



PUBLICANDO O BOT

Alex Oliveira e Willian Farias

ONDE IREMOS PUBLICAR?

- WEB PAGE DA IBM



- TELEGRAM



PUBLICANDO O BOT: WEB PAGE DA IBM

Willian Farias



CRIANDO O ASSISTENTE (IBM)

IBM Watson **Assistant**

Assistants

Skills

Assistants

An assistant helps your customers complete tasks and get information faster. It may clarify requests, search for answers from a knowledge base, and can also direct your customer to a human if needed.

Create assistant

ADICIONANDO A BASE DE CONHECIMENTO AO ASSISTENTE

< Assistants

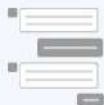
Pizzaria do Willian

Chatbot criado para o mini-curso da I Semana Acadêmica do IFS - Campus Lagarto.

Skill

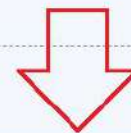
A dialog skill provides specific responses you've created. Choose one for your assistant. [Learn more](#)

Dialog



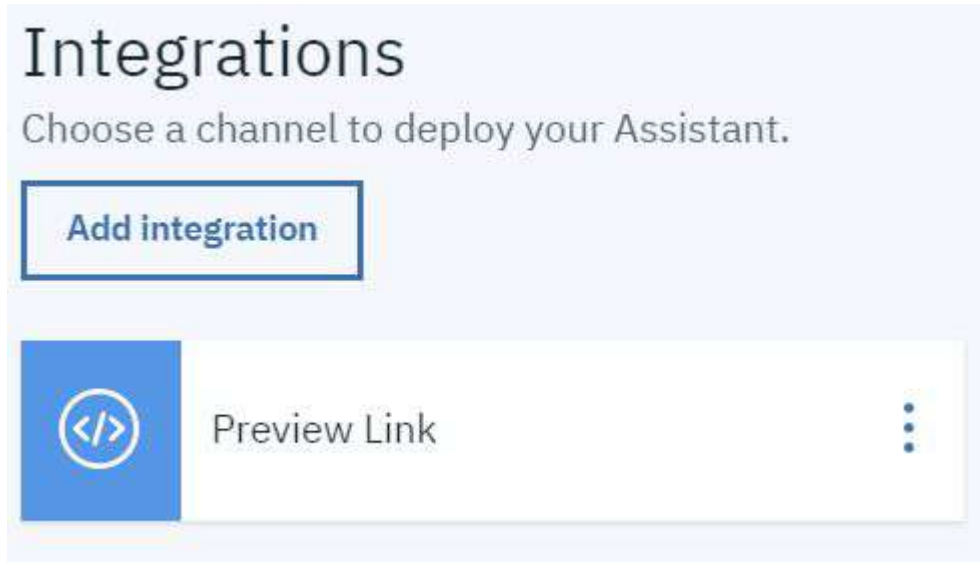
Add a dialog skill to design your conversation flow

Dialog skills use Watson natural language processing and machine learning technologies to understand user requests and respond appropriately.

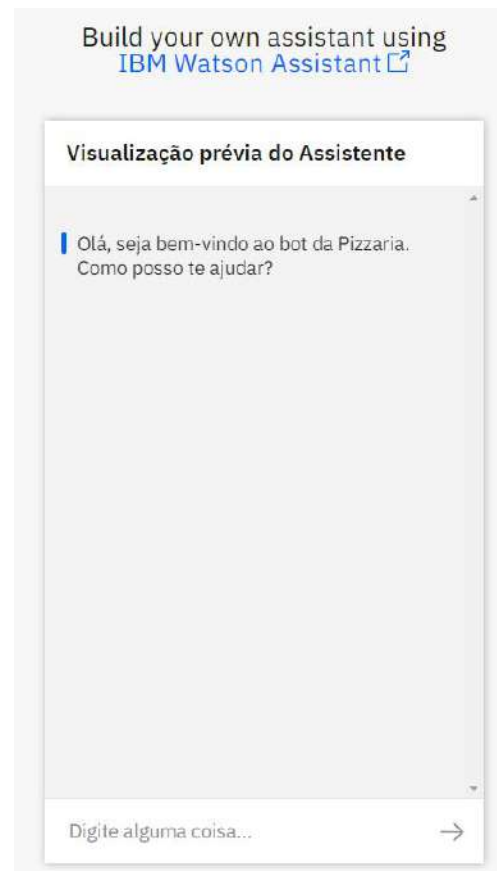


Add dialog skill

ACESSANDO O LINK DE PUBLICAÇÃO



PÁGINA DO BOT PUBLICADO (IBM)



INTEGRANDO O BOT: TELEGRAM

Alex Oliveira

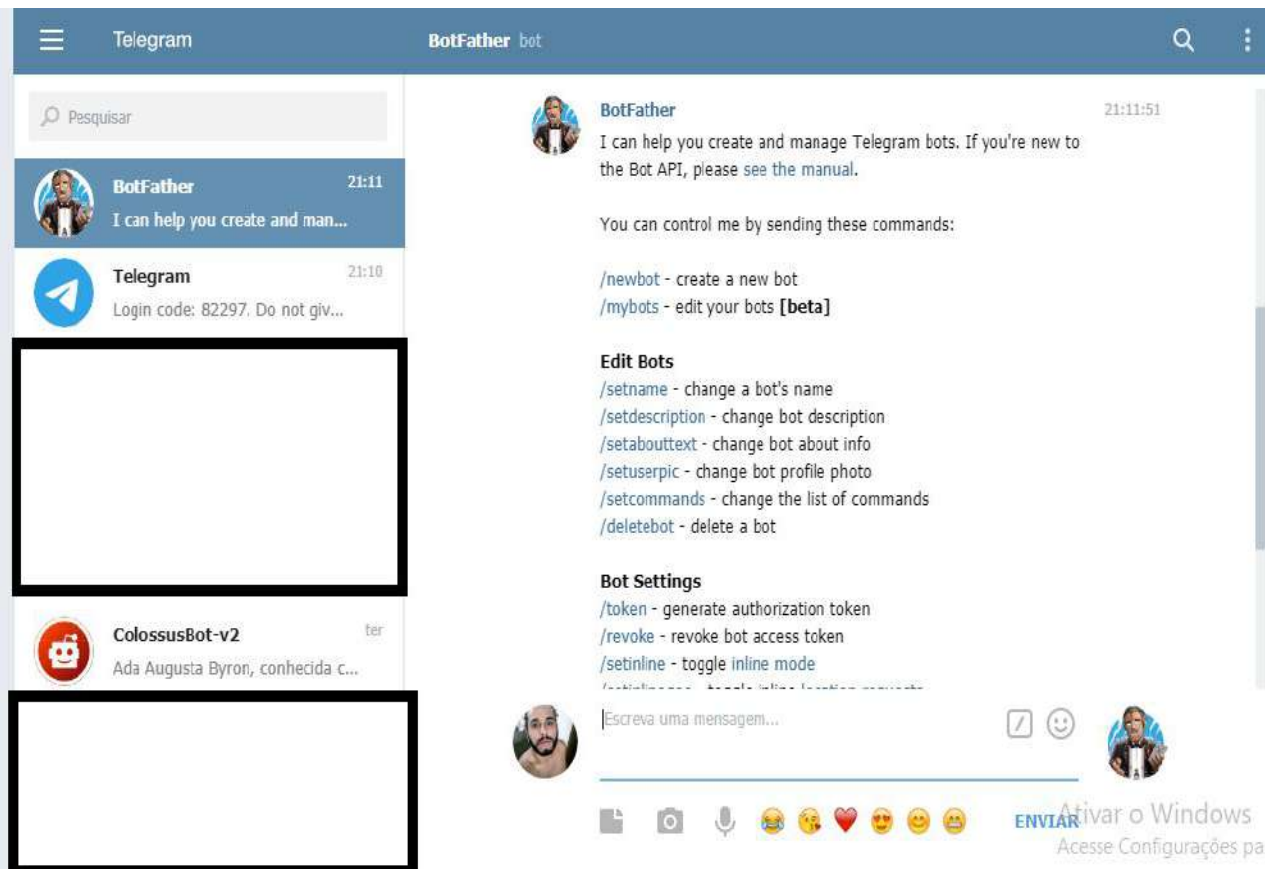


INTEGRAÇÃO COM O TELEGRAM

- O Bot criado no watson assistant pode ser integrado em vários ambientes como: Facebook Mensseger, Sites,etc.
- Para este Minicurso iremos demonstrar como integrar um Assistente virtual ao mensageiro instantâneo Telegram.

HANDS ON...

- Primeiramente é necessário **criar um bot no Telegram**, para que este bot criado seja vinculado ao assistente criado no Watson anteriormente e então consumir todo o conteúdo programado no Assistant.
- Para criar um bot no Telegram é necessário conversar com um bot chamado **BotFather** através do comando */newbot* Esse bot é o responsável por **criar** e **editar** bots no Telegram.



CRIAR BOT NO TELEGRAM (PASSOS)

1. Pesquisa por BotFather;
2. No chat com o Bot, usa-se o comando `/newbot` para criar um novo bot.
3. Insere um nome para o bot criado.
4. Insere um username com “bot” no final. Exemplo; MinicursoBot ou minicurso_bot.
5. Salve o Token que foi gerado.

BOT NO TELEGRAM

Após criado o bot no Telegram você receberá algumas informações sobre o bot criado, entre elas a informação mais importante de todas: O token.

Exemplo de token: 890839327:AAftDzqckqtIWJfTHI_t7sWiDjdKDIU9QRk

Esse token será usado para configurar o orquestrador de mensagens que será responsável pela integração entre o Watson Assistant e o Telegram.

IBM CLOUD


Com o bot criado no Telegram ,precisa-se de uma conta no IBM Cloud, para que seja possível a criação ambiente NodeRed para orquestrar as mensagens entre o Telegram e o Watson Assistant. A conta pode ser criada através do link abaixo:

<https://cloud.ibm.com/registration>




Após a criação da conta ,é necessário montar o ambiente NodeRed. Para isso pode ser usado o link abaixo:

<https://cloud.ibm.com/catalog/starters/node-red-starter>

AMBIENTE NODERED

 IBM Cloud

Procurar recursos e ofertas...

 Catálogo Documentos Suporte Gerenciar ▾ Alex Souza's Account  

Introdução

Visão geral

Tempo de execução

Conexões



Logs

API Management

Auto-scaling

Monitoramento

Lista de recursos /

 Minicurso-Watson-Assistant-Bot  Iniciando [Visite a URL do aplicativo](#)

Org: killersdalexsd02@gmail.com

Localização: Dallas

Espaço: dev

[Incluir tags](#)


Rotas ▾

Faça download, modifique e reimplente seu app Cloud Foundry com a interface da linha de comandos

Última atualização: 2019-08-13

Use a interface da linha de comandos (CLI) do IBM Cloud™ para fazer download, modificar e reimplentar seus aplicativos Cloud Foundry e instâncias de serviço.

Antes de iniciar, faça download e instale a [CLI](#).

 **Observe:** A CLI não é suportada pelo Cygwin. Use a ferramenta em uma janela de linha de comandos diferente da janela de linha de comandos do Cygwin.


1

Faça download do código do app em um novo diretório para configurar seu ambiente de desenvolvimento.

Atualize a versão do nó para `8.9.x` ou a versão mais recente no arquivo `package.json` antes de continuar. `{`
`note}{: hide-in-docs}`

Ativar o Windows
Acesse Configurações para ativar o Windows.

<

 IBM Cloud

Procurar recursos e ofertas...





Catálogo

Documentos

Suporte

Gerenciar ▾

Alex Souza's Account 



Introdução

Visão geral

Tempo de execução

Conexões


Logs

API Management

Auto-scaling

Monitoramento

Lista de recursos /

 Minicurso-Watson-Assistant-Bot  Esse app está ativo. [Visite a URL do aplicativo](#)


[Rotas ▾](#) 


Org: killersdalexsd02@gmail.com Localização: Dallas Espaço: dev [Incluir tags](#)

Faça download, modifique e reimplente seu app Cloud Foundry com a interface da linha de comandos

Última atualização: 2019-08-13

Use a interface da linha de comandos (CLI) do IBM Cloud™ para fazer download, modificar e reimplantar seus aplicativos Cloud Foundry e instâncias de serviço.

Antes de iniciar, faça download e instale a [CLI](#) .

 **Observe:** A CLI não é suportada pelo Cygwin. Use a ferramenta em uma janela de linha de comandos diferente da janela de linha de comandos do Cygwin.

① Faça download do código do app em um novo diretório para configurar seu ambiente de desenvolvimento.

Atualize a versão do nó para `8.9.x` ou a versão mais recente no arquivo `package.json` antes de continuar. `{`
`note}[: hide-in-docs]`

Ativar o Windows
Acesse Configurações para ativar o Windows.

AMBIENTE NODERED

- Ao criar o ambiente ,está sendo criado um servidor NodeJS.
- Ao clicar na url do aplicativo ,você será redirecionado para o ambiente NodeRed, onde primeiramente terá que criar um usuário juntamente com uma senha. **Não esqueça esses dados.**
- Ao criar esse usuário e senha você estará na página do NodeRed onde será possível acessar os flows que irão conectar o watson ao telegram

Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at nodered.org.



[Go to your Node-RED flow editor](#)

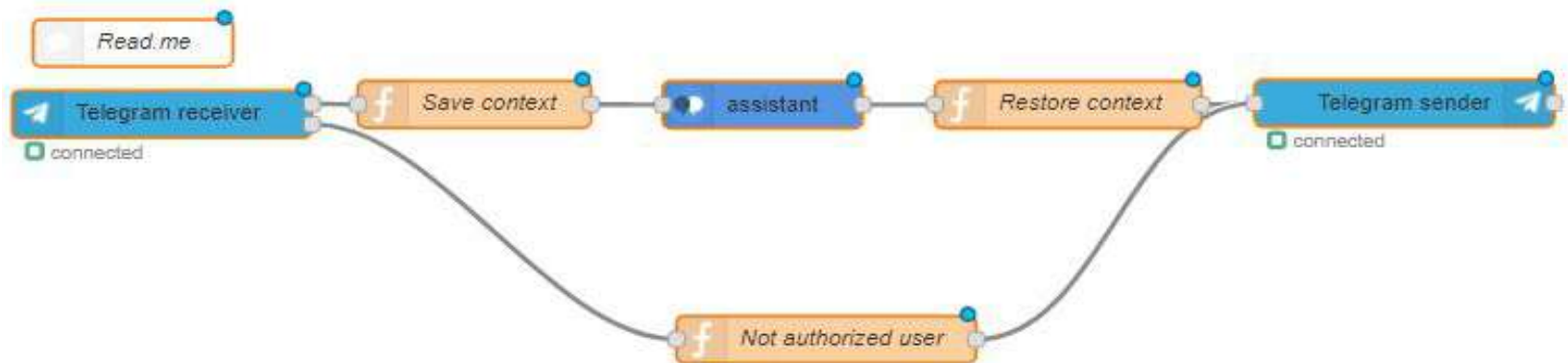
[Learn how to customise Node-RED](#)

Ativar o Windows
Acesse Configurações para ativar o Windows.

FLOWS

- São os fluxos que serão responsáveis por receber as mensagens que vem do telegram, converter para um formato que o watson entenda e pegue a resposta que está na base de conhecimento do watson e envie de volta ao telegram em um formato que o telegram entenda.
- Os flows nada mais são do que nós interligados que formam um fluxo para o processo descrito acima.

FLOWS



FLOWS

- Ao realizar o login, você chegará na tela onde o flow será montado.
- Primeiramente devemos adicionar dependências que são responsáveis por trazer os nós para serem usados para estabelecer uma comunicação com o telegram. Para isso iremos seguir o s seguinte passos.
- Acessa o Manage Pallet e pesquisa por: **node-red-contrib-telegrambot** e efetua a instalação
- Após isso precisamos buscar o flow de comunicação para a integração através do link abaixo:
 - <https://flows.nodered.org/flow/d8e99143e06720396aa2730d8015e4c0>

Telegram ChatBot using IBM Watson Conversation

This flow uses TelegramBot node and IBM Watson Conversation

```
{
  "id": "2222bafb.872c86",
  "type": "telegram",
  "receiver": {
    "z": "a79396ff.3eada",
    "name": "",
    "bot": "5485c65d.c921c8",
    "saveDataDir": "",
    "x": 109.375,
    "y": 420.0000305175781,
    "wires": [
      [
        [
          "3e0a95af.e5f08a"
        ],
        [
          "10222f18.29e139"
        ]
      ]
    ]
  },
  "id": "3e0a95af.e5f08a",
  "type": "function",
  "z": "a79396ff.3eada",
  "name": "Save context",
  "func": "context.flow.chatId = msg.payload.chatId;\ncontext.flow.type = msg.payload.type;\nmsg.payload = msg.payload.content;\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "x": 296.875,
  "y": 413.7500305175781,
  "wires": [
    [
      [
        "24ebd36b.04d1e4"
      ]
    ]
  ],
  "id": "24ebd36b.04d1e4",
  "type": "watson-conversation-v1",
  "z": "a79396ff.3eada",
  "name": "",
  "workspaceid": "e61968ba-db33-4b79-8597-f04f0d4177f8",
  "multiuser": false,
  "context": true,
  "empty-payload": false,
  "default-endpoint": true,
  "service-endpoint": "https://gateway.watsonplatform.net/conversation/api",
  "x": 470,
  "y": 414.0000305175781,
  "wires": [
    [
      [
        "3c133c0c.807744"
      ]
    ]
  ],
  "id": "3c133c0c.807744",
  "type": "function",
  "z": "a79396ff.3eada",
  "name": "Restore context",
  "func": "msg.payload.chatId = context.flow.chatId;\nmsg.payload.type = context.flow.type;\nmsg.payload.content = msg.payload.output.text[0];\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "x": 653.375,
  "y": 413.9375305175781,
  "wires": [
    [
      [
        "bf4a1e56.e282c"
      ]
    ]
  ],
}
```



sergiogama

Flow Info

created 1 year, 8 months ago

[view on github](#)

Node Types

Core

- comment (x1)
- function (x3)

Other

- telegram bot (x1)
- telegram receiver (x1)
- telegram sender (x1)
- watson-conversation-v1 (x1)

Tags

telegram

watson

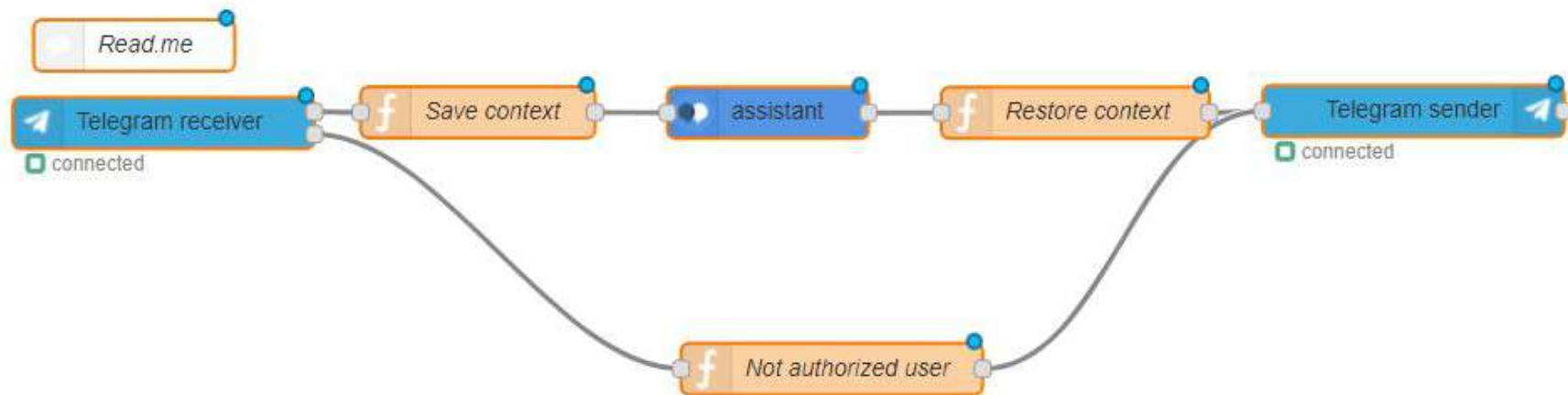
Windows

conversation

IBM

[Acesse Configurações pa](#)

FLOW DE INTEGRAÇÃO



NÓS DO *FLOW*

- Telegramreceiver
 - Carrega as mensagens que estão vindo do Telegram.
- Savecontext
 - Guarda variáveis de contexto na memória e joga o content que é a mensagem que veio do telegram para dentro do objeto msg.payload ,pois o Watson está esperando um msg.payload.
- Conversation
 - É o bot criado no Watson Assistant

NÓS DO *FLOW*

- Restorecontext
 - Retorna as informações que foram guardadas nas variáveis de contexto e pega a resposta do watson e manda para o content.
- Sender
 - Envia a resposta ao telegram.

Node-RED interface showing a flow named "Flow 1" and the "Edit function node" panel.

Flow 1:

- Starts with a "Telegram receiver" node.
- Followed by a "Save context" function node.
- Then an "assistant" node.
- Then a "Restore context" function node.
- A "Not authorized user" function node branches off from the "Save context" node and rejoins the flow after the "Restore context" node.
- The flow ends with a "Read me" node.

Edit function node:

Properties:

- Name: Save context

Function:

```
1 context.flow.chatId = msg.payload.chatId;  
2 context.flow.type = msg.payload.type;  
3 msg.payload = msg.payload.content;  
4 return msg;
```

Outputs: 1

Node-RED interface showing a flow named "Flow 1" and the "Edit function node" panel.

Flow 1:

- Starts with a "Read me" node.
- Followed by a "Telegram receiver" node.
- Then a function node "Save context".
- Then an "assistant" node.
- Then a function node "Restore context".
- Then a "Not authorized user" node.
- Finally, a "Telegram sender" node.

Edit function node:

Properties:

- Name: Restore context

Function:

```
1 msg.payload.chatId = context.flow.chatId;
2 msg.payload.type = context.flow.type;
3 msg.payload.content = msg.payload.output.text[0];
4 return msg;
```

Outputs: 1

CONFIGURAÇÃO DOS NÓS

- No nó Telegram receiver e no Telegram sender temos fazer algumas configurações para que o flow consiga se conectar ao telegram.
- É aqui que será preciso usar o token do bot criado no telegram.
- Primeiro coloca-se o nome do bot criado e em seguida o token gerado.
- Feito isso o bot já está conectado ao ambiente criado para a orquestração.

CONFIGURANDO O NÓ ASSISTANT

- Este nó é o responsável em acessar o bot criado no Watson Assistant, ele é quem vai receber as mensagens do telegram que vieram do Save context, e irá fazer as inferências para responder ao telegram enviando a resposta para o Restore context.
- Para isso será necessário o uso de credenciais do Assistant como o Workspace ID e a API Key.
- Ao colocar essas credenciais o bot está pronto para uso.

CONFIGURANDO O NÓ ASSISTANT

- É necessário abrir o Watson assistant onde são mostradas todas as skills criadas.
- Verifica os detalhes da skills clicando nos pontinhos da skill desejada.
- Copia o API Key e o Workspace ID para o nó Assistant.
- Faz o Deploy para colocar o flow no ar.
- Pronto!

Node-RED interface showing a flow named "Flow 1" and the "Edit assistant node" configuration panel.

Flow 1:

- Nodes: Save context, assistant, Restore context, Telegram sender, Not authorized user.
- Connections: Save context connects to assistant. assistant connects to Restore context. Restore context connects to Telegram sender. Not authorized user connects to Telegram sender.

Edit assistant node configuration:

- Buttons: Delete, Cancel, Done.
- Section: Properties
- Fields:
 - Name: Name
 - Username: Username
 - Password: Password
 - API Key:
 - Use Default Service Endpoint: ☒
 - Workspace ID: 63999b51-79c0-43bd-8e88-10ff934d986f
 - Timeout Period: Leave empty to disable
 - Save context: ☒
 - Multiple Users: ☐

At the bottom right, there is a watermark text: "Ative o Windows Acesse Configurações para ativar o Windows."

Referências

<https://www.ibm.com/developerworks/br/library/desenvolvendo-chatbots-com-watson-conversation/index.html>

<https://cloud.ibm.com/docs/services/assistant?topic=assistant-getting-started>