DOCUMENTO DE PROJETO
PROJETO MARVEL

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - 1º TDDS

Lilian Akemi Ofusa Miyamura

INDICE

1. DECLARAÇÃO DE ESCOPO 3
2. ACESSO A SOLUÇÃO 3
3. FUNCIONAMENTO DOS SERVIÇOS UTILIZADOS 8
4. DETALHAMENTO DE CONFIGURAÇÃO DE CADA SERVIÇO E DO FLUXO NODE – RED 8

1. DECLARAÇÃO DE ESCOPO

O Bot Uatu foi criado com o foco em disseminar para todas as idades o conhecimento sobre o universo da Marvel, explicando a história, quem são seus inimigos, poderes e outras curiosidades dos personagens. O nome “Uatu” é inspirado em um dos Vigias personagens do universo da Marvel que são os observadores do universo, eles armazenam tudo, mas não podem interferir nas atitudes dos seres e Uatu é o Vigia que observa a Terra e o sistema solar.

O sistema disponibiliza um diálogo simples e o mais natural possível, permitindo mensagens de texto, imagem e voz.

O bot irá analisar a mensagem e retornar de formas diferentes suas respostas pelo fato de ser baseado nos recursos da Watson Assistant e Text to Speech para configurar e disponibilizar o diálogo e o Watson Visual Recognition para identificar as imagens dos personagens.

2. ACESSO A SOLUÇÃO

O Node – RED é conectado diretamente a ferramenta TELEGRAM, simplificando a conexão do chatbot e prevendo os assuntos e os rumos das conversas pré-programadas.

Ao invés de desenvolver a linguagem de integração com cada aplicativo, simplesmente basta integrá-lo a plataforma TELEGRAM, uma vez que ela faz a tradução da conversa para o App, podendo aperfeiçoar o desempenho para os atendimentos posteriores.

Primeiro Passo: Criação do Bot no Telegram – O @BotFather auxilia a criação de um novo bot ou alterar a configuração de um bot já criado.

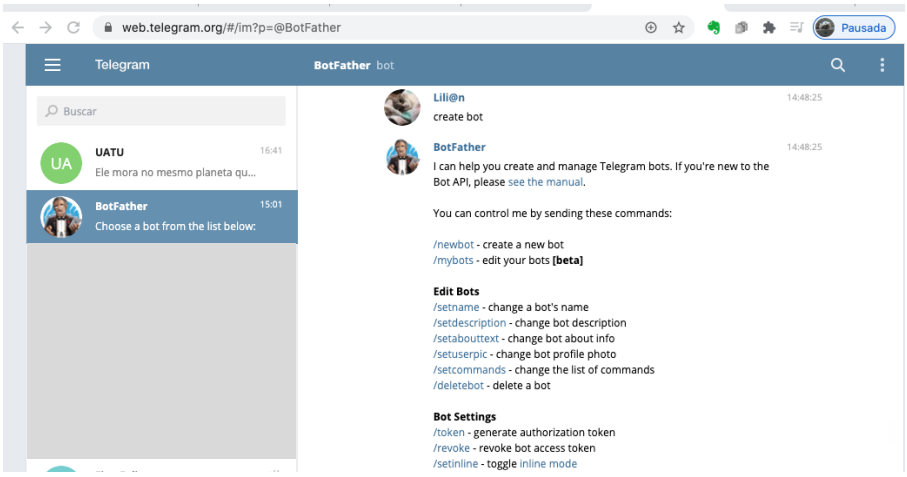


Figura 01: : Exemplo de configuração no Telegram

Após a validação do nome do Bot escolhido – UATU, foi criado um nome de usuário – MarvelEagleBot - que possibilitará a busca. Após as boas vindas do BotFather, foi fornecido um token que é um código necessário para configurar o bot do Telegram no Node-Red .

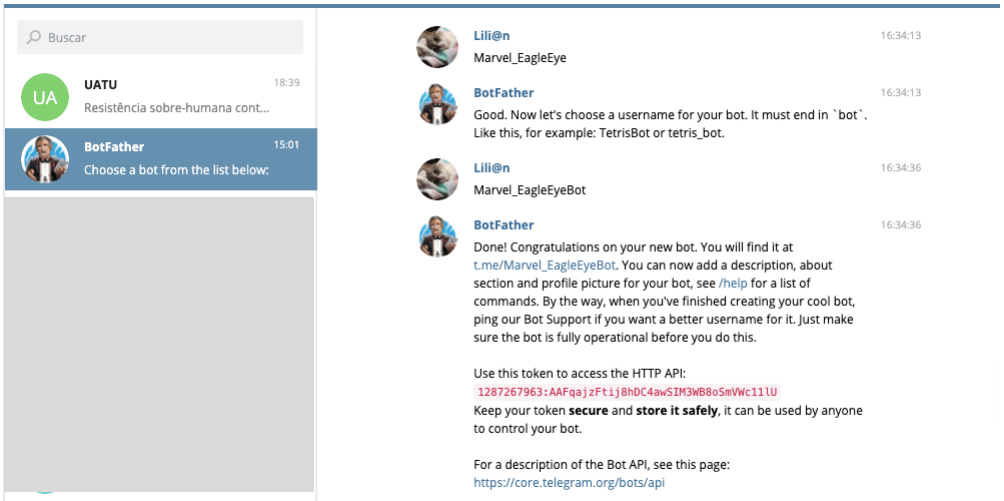


Figura 02: : Exemplo de configuração no Telegram

Segundo Passo: Edição dos nodes no Telegram – Com o nome e o usuário fornecido pelo BOT, configuramos o node Telegram Receiver e o node Telegram Sender e conectamos com o node visual recognition, watson assistant e speech to text – vide maiores detalhes de configuração no item 4 – detalhamento de configuração de cada serviço e do fluxo Node-Red.

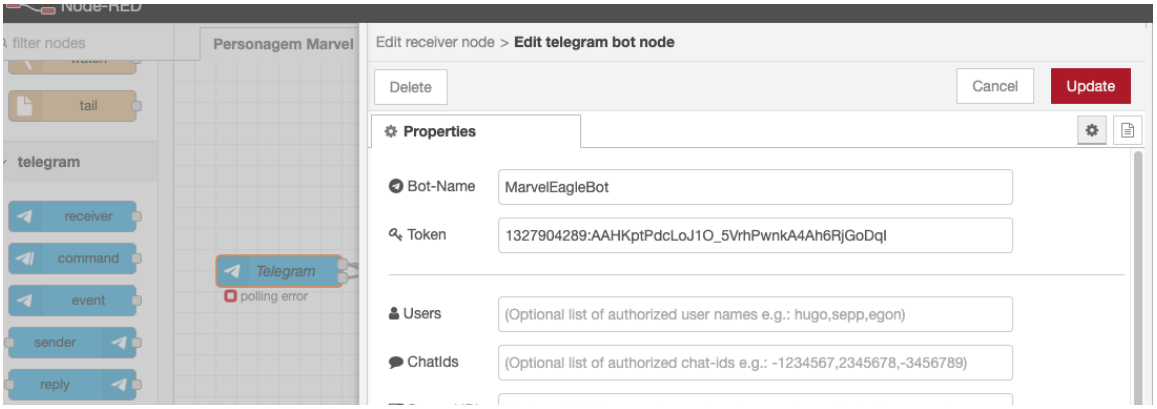


Figura 03 : Exemplo de configuração no Node Telegram

Exemplos de utilização no Telegram – Reconhecimento Visual

Com o sistema criado e integrado, o nosso BOT reconhece as fotos dos personagens Marvel enviados para ele.

Figura 04 : Exemplo de utilização no Telegram

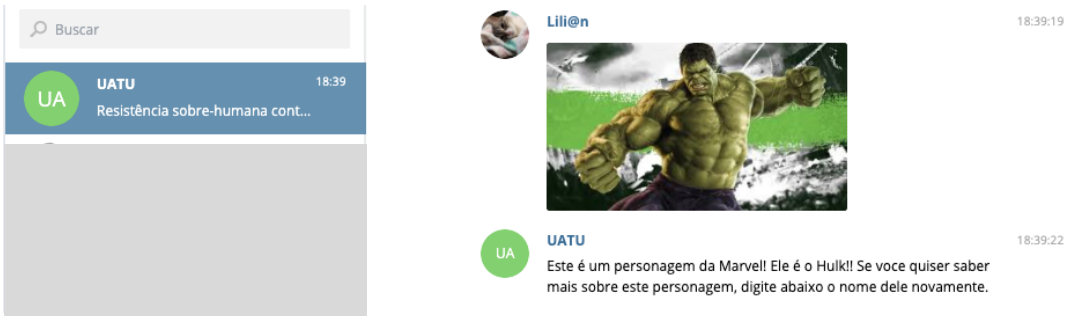


Figura 05 : Exemplo de utilização no Telegram

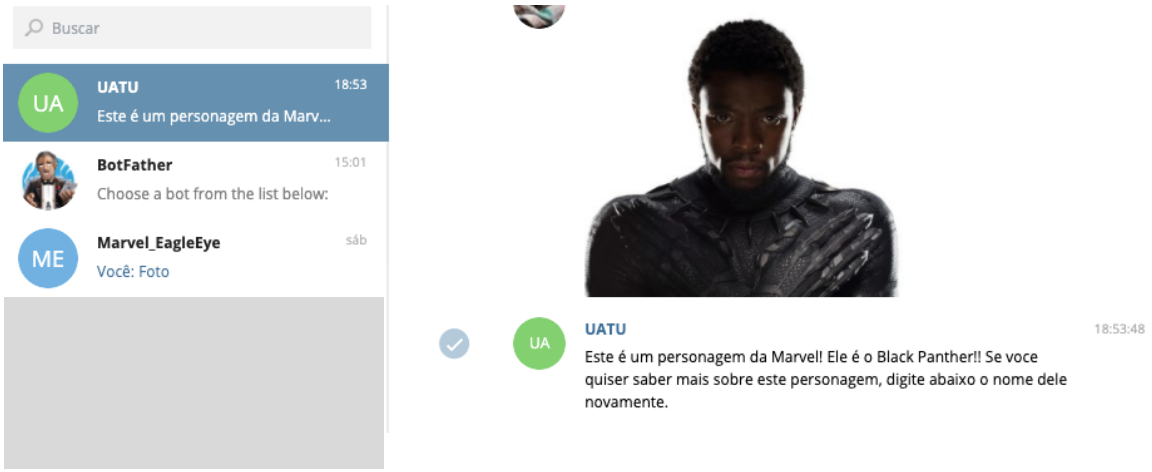


Figura 06 : Exemplo de utilização no Telegram

Figura 07 : Exemplo de utilização no Telegram

Exemplos de utilização no Telegram – Reconhecimento Watson Assistant

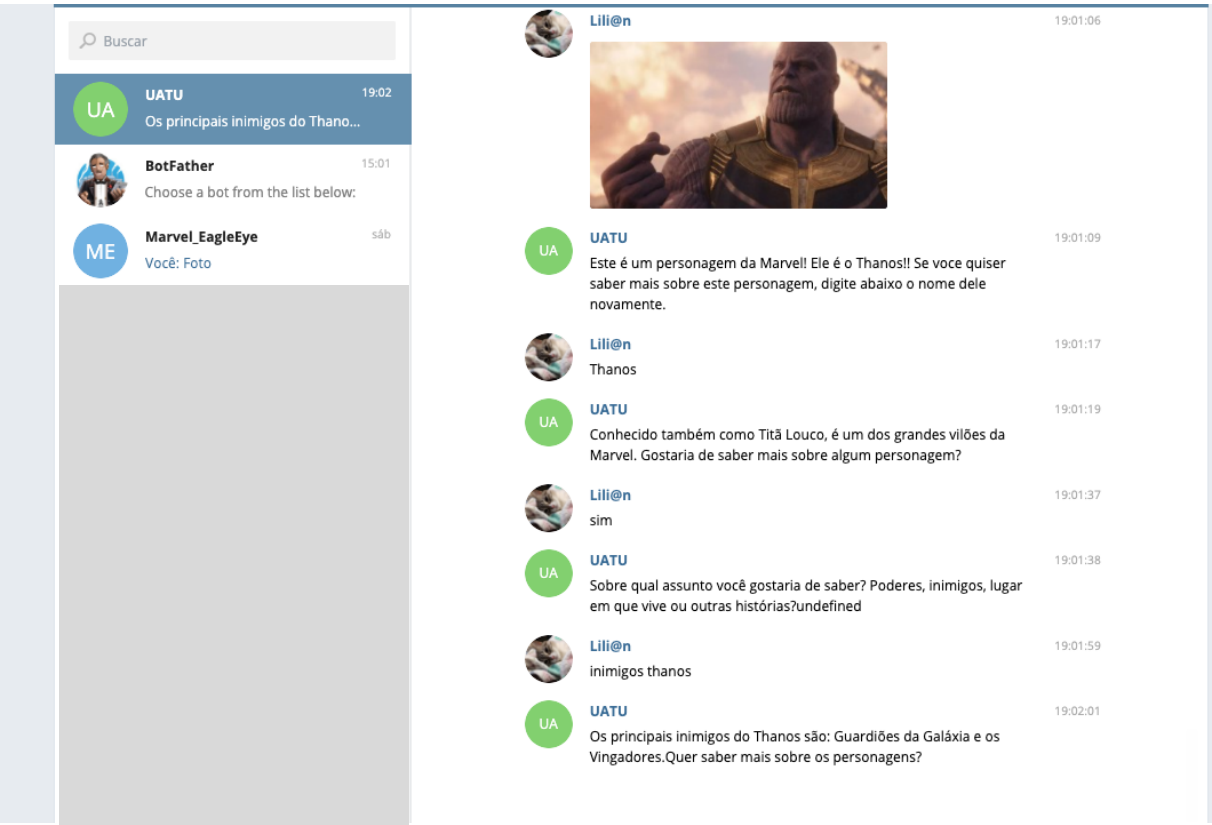


Figura 08 : Exemplo de utilização no Telegram

Exemplos de utilização no Telegram – Speech to Text

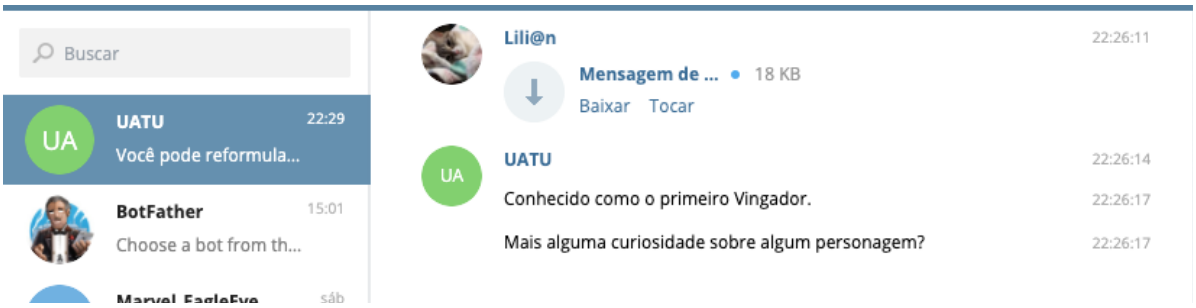


Figura 09 : Exemplo de utilização no Telegram

3. FUNCIONAMENTO DOS SERVIÇOS UTILIZADOS

A IBM Cloud é uma plataforma que disponibiliza diversos serviços com customização de uso para cada cliente de maneira individual atendendo às necessidades de seus clientes de forma personalizada.

Através dela é possível compartilhar, criar aplicativos, armazenar dados, documentos, softwares, entre outras funções permitindo assim a portabilidade de acesso a toda essa gama de possibilidades em Nuvem a qualquer momento e local, sem a necessidade de uma infraestrutura. Dessa forma seus clientes têm a possibilidade de diminuição de custos de manutenção necessários para se manter esses produtos de forma independente, além de permitir que tenhamos acesso as tecnologias mais recentes de forma mais rápida.

Dentre as diversas funções podemos citar o armazenamento de dados que garante de forma segura o armazenamento de backups de seus clientes otimizando esse processo oneroso e preocupante. Podemos também utilizar o serviço de banco de dados conjuntamente a de desenvolvimento de aplicativos o que facilita para as startups seus processos de criação de forma mais ágil e funcional além de permitir o gerenciamento de forma mais eficaz dos processos das empresas de diversos setores, a IBM Cloud conta ainda com serviços de segurança de rede, aplicativos e infraestrutura que seus clientes possam precisar.

Para encerrar não podemos esquecer do IBM Watson que é uma ferramenta completa para construção de um modelo de IA, gerando de forma integrada uma gama enorme de possibilidades de customização e funções, desde transformação de voz em texto, como tradutor e reconhecimento de imagem até a possibilidade de análise de emoções e tonalidades em conteúdo escrito.

4. DETALHAMENTO DE CONFIGURAÇÃO DE CADA SERVIÇO E DO FLUXO NODE – RED

Neste item explicaremos o desenvolvimento de aplicações integrando serviços web, fontes de dados e dispositivos IoT com o uso do Node-RED

a) Conceito do Node-RED

O Node -RED é uma ferramenta para criação de aplicações IoT, que simplifica o desenvolvimento conectando blocos de códigos para execução de tarefas.

A principal característica da ferramenta é sua abordagem de programação visual e desenvolvimento de eventos em formas de nodes, que permite facilmente os

desenvolvedores a conectar blocos de códigos pré-definidos em uma IDE drag and drop, para executar uma determinada tarefa.

b) Execução do Node-RED

1º PASSO: Executamos o Terminal (prompt de comando) como administrador e rodar o comando node-red para iniciar o servidor Node-RED.

2º PASSO: Tendo iniciado o servidor, acessamos o endereço http://127.0.0.1:1880

```
Last login: Sat Aug 29 18:13:30 on ttys000
lilianofusa@MBP-de-Lilian ~ % node-red
30 Aug 15:09:22 - [info]

Welcome to Node-RED
===================

30 Aug 15:09:22 - [info] Node-RED version: v1.1.3
30 Aug 15:09:22 - [info] Node.js  version: v12.18.3
30 Aug 15:09:22 - [info] Darwin 19.6.0 x64 LE
30 Aug 15:09:22 - [info] Loading palette nodes
30 Aug 15:09:25 - [info] Settings file  : /Users/lilianofusa/.node-red/settings.js
30 Aug 15:09:25 - [info] Context store  : 'default' [module=memory]
30 Aug 15:09:25 - [info] User directory : /Users/lilianofusa/.node-red
30 Aug 15:09:25 - [warn] Projects disabled : editorTheme.projects.enabled=false
30 Aug 15:09:25 - [info] Flows file     : /Users/lilianofusa/.node-red/flows_MBP-de-Lilian.json
30 Aug 15:09:25 - [info] Server now running at http://127.0.0.1:1880/
30 Aug 15:09:25 - [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.
---------------------------------------------------------------------

30 Aug 15:09:25 - [info] Starting flows
30 Aug 15:09:25 - [info] Started flows
```

Figura 10 : servidor Node-RED

c) Visão geral da interface Node-RED

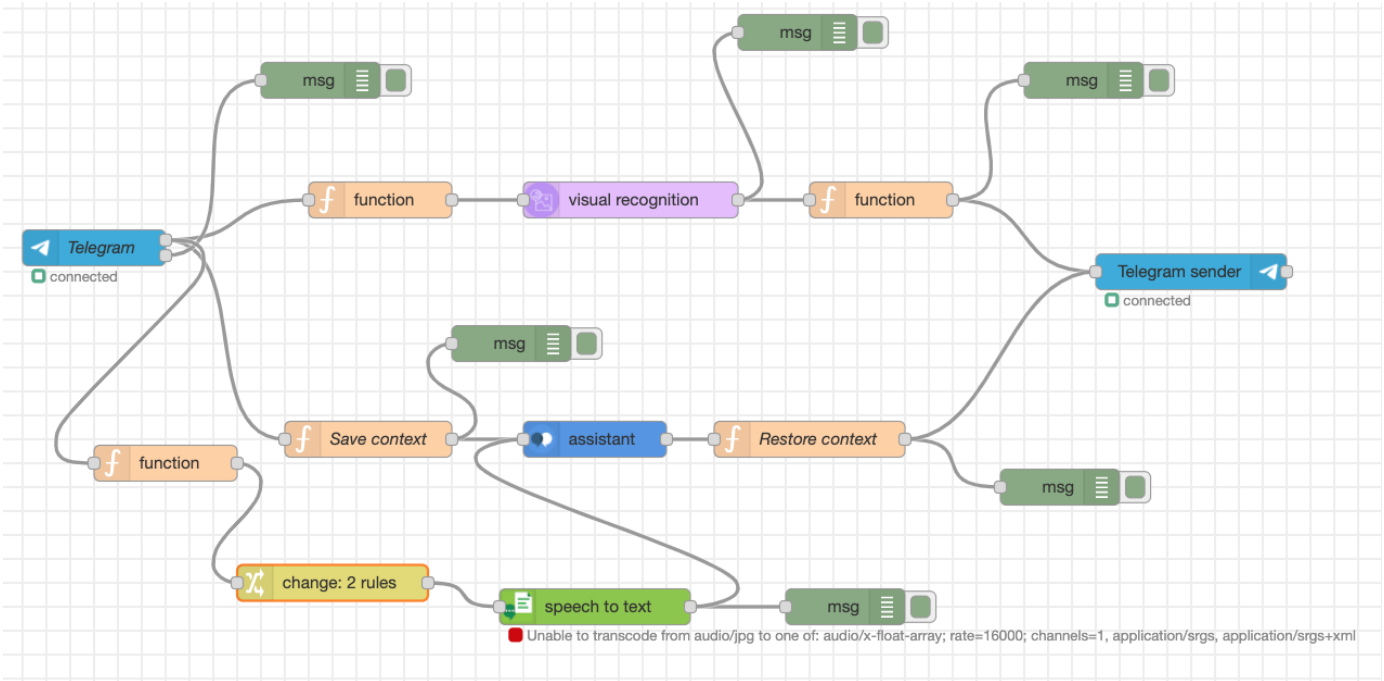


Figura 11 : Fluxo Node – RED projeto Personagem Marvel

d) Visual Recognition e Configuração do nó no Node-RED

O serviço da IBM Watson Recognition tem como objetivo principal, identificar, classificar e procurar conteúdos visuais usando machine learning. Com ele é possível reconhecer imagens, comidas, objetos, cores e outros conteúdos. Além disso é possível criar e treinar seu próprio classificador de imagens.
Esse recurso pode ser implementado em diversas áreas, alguns exemplos de caso de uso no mercado são no setor de manufatura, auditoria visual, comércios e outros.

https://www.ibm.com/br-pt/cloud/watson-visual-recognition

Em IBM Cloud, Services - Visual Recognition:

Criação de um modelo que é responsável por analisar a imagem, separar por classes e classificar.

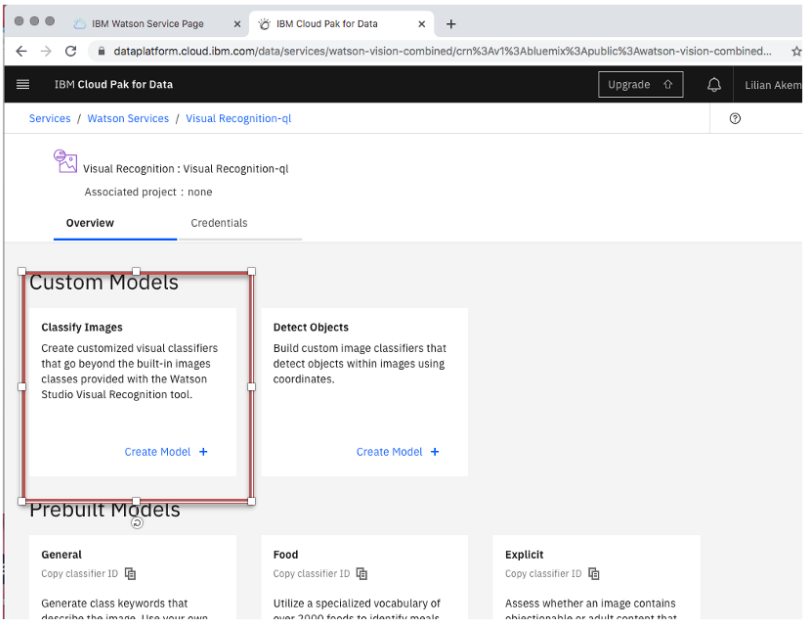


Figura 12 : Visual Recognition na IBM Cloud

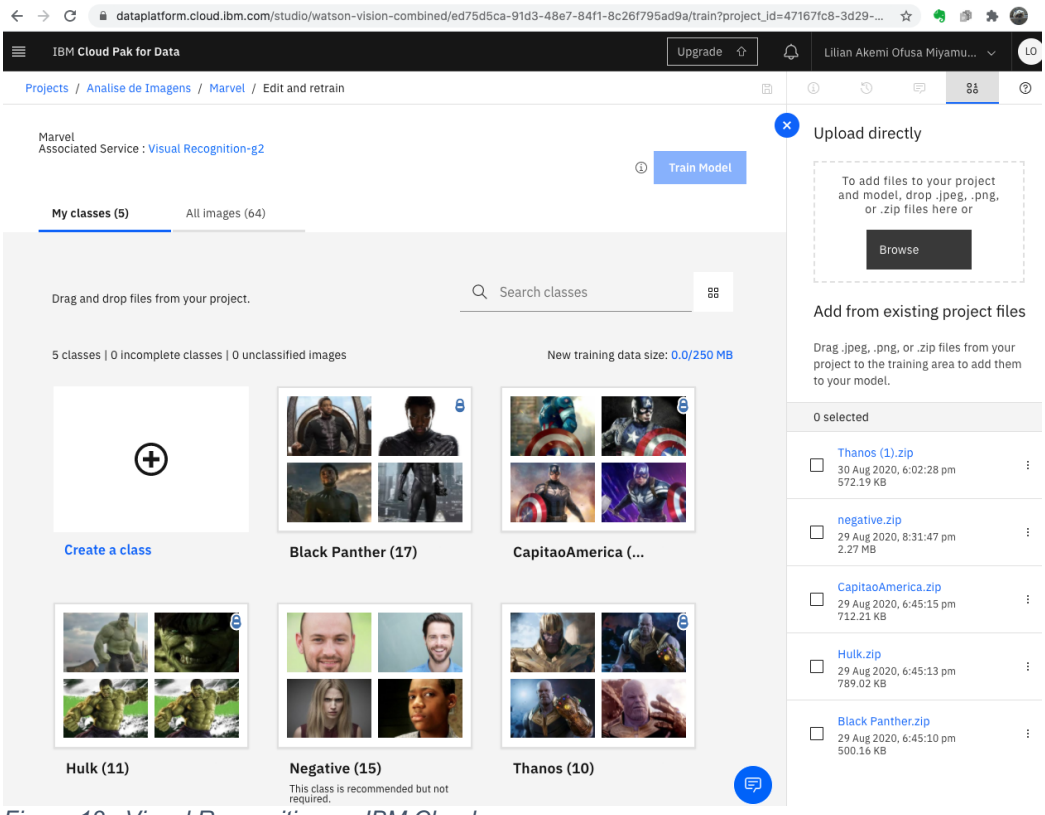


Figura 13 : Visual Recognition na IBM Cloud

Classifier ID:

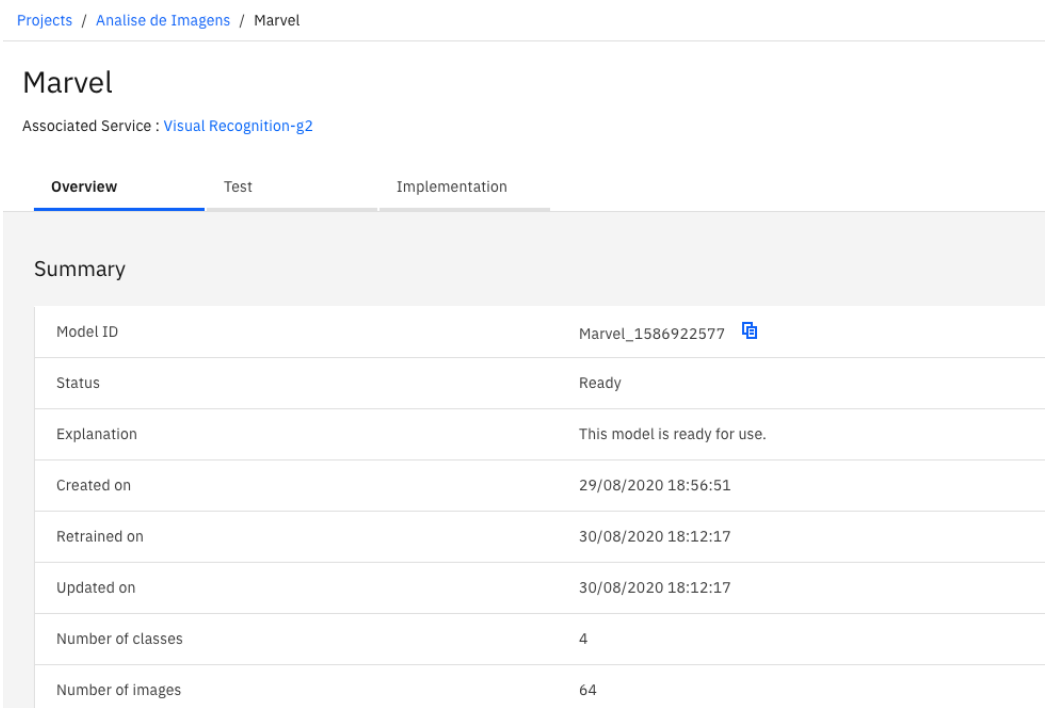


Figura 13: Classifier ID no Visual Recognition na IBM Cloud

Configuração do nó Visual Recognition no Node-RED:

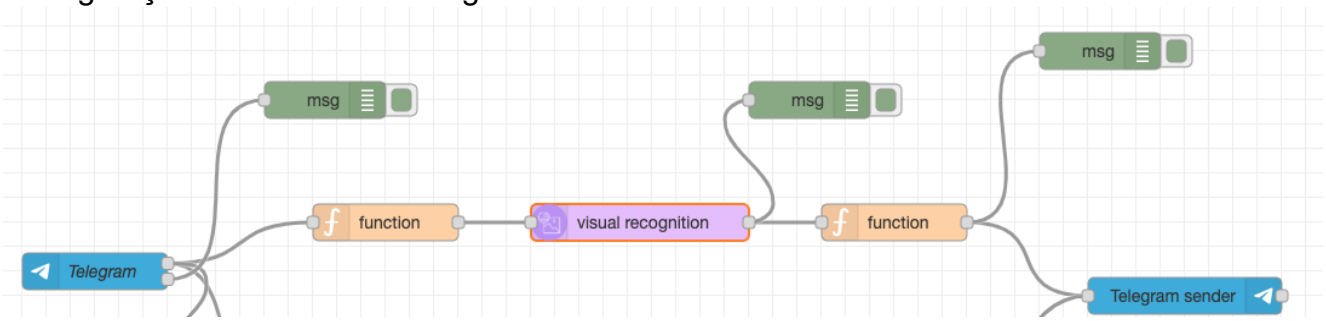


Figura 14: Visão geral do nó Visual Recognition no Node-RED

- Function **antes** do nó Visual Recognition

```
msg.telegramChatID= msg.payload.chatId;
msg.payload = msg.payload.weblink;

msg.params = {
   classifier_ids: ["Marvel_1586922577"],
   // threshold: 0.5
}
```

```
return msg;
```

- Function **depois** do nó Visual Recognition -

```
models=msg.result.images[0].classifiers

str=[]

if(models){
    models.forEach(function (model,idx1) {
        if (model.classes){
            model.classes.forEach(function(classes,idx2){
                str.push(classes.class)

            });
        }
    });
    output="Este é um personagem da Marvel! Ele é o " + str.join(', ')+ "!! Se você quiser saber mais sobre este personagem, digite abaixo o nome dele novamente."
}

if(str.length===0){
    output="Desculpe! Não reconheci como personagem Marvel!"
}
msg.payload={
    chatId: msg.telegramChatID,
    content:output,
    type: "message"
}

return msg;
```

e) Watson Assistant e configuração do nó Assistant no Node-RED

Watson Assistent - Bot Uatu

Dentro do Watson Assistent, trabalhamos com informações e curiosidades sobre os personagens da Marvel que escolhemos, e para isso implementamos quatro intenções, que o bot Uatu responderá e dará continuidade ao diálogo no telegram.

São eles: os poderes, os inimigos, lugares onde vivem e por último uma resumida história de cada personagem.

O usuário poderá iniciar o diálogo indicando apenas o nome do personagem escolhido onde o bot trará uma informação geral sobre o personagem, ou então poderá iniciar já indicando o assunto desejado para ter acesso às informações.

No fluxo do diálogo se o usuário optar em saber sobre as histórias dos personagens, ele poderá indicar apenas a intenção e em seguida após será solicitado o personagem que deseja escolher para continuar o fluxo, ou então poderá indicar a intenção juntamente com o nome do personagem desejado (exemplo: “Quero saber sobre a história do Pantera Negra”) para ir direto a história desse personagem. Já dentro desse nó, haverá etapas onde será perguntado ao usuário se deseja continuar com o fluxo da história, se ele optar por interromper, será direcionado (*Jump to*) ao *Agradecimento* onde poderá escolher outros assuntos sobre os personagens, ou então encerrar a conversa.

No caso do usuário escolher saber sobre os poderes que os personagens tem, ele entrará no nó da intenção #poderes, e poderá escolher o personagem que deseja e em seguida receber as informações, ou então uma outra forma de entrar nesse nó é indicando diretamente a intenção junto com o nome do personagem desejado e assim encontrará a resposta diretamente (exemplo: “Quero saber sobre os poderes do Hulk”). Ao término da explicação será direcionado (*Jump to*) ao *agradecimento* onde será perguntado se ele irá querer saber mais informações sobre os personagens, ou terminar a conversa.

Uma outra possibilidade é a intenção #inimigos, onde mostrará os principais inimigos de cada personagem. Indicando apenas a intenção, será perguntado em seguida de qual personagem o usuário deseja saber, ou então poderá indicar juntamente a intenção e o nome do personagem (exemplo: “Quais são os inimigos do Capitão América?”) que o bot irá reconhecer e responderá as informações. No

final haverá também o direcionamento (*Jump to*) ao *Agradecimento*, onde ele poderá escolher outras curiosidades ou então encerrar o diálogo.

A última intenção que o nosso bot conseguirá responder é #lugra_que_habita, onde mostrará em que lugar os personagens da Marvel vivem. Igualmente como está organizado os nós com as intenções #poderes e #inimigos, o usuário conseguirá acessar as informações indicando apenas a intenção e depois indicar qual o perguntado escolhido, ou então indicando juntamente o nome do personagem mais a intenção para ir diretamente para as informações. No final também haverá o direcionamento (*Jump to*) ao *Agradecimento.*

No final do diálogo, o nó de *Agradecimento* será importante para dar a opção ao usuário a escolher outros assuntos e curiosidades sobre os personagens. Ele poderá escolher de uma forma direta, indicando o assunto junto com o nome do personagem, o bot irá direcionar o usuário para a informação desejada, ou então poderá escolher primeiramente o assunto e em seguida o bot perguntará qual o personagem desejado, para assim ser direcionar ao ponto escolhido. E será nesse nó também onde o usuário pode encerrar a conversa se ele não quiser dar continuidade ao diálogo.

Tabela de intenções

Intenções	Valores	Descrição

#historia	- Qual a história do personagem? - Qual a história? - Quero saber da história - Quero saber sobre a história. - Gostaria de saber sobre a história.	Essa intenção irá reconhecer que o usuário está interessado em saber sobre as histórias dos personagens. O bot irá mostrar uma resumida história do personagem.
#inimigos	- Gostaria de saber quem é o inimigo - Inimigos? - Quais os inimigos? - Qual o principal inimigo?	O bot irá reconhecer que o usuário quer informações sobres os inimigos dos personagens e irá informá-lo.
#lugar_que_habita	- Gostaria de saber em que lugar mora - Gostaria de saber onde mora - Lugar em que vive? - Lugar que mora? - Qual lugar ele vive? - Que planeta vive?	A intenção será utilizada quando o usuário apontar que deseja saber onde os personagens vivem.

#poderes	- Super poderes? - Quero saber sobre os poderes - Que força ele tem? - Qual o poder? - Quais os poderes? - Gostaria de saber quais os poderes	A intenção irá reconhecer quando o usuário usar frases que apontam o interesse em saber sobre os poderes que os personagens têm.

Tabela de entidades

Entidades	Valores	Descrição
@personagens	- **Hulk** (O cara verde, Bruce Banner, Cicatriz verde, Gigante Esmeralda, O incrível Hulk, Gigante verde, Verde) - **Pantera Negra** (Black Panther, Pantera, T'Chaka) - **Capitão América** (Capitão, Homem sem pátria, Captain America, Steve Rogers, Cap) - **Thanos** (O Titã Louco, Avatar da Morte, Queixos, Amante da Morte)	Essa entidade irá auxiliar o bot a identificar de qual personagem o usuário está se referindo, e assim trazer informações corretas para as perguntas.
@sim_nao	- **Sim** (yes, yep, sim, s, claro, com certeza, quero sim, gostaria) - **Não** (no, n, naum, não valeu, não obrigada, não quero, nop)	Essa entidade irá auxiliar no fluxo de alguns pontos da conversa, onde o usuário conseguirá decidir qual caminho quer seguir no diálogo.

Fluxo do diálogo no Watson Assistant

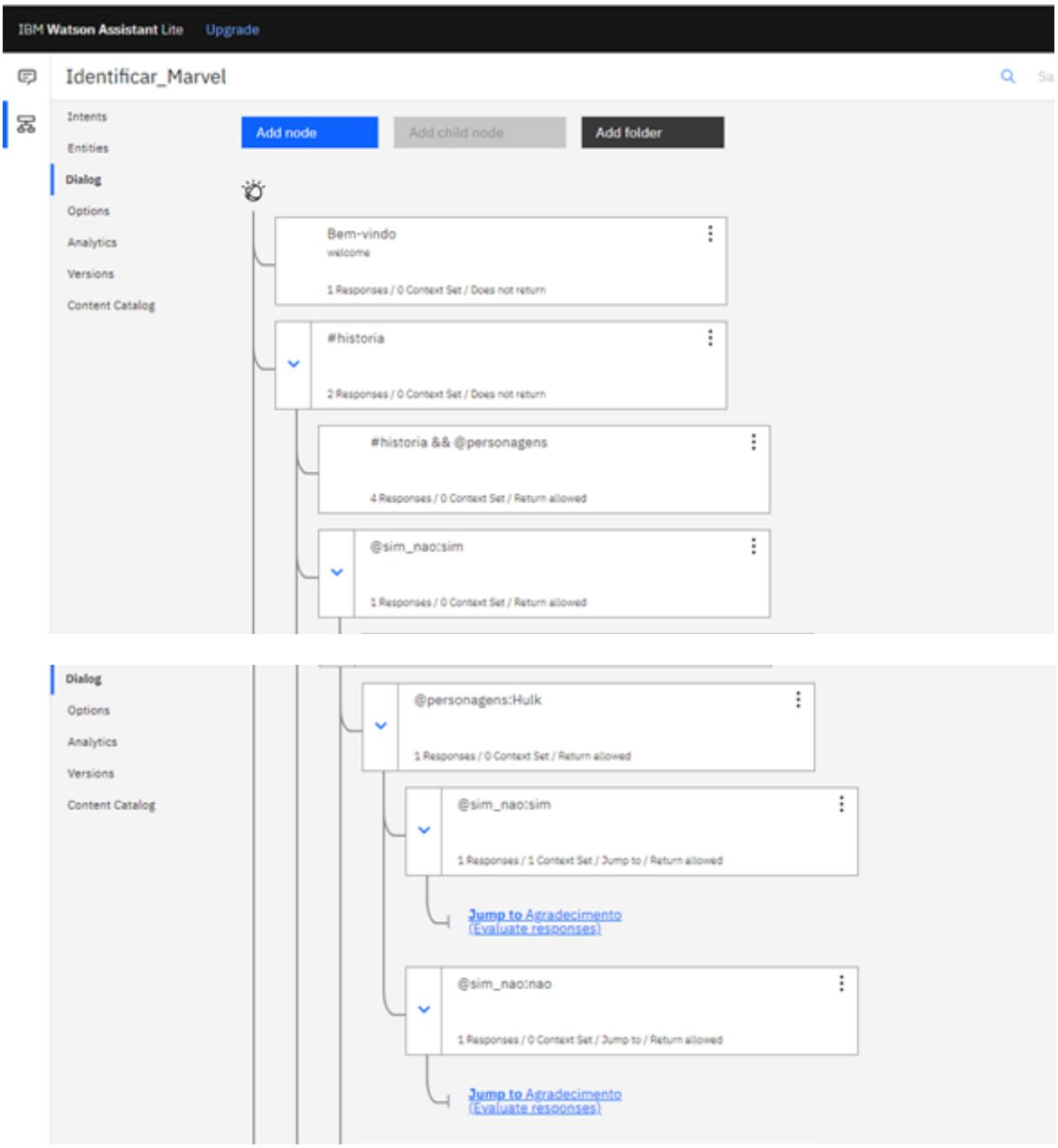


Figura 15: Visão fluxo no Watson Assistant

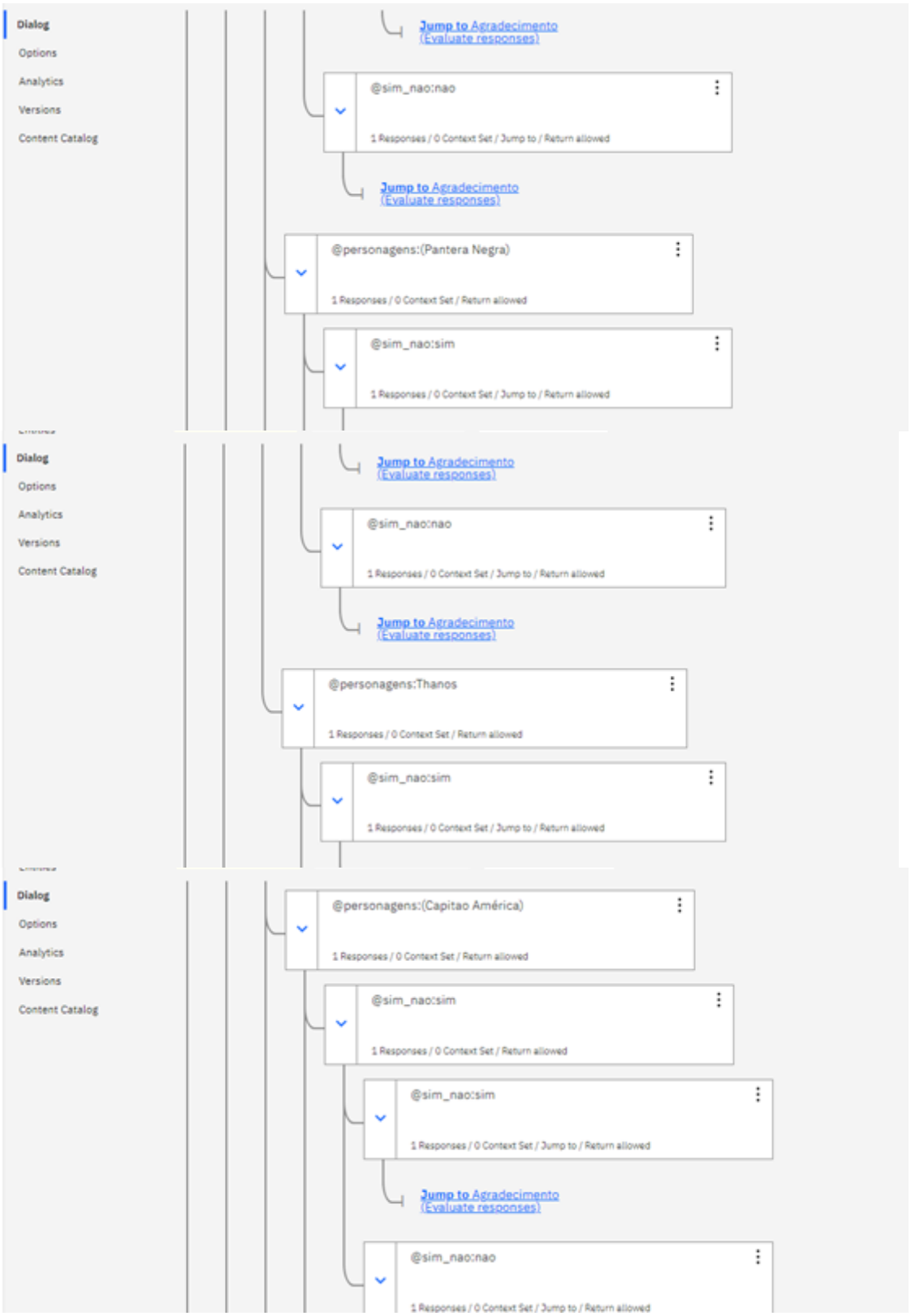


Figura 15: Visão fluxo no Watson Assistant

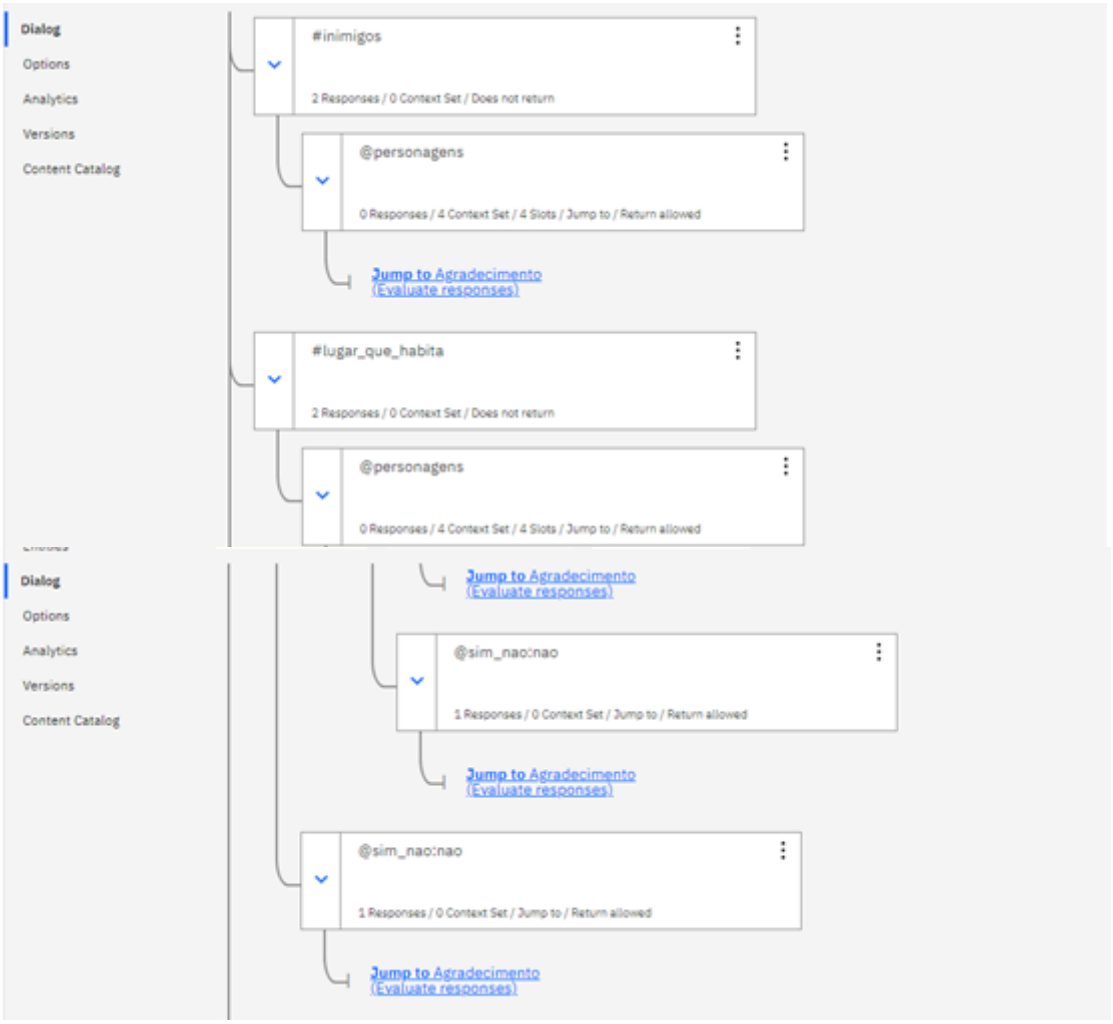
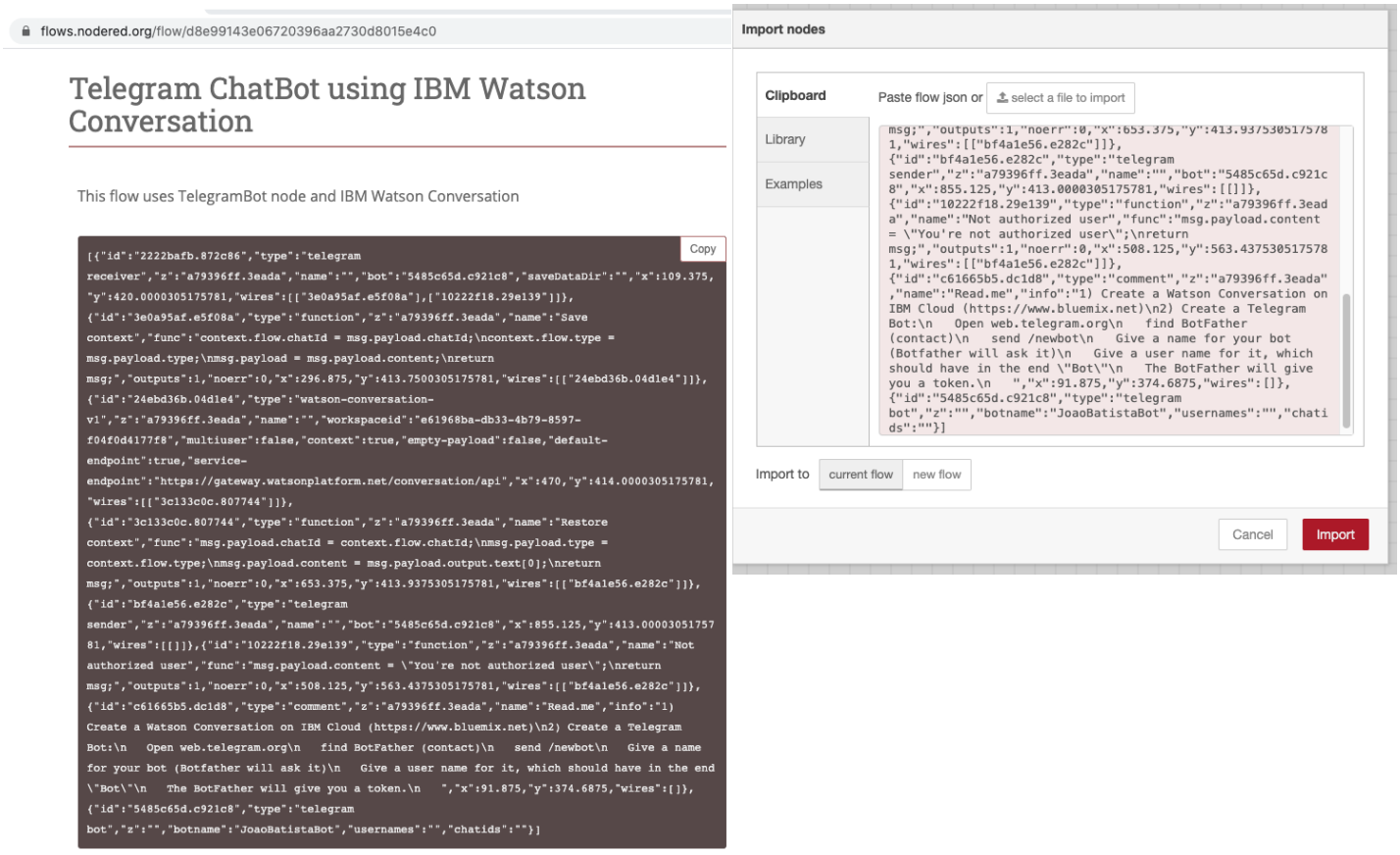


Figura 16: Visão fluxo no Watson Assistant

Configuração do nó no Watson Assistant no Node-RED:

Utilizamos como referência o video do video do Sergio Gama, https://www.youtube.com/watch?v=LOX3C6_upu4, para obter um flow de integração do Watson Assistant ao Telegram usando Node-red como orquestrador.

Dentro da biblioteca do site : https://flows.nodered.org/flow, obtemos o código e importamos dentro do nosso projeto "Personagem Marvel"



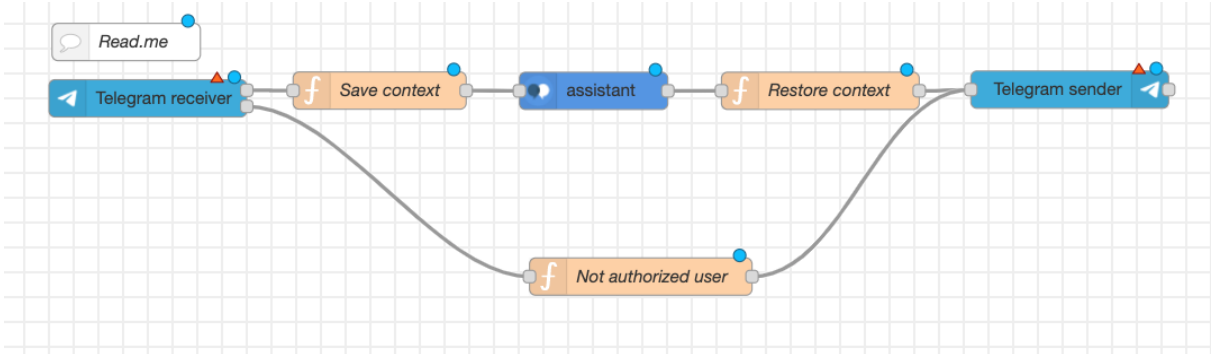
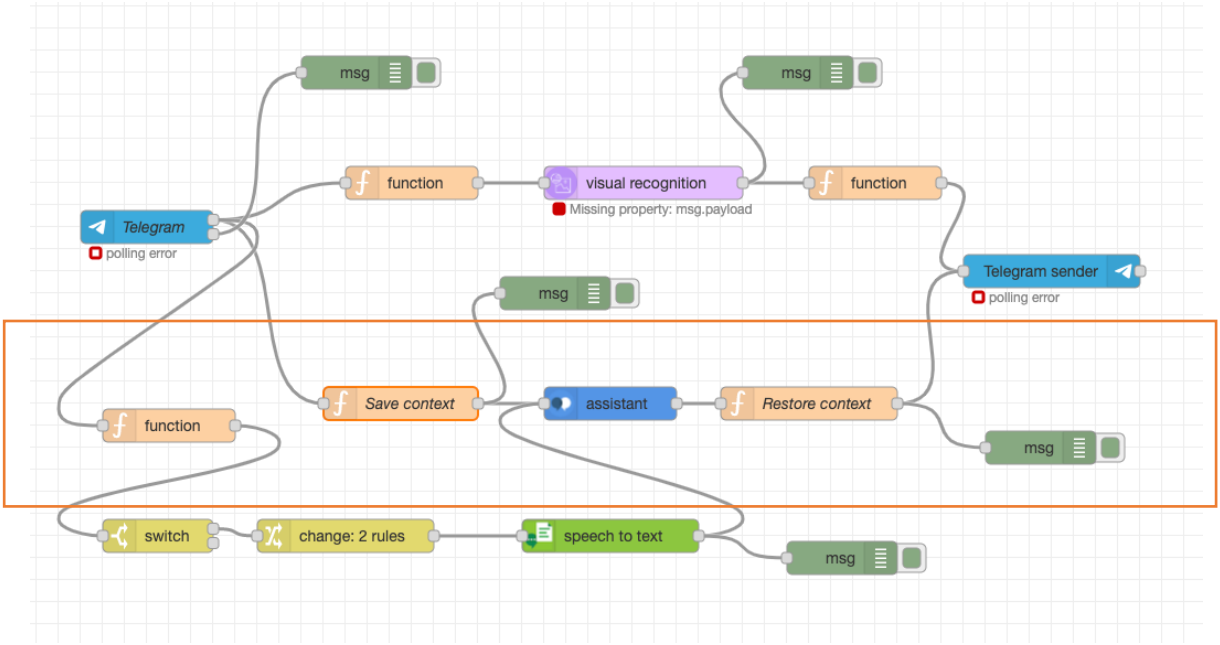


Figura 16: Exemplo de flow no site https://flows.nodered.org/flow

Flow projeto Watson Assistant Personagem Marvel:



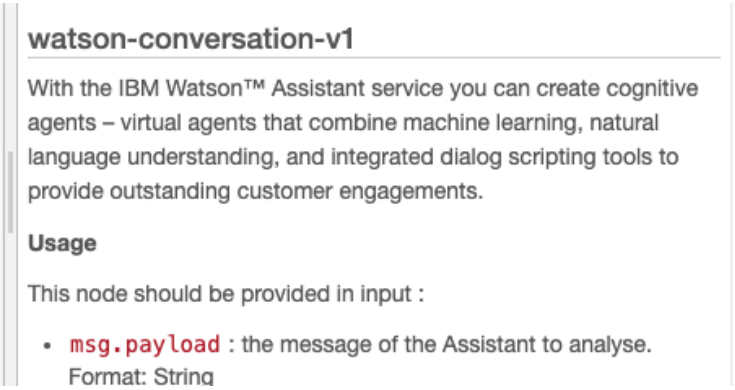
- Function **antes** do nó Watson Assistant

context.flow.chatId = msg.payload.chatId;
context.flow.type = msg.payload.type;
msg.payload = msg.payload.content;
return msg;

O "context" está disponível apenas para os nós de função é usado para armazenar dados na memória que podem ser acessados posteriormente e joga o *content* dentro do objeto msg.payload.

Existem 3 tipos de context :
■ Contexto local (context) - que pode ser acessado apenas pelo nó que criou o índice
■ Contexto do fluxo (flow) - que pode ser acessado por todos os nós de um fluxo
■ Contexto global (global) - que pode ser acessado por todos os nós de uma instância

O Watson estará esperando o msg.payload como mensagem.



- Function **depois** do nó Watson Assistant

```
for(x=0;x<msg.payload.output.text.length;x++)
{
   msg.payload.chatId = flow.get("chatId");
   msg.payload.type = "message"
   msg.payload.content = msg.payload.output.text[x];

}
return msg;
```

Modificando as propriedades utilizando o msg.payload, o nó de função aceita um objeto msg como entrada e pode retornar 0 ou mais objetos de mensagem como saída. Ao enviar apenas uma mensagem, ele entra num loop e cria mais mensagens necessárias.

Configuração do Nó Watson Assistant

Preenchimento de acordo com as informações fornecidas na IBM Cloud do Watson Assistant:

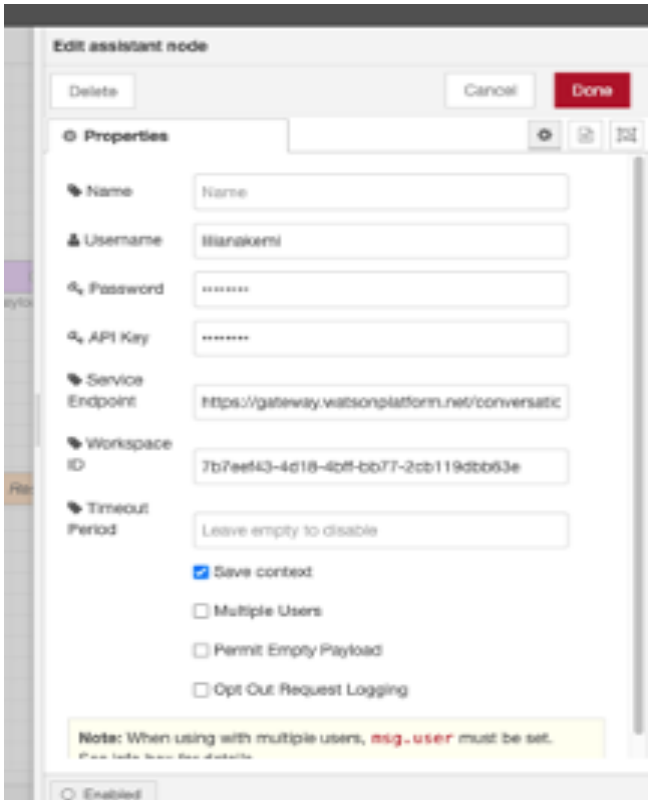


Figura 17: Propriedades do Nó Watson Assistant

f) Speech to Text e configuração do nó no Node-RED

O Watson Speech to Text é um ferramenta da IBM que permite transcrever textos através de um áudio. Ele consegue transcrever de forma rápida e compreender 7 idiomas, entre eles o português (Brasil) e o espanhol. Essa ferramenta pode ser implementada em diversos casos, como em centrais de atendimento e até mesmo promovendo facilidades em projetos para deficientes visuais.

https://www.ibm.com/br-pt/cloud/watson-speech-to-text

Configuração do nó no Speech to Text no Node-RED:

Dentro da biblioteca do Node-RED.org, utilizamos como referência do fluxo um flow de integração do no Speech to Text ao Telegram usando Node-red como orquestrador.

Dentro da biblioteca do site : https://flows.nodered.org/flow, obtemos o código e importamos dentro do nosso projeto "Personagem Marvel"

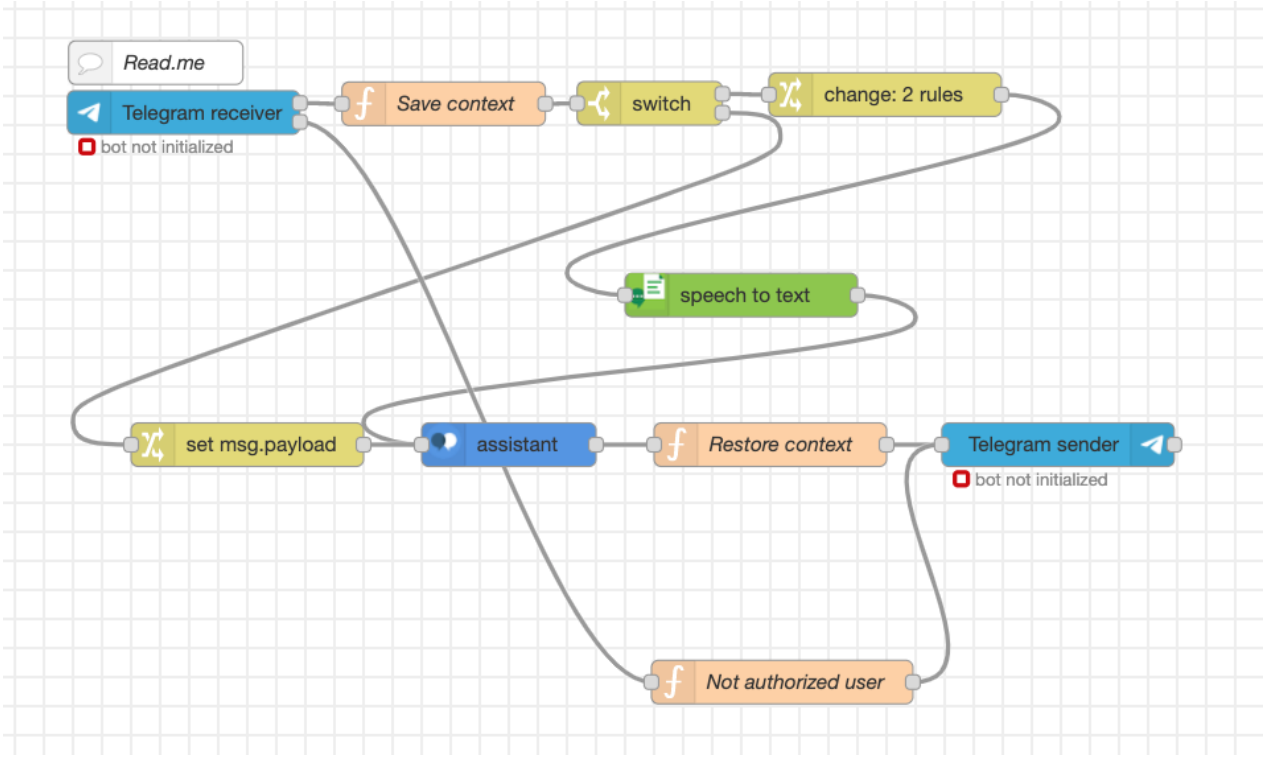


Figura 18: Exemplo de flow no site https://flows.nodered.org/flow

Flow projeto Speech to Text Personagem Marvel:

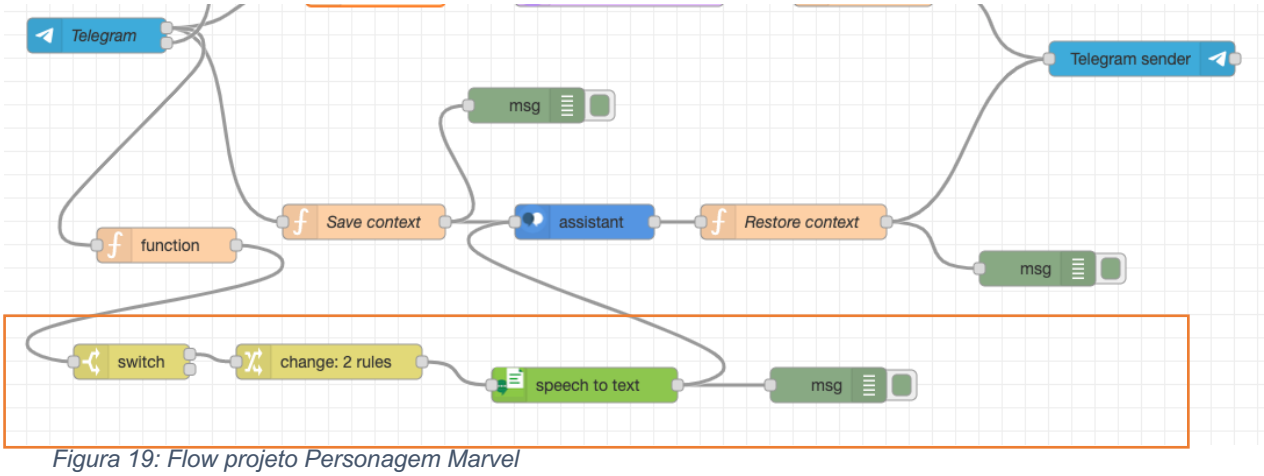


Figura 19: Flow projeto Personagem Marvel

- Function **antes** do nó Speech do Text

```
flow.set("chatId", msg.payload.chatId);
flow.set("type", msg.payload.type);
flow.set("content",msg.payload.content);
flow.set("weblink",msg.payload.weblink);
flow.set("audio",false);
return msg;
```

Propriedade flow.set - define uma propriedade de contexto com escopo do fluxo.