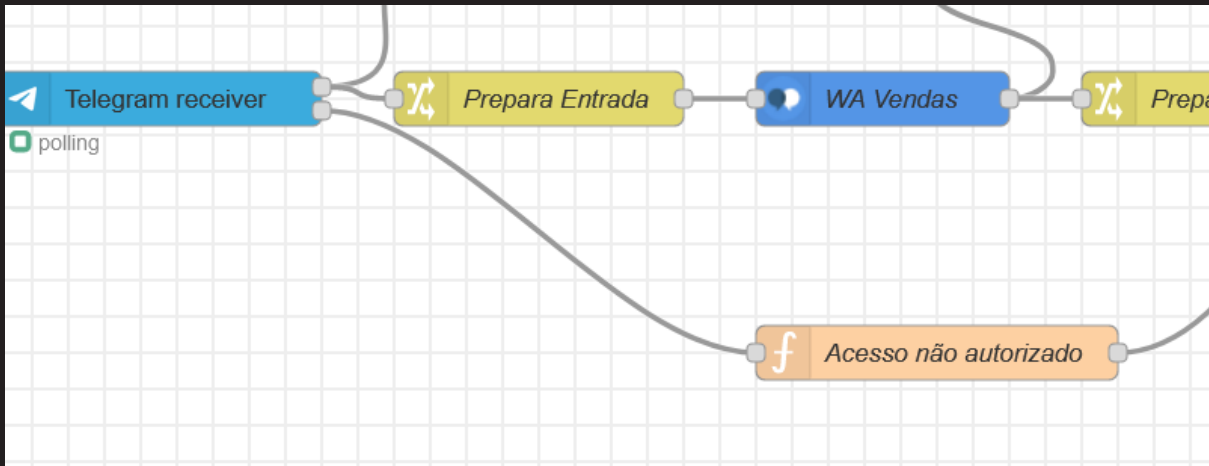
**Rules**

- Set `msg.payload.chatId` to `msg.chatId`
- Set `msg.payload.type` to `message`
- Set `msg.payload.content` to `msg.payload.output.generic[0].text`

+ add

# Fluxo de integração: mantendo o ID do usuário

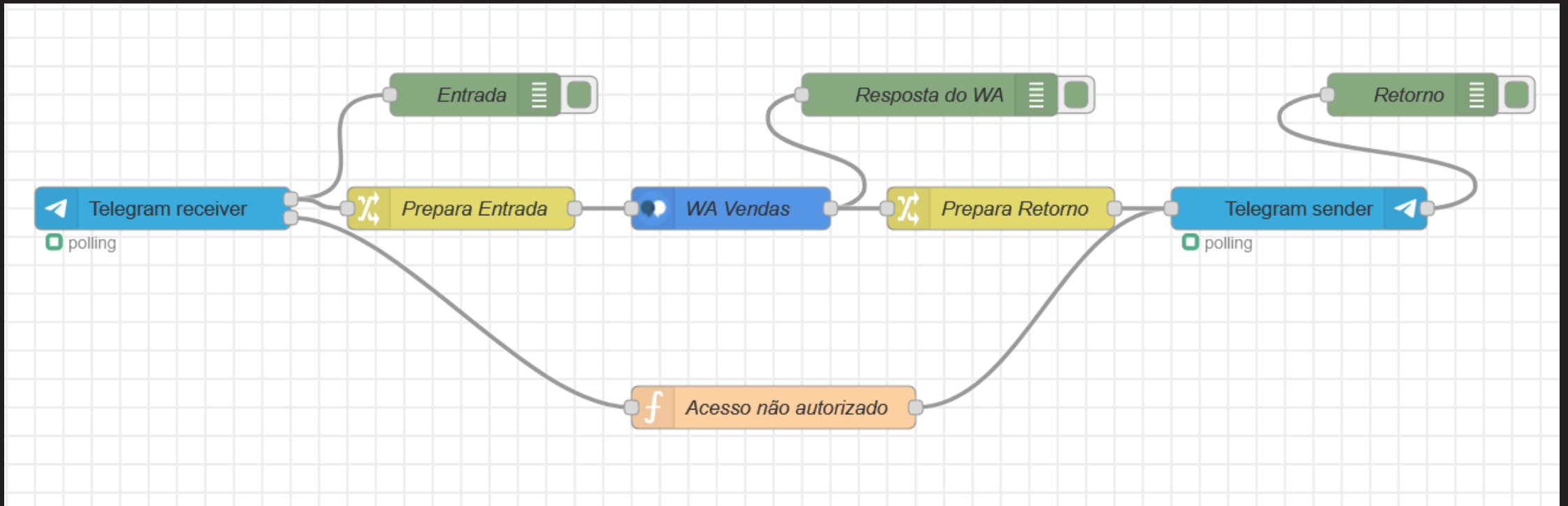
- Vamos adicionar um nó para negar acesso não autorizado:





# Fluxo de integração: mantendo o ID do usuário

- O fluxo final será assim:



# Testando...

- Salve o fluxo final!
- Observação: se você estiver tendo problemas, certifique-se que há apenas um fluxo com nós do Telegram;

The screenshot shows a Telegram chat interface with a white background. The chat history includes the following messages:

- Henrique (HF):** /start
- Henrique (HF):** Quero ajuda para comprar um celular
- Vendas (VE):** Legal vou te ajudar. Irei fazer algumas perguntas para selecionar o melhor produto para você, ok?
- Henrique (HF):** sim
- Vendas (VE):** Você tem preferência por alguma marca de celular?
- Henrique (HF):** da samsung
- Vendas (VE):** E você tem algum valor máximo para comprar esse produto?

# Descanso

Do professor =D

# Exercícios

1. Mande um emoticon para seu bot. Observe a mensagem pelos nós de Debug para entender o que acontece com ela durante o fluxo de processamento. E se você mandar uma imagem? São criados novos atributos (parâmetros) dentro da mensagem do Telegram? Como o Watson Assistant lida com isso?
2. Explore as propriedades das mensagens para entender como elas são passadas de um nó para o outro no Node-RED. Tente fazer o rastreamento do ID de usuário ao longo do fluxo.

**Copyright © 2025**

**Slides do Prof. Daniel Sória,  
com adaptações dos slides dos Prof. Henrique Ferreira - FIAP**

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