

**UNIVERSIDADE CATÓLICA DE SANTOS**  
**CIÊNCIA DA COMPUTAÇÃO**

**CHATBOT - ATENDENTE DE RESTAURANTE**

Hiero Bartholo de Carvalho

Santos – SP

2018

Hiero Bartholo de Carvalho

## **CHATBOT - ATENDENTE DE RESTAURANTE**

Pré-projeto de Trabalho de Conclusão de Curso apresentado na faculdade de Ciência da Computação na Universidade Católica de Santos como requisito para a conclusão do curso de Ciência da Computação.

Orientador: Marcio Luiz Piva

---

Santos – SP

2018

## ABSTRACT

As etapas de um fluxo de atendimento tradicional a clientes por *call center* ou *message center* geralmente apresentam atividades complexas e/ou de alto custo, enquanto que os *chatbots* vêm cada vez mais tomando conta do mercado de atendimento para realizar atendimentos eficientes a um custo reduzido. Neste projeto de pesquisa será estudada a criação de um *chatbot* personalizado para realizar atendimento no lugar de um atendente humano, que originalmente traria custos à empresa que fornece os serviços, via Facebook Messenger.

The steps of either a traditional callcenter service or a messaging customer service usually includes complex and/or high-cost activities. Meanwhile, chatbot have been gradually taking over the marketplace due to its low cost, efficiency and high availability. In this research project the development of a personalized chatbot for answering customers via Facebook Messenger instead of the need for a costly human attendant will be studied and implemented.

## LISTA DE FIGURAS

Figura 1: Arquitetura do Projeto.....	15
---------------------------------------	----

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>6</b>
1.1. CONTEXTUALIZAÇÃO .....	6
1.2. PROBLEMA .....	7
1.3. PROBLEMATIZAÇÃO .....	8
1.4. JUSTIFICATIVA .....	9
1.5. OBJETIVOS .....	10
1.5.1. GERAIS .....	10
1.5.2. ESPECÍFICOS .....	10
<b>2. METODOLOGIA.....</b>	<b>11</b>
<b>3. REVISÃO DE LITERATURA.....</b>	<b>13</b>
<b>4. PROJETO.....</b>	<b>15</b>
4.1. ARQUITETURA .....	15
4.2. WORKSPACE .....	17
4.2.1. INTENÇÕES .....	17
4.2.2. ENTIDADES .....	18
4.2.3. DIÁLOGO .....	20
4.3. O MIDDLEWARE.....	22
4.4. TREINAMENTO E INCREMENTO.....	24
4.4.1. Curadoria.....	24
4.4.2. Método 80/20.....	25
4.5. CRONOGRAMA .....	27
4.6. REFERÊNCIAS BIBLIOGRÁFICAS .....	29

# 1. INTRODUÇÃO

## 1.1. CONTEXTUALIZAÇÃO

De acordo com o artigo redigido por Guilherme Felliti e publicado em Maio de 2014 na revista Época Negócios, há muito tempo as máquinas ultrapassaram nossa eficiência em trabalhos pesados ou repetitivos. Agora elas avançam sobre o terreno das tarefas complexas – e ameaçam quase metade dos empregos que existem hoje. Isso significa que o mundo está cada vez mais inteligente. As tarefas mais complexas que Guilherme aborda são as que exigem uma certa inteligência, como categorizar imagens ou sugerir produtos. Não é, portanto, impensável que as máquinas tenham a capacidade de conversar diretamente com humanos, utilizando linguagem natural.

Larissa Lotufo, jornalista digital, publicou em julho de 2017 no site ecommercebrasil.com.br um artigo sobre a ascensão dos *ChatBots*<sup>[1]</sup>. “Ainda temos receio das inovações tecnológicas que surgem, é o caso dos Chatbots, uma opção no atendimento ao cliente que para alguns é tendência e para outros impossível de dar certo.”.

Sendo utilizados com frequência para tirar dúvidas, instruir passo a passo usuários de determinado serviço, e realizar atendimentos, os *ChatBots* têm se popularizado. Um bom exemplo disso, é o Banco Next, que conta com um atendente virtual que ajuda seus clientes a realizar operações bancárias. No final de 2016, em parceria com o Governo de São Paulo, a empresa Nama desenvolveu um *ChatBot* de atendimento especial para realizar agendamentos para as agências do Poupatempo. O *ChatBot*, apelidado de “Poupinha”, veio com o objetivo de diminuir o índice de pessoas que chegam às agências sem o agendamento realizado. Os dados publicados no site do governo do estado de São Paulo revelaram que em seus primeiros 33 dias, Poupinha realizou mais de 53 mil agendamentos e trocou mais de 2,4 milhões de mensagens com usuários, e teve seu objetivo alcançado.

<sup>[1]</sup>*Chatbot*. Termo utilizado para representar um sistema de inteligência artificial que tem a capacidade de conversar com um ser humano utilizando processamento de linguagem natural.

## 1.2. PROBLEMA

Em um restaurante qualquer, o treinamento de atendentes humanos é uma situação que gera custo para o restaurante, e caso um desses profissionais saia da empresa, seja qual for o motivo, há necessidade de treinar uma nova pessoa para exercer esse papel. Entre os problemas citados no ebook “Já pensou em melhorar o atendimento a clientes usando chatbots?” disponibilizado gratuitamente na web pela Via Cognitiva, esses são apenas alguns envolvendo atendimento humano em estabelecimentos comerciais que poderiam ser contornados utilizando *ChatBots*. Com um atendente virtual, essa necessidade de “re-treino” de um profissional deixa de existir, bem como a saída dos atendentes, que estarão sempre disponíveis

De qualquer forma, a criação de um *ChatBot* exige um estudo delicado, pois necessita de um forte embasamento de exemplos, ou seja, a inteligência artificial de um *ChatBot* se dá ao reconhecimento de linguagem natural, mas antes esse conhecimento deve ser passado de um humano ao *ChatBot*, conhecimento esse que precisa ser bem definido, porque se não o projeto corre o risco de sair do escopo.

### 1.3. PROBLEMATIZAÇÃO

Dar uma personalidade e conhecimento a uma máquina que conversa pode ser desafiador.

- Será que um *ChatBot* conseguiria realizar todo o atendimento que um ser humano está apto a realizar?
- Será que um *ChatBot* poderia agir de forma profissional, mas com um tom ligeiramente informal?
- Será que utilizar *Processamento de Linguagem Natural* como técnica de Inteligência Artificial é a melhor maneira de se realizar uma conversa entre um humano e uma máquina?



## 1.4. JUSTIFICATIVA

O estudo de tecnologias inteligentes tem estado em alta desde o século passado, alavancado por Alan Turing, que em seu livro “*Computing Machinery and Intelligence*” (1950) levantou a seguinte pergunta: “As máquinas podem pensar?”. Desde então, os estudos acerca da inteligência artificial tem sido propagados. Grandes empresas, desde as mais consagradas até as mais jovens (IBM, Microsoft, Amazon), têm investido em pesquisas no ramo da inteligência artificial.

Inicialmente, o estudo da inteligência artificial era meramente científico, mas a aplicabilidade desse tipo de tecnologia no mercado mostrou-se bem valorizado, no atendimento de clientes, classificação automática de padrões, reconhecimento facial e até mesmo recomendações de produtos.

Um *ChatBot* atendente de restaurante pode otimizar a quantidade de pedidos que esse restaurante pode receber simultaneamente. Assim, um restaurante pode crescer aumentando sua equipe de cozinheiros, mas sem a necessidade de aumentar a equipe de atendentes telefônicos.

Mas por que isso seria importante no viés da Ciência da Computação? A lei de Moore, criada por Gordon Earl Moore em 1965 diz que o poder de processamento dos computadores dobram a cada 18 meses. Além disso, de acordo com Erik Brynjolfsson em seu livro “*Rage Against The Machine – 2011*” o poder de software cresce pelo menos na mesma velocidade que o poder de hardware. Conclui-se que o poder de software dobra a cada 18 meses ou menos. É quase como se software e hardware estivessem em uma eterna corrida, e portanto, o estudo de inteligência artificial que envolve algoritmos de complexidade elevada, colabora com o lado do software nessa corrida.

## 1.5. OBJETIVOS

### 1.5.1. GERAIS

Neste projeto, será abordada a visão de que um *ChatBot* pode substituir o atendimento de um ser humano em um restaurante especializado em *pizzas* e *esfihas*.

### 1.5.2. ESPECÍFICOS

Desenvolver um *ChatBot* e um sistema *Middleware*<sup>[2]</sup> para provar que um *ChatBot* tem a capacidade de interagir com clientes de forma natural, tentando minimizar a robotização da conversa, de forma que o cliente sinta-se confortável ao conversar com o *ChatBot*.

Esse *ChatBot* será responsável por automatizar o atendimento de um restaurante especializado em *Pizzas* e *Esfihas*

Para isso, o *ChatBot* receberá uma persona chamada “Aria”, sigla para “Atendente de Restaurante Inteligência Artificial”. A Personalidade de Aria será profissional, com um tom levemente informal, para que os clientes sintam-se à vontade para conversar.

O usuário irá conversar com o *ChatBot* através do *Facebook Messenger*. O usuário poderá tirar dúvidas gerais sobre o restaurante, como localização, datas e horários de funcionamento. O usuário poderá solicitar reserva de lugares, informando quantidade de pessoas, data e horário. O usuário poderá realizar pedidos do cardápio e solicitar entrega em sua residência.

Ao final do atendimento, o *ChatBot* solicitará ao usuário atendido um breve *feedback*<sup>[3]</sup> quanto ao atendimento, para que seja possível provar a hipótese.

<sup>[2]</sup>*Middleware*: Sistema intermediário que realiza a intermediação da comunicação entre dois outros sistemas, para que estes “conversem” entre si.

<sup>[3]</sup>*Feedback*: De acordo com o dicionário Dicio, feedback é o processo em que a mensagem emitida obtém uma reação de quem a recebe, sendo usada para avaliar os efeitos desse processo, podendo ser uma reação positiva ou negativa.

## 2. METODOLOGIA

Em primeiro lugar, serão revisados livros que abordam socialmente o conceito de processamento de linguagem natural como técnica de inteligência artificial, atentando-se às consequências que um atendente virtual pode trazer.

O *framework*<sup>[4]</sup> utilizado para criar este *ChatBot* será o *IBM Watson Assistant*, que é gratuito para um número de chamadas de API mensais limitado a 10.000, integrado ao *Facebook Messenger* através de um *middleware* escrito em *NodeJS*<sup>[5]</sup>.

O *Middleware* utilizará os pacotes de API do IBM Watson e Facebook e será hospedado em um servidor da *IBM Cloud*, com recursos alocados de forma a não exceder os limites do plano gratuito (1 Gigabyte de memória RAM).

Para que se possa treinar um *ChatBot* de forma imparcial, a melhor metodologia de coleção de exemplos para treino, é uma pesquisa de campo. Essa pesquisa será realizada através do *Google Forms*<sup>[6]</sup>, onde serão solicitados entre 5 e 7 exemplos de perguntas.

Após a coleta dos exemplos de perguntas, esses exemplos serão divididos por intenção\* do usuário. Por exemplo: os exemplos “Quero fazer um pedido” e “Gostaria de pedir uma pizza” podem fazer parte de uma intenção de usuário denominada “FazerPedido”

As mensagens para sistemas externos serão enviadas a partir do *middleware*. Essa mensagem será enviada em formato *JSON*<sup>[7]</sup>.

<sup>[4]</sup>*Framework*: abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. (Wikipedia)

<sup>[5]</sup>*NodeJS*: *Framework* criado por Ryan Dahl que possibilita executar a linguagem de programação EcmaScript sem a necessidade de um navegador web.

<sup>[6]</sup>*Google Forms*: Serviço para criar questionários e prover dados estatísticos sobre as respostas.

<sup>[7]</sup>*JSON*: JavaScript Object Notation. É uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro – 1999

A logística na reserva de horários no restaurante e a logística de entrega dos pedidos não será tratada neste trabalho, pois essas funcionalidades exigem sistemas externos que fogem do escopo e intuito do *ChatBot*. Ainda sim, o *ChatBot* terá como enviar uma mensagem para um sistema externo com as informações que tais sistemas precisariam para desempenhar suas funções através do *middleware*.

Lembrando que esse atendimento não será feito por telefone, e não substitui o atendimento telefônico, mas otimiza o tempo livre de um atendente telefônico, pois enquanto um atendente humano atende um cliente de cada vez, um *ChatBot* pode atender, por exemplo, mais de 5 ao mesmo tempo.

### 3. REVISÃO DE LITERATURA

Ao decidir criar um *chatbot* para interagir com a plataforma *Facebook Messenger*, torna-se necessário o estudo e aprendizado de como fazer a estruturação do *chatbot* e como realizar a comunicação com a plataforma de interface (no caso, o próprio *Facebook Messenger*).

Após algumas pesquisas na *web*, alguns artigos públicos sobre como integrar *chatbots* ao *Facebook Messenger* vieram à tona (vide seção 6. REFERÊNCIAS BIBLIOGRÁFICAS). Esses artigos mostram como criar as rotas de comunicação necessárias para que seja feita uma interação com o *Facebook Messenger*. Todos os artigos apontam o uso das bibliotecas ‘*express*’, ‘*request*’ e ‘*body-parser*’.

As definições dessas bibliotecas, segundo os autores das mesmas são:

*Express: Fast, unopinionated, minimalist web framework for Node* (Em português, *Framework web rápido, discreto e minimalista para Node*).

*Request: Simplified HTTP Client* (Em português, *Cliente HTTP Simplificado*).

*Body-parser: Node.js body parsing middleware* (Em português: *middleware para conversão do corpo de mensagens para Node.js*).

Para o estudo de como criar um *chatbot*, foi consultada a referência oficial do *IBM Watson Assistant*, disponibilizada na *IBM Cloud*.

As seções consultadas para o aprendizado foram:

Nós de diálogo;

Intenções;

Entidades;

*Slots*;

Variáveis de Contexto;

Além da referência para criação do *chatbot*, foi consultada a referência do ‘*watson-developer-cloud*’, que também é uma biblioteca disponibilizada gratuitamente pela *IBM*

para comunicação com os serviços do *Watson* hospedados na *IBM Cloud*. Essa biblioteca possui diversos componentes de comunicação com os serviços do *Watson*, mas será utilizado apenas o componente referente ao *Watson Assistant*.

## 4. PROJETO

### 4.1 ARQUITETURA

Figura 1 - Arquitetura do Projeto



O *Facebook Messenger* estará conectado diretamente ao *middleware*, acionando um *webhook*<sup>[8]</sup> quando uma nova conversa for iniciada por um usuário.

Quando o *webhook* é ativado, o *middleware* abre uma instância de comunicação com o *Watson Assistant* para o usuário que iniciou a conversa e a sessão só é encerrada quando a janela do *Messenger* é fechada pelo usuário.

<sup>[8]</sup>*Webhook*: Maneira prática para um app ou sistema fornecer outras aplicações com informações em tempo real. O *webhook* fornece dados para outros aplicativos de forma passiva através de comunicação realizada pelo protocolo *Http*.

<sup>[9]</sup>*npmjs.com*: É o site responsável por apresentar os pacotes de *NodeJS* disponíveis. *NPM* significa 'Node Package Manager', em português, Gerenciador de Pacotes do *Node*.

Para a conexão do *middleware* com o Assistant será usada a biblioteca 'watson-developer-cloud', disponível gratuitamente no site *npmjs.com*<sup>[9]</sup>. A conexão é feita por uma API Rest<sup>[10]</sup>.

Para a conexão do *middleware* com o Facebook Messenger será usada a Rest API do próprio facebook, configurando os tokens<sup>[11]</sup> e webhooks necessários manualmente por rotas dentro do *middleware*

<sup>[10]</sup>*Rest API*: É um modelo de interface de aplicação web para a criação de web services que utiliza o protocolo Http de forma a reduzir a largura de banda utilizada na comunicação entre duas aplicações (um cliente e um servidor).

<sup>[11]</sup>*Token*: Um token de acesso é uma cadeia de caracteres opaca que identifica um usuário, aplicativo ou Página. O token inclui informações sobre quando o token expirará e qual aplicativo gerou o token. Devido às verificações de privacidade, a maioria das chamadas de API no Facebook precisa incluir um token de acesso.



## 4.2 WORKSPACE

*Workspace* é a configuração em formato JSON do Watson e todas as informações de Intenções, Entidades e Diálogo presentes em uma *Skill*. Cada *Skill* é um conjunto de Intenções, Entidades e nós de Diálogo que pode ser alterada manualmente pela plataforma do Watson Assistant dentro da IBM Cloud.

A *Skill* foi inicialmente configurada com algumas intenções, entidades e nós de diálogo básicos que serão discutidos mais a frente.

### 4.2.1. INTENÇÕES

Uma intenção é definida como “O que um usuário quer fazer”. Cada intenção carrega um conjunto de exemplos de sentenças que a definem. Por exemplo, uma intenção que contém exemplos de frases como “bom dia”, “boa noite”, “Olá” e “Como vai?” pode receber o nome de “cumprimento”. A representação de uma intenção é um símbolo de cerquilha (#) seguido pelo nome da intenção, nome este que deve conter apenas caracteres alfanuméricos.

Boas práticas ao registrar exemplos em intenções são:

- I. Manter a quantidade de exemplos equilibrada entre todas as intenções na *Skill*. O motivo disso é para que uma intenção não fique muito mais “pesada” que outras, pois pode fazer com que uma pergunta enviada pelo usuário seja identificada incorretamente como uma intenção “pesada”. A qualidade e diversidade dos exemplos é mais importante que a quantidade.
- II. Evitar o uso de caracteres maiúsculos ao iniciar frases de exemplo. O motivo por trás dessa boa prática é que o Watson entende letras maiúsculas como nomes próprios, portanto se um exemplo está iniciando com letra maiúscula, ao enviar uma pergunta para o Watson com a mesma frase, mas iniciando em letra minúscula, a confiança retornada não será de 100%. No entanto,

perguntar algo iniciando-se com letra maiúscula quando há um exemplo catalogado apenas com letras minúsculas, o Watson retornará 100% de confiança.

A seguir, a lista de intenções da *Skill* “Atendente Restaurante”, com base na configuração inicial:

- agradecimento: 4 exemplos
- cumprimento: 6 exemplos
- despedida: 5 exemplos
- perguntarDisponibilidade: 3 exemplos
- solicitacao: 10 exemplos

As intenções e exemplos mencionados acima não representam o projeto finalizado. Haverão novas intenções e exemplos conforme a necessidade verificada pela análise de conversas durante a versão beta<sup>[12]</sup>.

#### 4.2.2. ENTIDADES

Uma entidade é uma palavra ou um termo que serve para filtrar informações contidas em uma pergunta. Para exemplificar, vamos utilizar a intenção “solicitacao”: Quando um usuário quer solicitar algo, ele envia sua mensagem ao bot, que reconhece a intenção “solicitacao”, mas a intenção não define o que o usuário quer solicitar. Para isso é utilizado o conceito de entidades. No contexto do atendente, uma solicitação pode ser um pedido ou uma reserva, que são as entidades utilizadas para distinguir o que o usuário quer fazer.

<sup>[12]</sup> *Versão Beta*: De acordo com um artigo publicado no site canaltech.com, a versão beta de um software ou produto é a versão em estágio ainda de desenvolvimento, mas que é considerada aceitável para ser lançada para o público, mesmo que ainda possua bugs e problemas que precisarão ser reparados pelos desenvolvedores antes do lançamento definitivo do produto ao mercado na sua versão final.

A representação de uma entidade é um símbolo de arroba (@) seguido pelo nome da entidade, nome este que deve conter apenas caracteres alfanuméricos ou espaços.

Além das entidades que são possíveis de criar, existem entidades de sistema, configuradas com padrões a facilitar a utilização de alguns contextos. As entidades de sistema são:

- sys-date: identifica datas e intervalos de data;
- sys-number: identifica números;
- sys-currency: identifica valores em moeda corrente;
- sys-time: identifica horas, minutos e segundos;
- sys-percentage: identifica porcentagens
- sys-location: identifica países e cidades (atualmente indisponível no idioma português);
- sys-person: identifica nomes próprios (atualmente indisponível no idioma português).

As boas práticas ao definir entidades envolvem pensar com cautela no contexto em que serão utilizadas. Cada entidade pode possuir 0 ou mais valores, e cada um desses valores pode ter 1 ou mais sinônimos, contando o próprio valor sempre como um sinônimo.

A seguir, a lista de intenções da *Skill* “Atendente Restaurante”, com base na configuração inicial:

- sys-date, para identificar datas de reserva;
- sys-number, para identificar quantidade de pessoas na reserva;
- sys-time, para identificar horários da reserva;
- Acao: 1 valor;
- Posicionamento: 2 valores;
- Produto: 3 valores;
- Sabor: 46 valores;

- Solicitacao: 2 valores;

As entidades e valores mencionados acima não representam o projeto finalizado. Haverão novas entidades e valores conforme a necessidade verificada pela análise de conversas durante a versão beta.

### 4.2.3. DIÁLOGO

O diálogo é o terceiro componente de uma skill. Um diálogo contém diversos nós. Cada nó representa uma condição e um tratamento. As condições geralmente envolvem comparações de intenções e entidades. Um nó de diálogo pode conter de 0 a N nós filhos, que pode ser avaliado e executado caso o nó pai seja validado. Os tratamentos de nós envolvem respostas, gravação de valores em variáveis.

Os nós podem ser agrupados em pastas, sem influenciar o fluxo de execução dos nós que estão na raiz daquela ou outra pasta.

Como os nós são avaliados sequencialmente na ordem em que estão dispostos na interface gráfica do ambiente de configuração do IBM Watson Assistant, será imprescindível que nós que avaliam múltiplas condições venham na frente dos nós que avaliam apenas uma.

Os nós serão divididos em cinco grupos (pastas): Miscelânea (em que ficarão os nós que identificam intenções simples, tais como Cumprimento e Agradecimento), Entender Solicitação (que vai conter o fluxo responsável por identificar que o usuário deseja fazer uma solicitação, e descobrir se o usuário quer solicitar uma reserva ou se quer realizar um pedido), Fluxo de Reserva (que conterá todos os nós responsáveis pela reserva), Fluxo de Pedido (contendo os nós que ajudam o usuário a realizar seu pedido) e Duvidas Restaurante (onde estarão os nós que identificam qual a dúvida que o cliente tem e responde).

Caso nenhuma condição seja satisfeita, Aria responderá a seguinte mensagem: “Me desculpe, ainda estou em desenvolvimento e não consigo te responder ainda. Você pode tentar perguntar de outra maneira.”.

O ambiente de configuração do workspace do Watson Assistant conta com uma janela de testes, onde serão submetidas as perguntas em fase de desenvolvimento para checar se o diálogo está fluindo conforme o esperado pela configuração.

### 4.3. O MIDDLEWARE

O *middleware*, conforme dito, será escrito em JavaScript utilizando o framework NodeJS.

Os pacotes utilizados para realizar a implementação do *middleware* serão:

- express
- body-parser
- watson-developer-cloud
- request

O pacote ‘express’ fornece métodos simples para a criação de endpoints<sup>[13]</sup> e utiliza como base a biblioteca-padrão ‘http’ do NodeJS. O ‘express’ é um dos pacotes mais bem avaliados da comunidade para a utilização com Rest API’s.

‘body-parser’ é um pacote que permite a circulação (envio e recebimento) de dados em formato JSON pelas rotas configuradas com ‘express’.

O ‘watson-developer-cloud’ é o pacote que a IBM fornece para a interação simplificada com todos os serviços de Watson oferecidos na plataforma IBM Cloud. Dentro desse pacote, será utilizado o componente ‘ConversationV1’ para realizar a comunicação com o IBM Watson Assistant configurado na IBM Cloud.

<sup>[13]</sup>*Endpoint*. O site devbrasil.com explica endpoint como “o que um serviço expõe”, podendo ser uma rota de comunicação Http. Um endpoint contém três principais características que chamamos de ABC (Address, Binding e Contract). Uma analogia para estas características seria que o Address é onde o serviço está hospedado, Binding é como o serviço pode ser acessado, e Contract seria o que pode ser visto no serviço.

Por fim, o pacote 'request', que também é baseado na biblioteca 'http' do NodeJS, será utilizado para realizar requisições do tipo POST<sup>[14]</sup> a um endpoint fantasma, simulando o servidor do restaurante, que irá apenas receber as informações de pedido e reserva no formato JSON, mas não tratará nenhuma dessas informações.

Em resumo, os pacotes 'express' e 'body-parser' tornarão possível a conexão do *middleware* com o web service do facebook messenger. Serão disponibilizadas duas rotas para que o facebook messenger realize a comunicação com o *middleware*. Ambas as rotas terão o endereço '/webhook' sendo uma GET<sup>[15]</sup> e uma POST. A rota GET servirá para realizar a autenticação do facebook messenger, e a rota POST será por onde a informação das mensagens enviadas pelo usuário ao facebook messenger serão repassadas ao *middleware*, que comunicar-se-á com o Watson Assistant através de uma requisição utilizando o 'watson-developer-cloud', e caso a chamada do Watson retorne a informação de que foi feita uma reserva ou um pedido o *middleware* vai realizar o envio dessa informação ao endpoint fantasma utilizando o pacote 'request'.

<sup>[14]</sup>POST: É um método de comunicação de dados por meio do protocolo Http em que a informação a ser enviada é passada no corpo da mensagem, separadamente da URL da requisição. O fato da informação ser passada no corpo da mensagem não significa que este método de comunicação seja mais ou menos seguro que uma comunicação realizada por meio do método GET.

<sup>[15]</sup>GET: É um método de comunicação de dados por meio do protocolo Http em que a informação a ser enviada é passada junto com o endereço, na URL da requisição.

## **4.4. TREINAMENTO E INCREMENTO**

O treinamento do bot será realizado inicialmente com a configuração inicial apresentada na seção 4.

### **4.4.1. CURADORIA**

A curadoria é a função que um humano desempenha para melhorar um chatbot. Essa função envolve análises minuciosas do estado de um chatbot e exemplos de usuários, e decisões de alteração para que o chatbot desempenhe seu papel da melhor forma possível.

Neste projeto serão utilizadas três formas de curadoria:

- Google Forms
- Análise de conversas
- O método 80/20

Em Junho será criado e liberado um formulário no Google Forms para coletar exemplos de perguntas que podem ser feitas para Aria. Esses exemplos serão analisados e filtrados para que a base de conhecimento do bot fique consistente com a regra de negócio proposta.

Após a implementação do middleware e a conexão do Watson com o Facebook Messenger, será anunciado a fase de testes do bot para uso livre. Conforme usuários forem conversando com a Aria, as conversas ficarão registradas na página do facebook e poderão ser analisadas.

As análises mostrarão se a configuração do workspace precisa de melhorias, modificações ou correções de bug. Exemplos novos poderão ser verificados e adicionados caso não fujam do escopo proposto.

Os exemplos coletados no Google Forms, e as conversas salvas no Facebook Messenger servirão para alimentar o modelo do Watson Assistant e treiná-lo na medida do necessário.



#### 4.4.2. MÉTODO 80/20

Para checar se os níveis de confiança geral das intenções de uma skill estão boas, é utilizado o método de análise 80/20.

O método 80/20 é um experimento analítico qualitativo que consiste em separar aleatoriamente os exemplos contidos nas intenções de uma skill em cinco grupos, cada grupo com 20% dos exemplos presentes na skill.

Após separados os grupos, serão criadas 5 skills de teste (denominadas *folds*), onde cada skill será treinada com os exemplos contidos em 4 dos grupos gerados aleatoriamente.

Após o período de treino das skills, cada uma será bombardeada com perguntas referentes aos exemplos faltantes naquela skill.

Para exemplificar:

*Fold 1:*

Treinado com os grupos: 1, 2, 3 e 4

Testado com o grupo: 5

*Fold 2:*

Treinado com os grupos: 1, 2, 3 e 5

Testado com o grupo: 4

*Fold 3:*

Treinado com os grupos: 1, 2, 4 e 5

Testado com o grupo: 3

*Fold 4:*

Treinado com os grupos: 1, 3, 4 e 5

Testado com o grupo: 2

*Fold 5:*

Treinado com os grupos: 2, 3, 4 e 5

Testado com o grupo: 1

Cada teste produz um retorno de corretude contendo: A intenção que a que um exemplo pertence, a intenção em que esse exemplo foi reconhecida e a confiança do algoritmo ao reconhecer a intenção.

Após a checagem dos resultados desse teste, é gerado um resultado consolidado com as informações obtidas e pode ser realizada uma análise humana de curadoria para decidir quais as alterações (caso necessárias) devem ser realizadas.

## 5. CRONOGRAMA

O cronograma a seguir possui X, I e F, significando respectivamente que uma tarefa será realizada naquele mês, que uma tarefa se inicia naquele mês e que uma tarefa tem finalização prevista para aquele mês.

Atividade	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
Definição do tema	X										
Definição da proposta de estudo		X									
Início da pesquisa de referências da integração do <i>IBM Watson Assistant</i> com chat de <i>Facebook</i>			X								
Definição da arquitetura do <i>middleware</i>			X								
Criar a base de conhecimento inicial				X							
Definir as melhores entidades para um diálogo consistente				X							
Configuração inicial do diálogo				X							
Entrega da primeira parte do trabalho de pesquisa					X						

[illegible]

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ANWESHAN; GERMANATTANASIO; NFRIENDLY. Watson-developer-cloud. 2018. < <https://www.npmjs.com/package/watson-developer-cloud> >

Race Against the Machine. 2011.

BRYNJOLFSSON, Erik; MCAFEE, Andrew. The Second Machine Age. 2014.

DANTAS, Rodrigo. O que são webhooks?. 2017. < <https://blog.vindi.com.br/o-que-sao-webhooks/> >

DOUGWILSON. Body-parser. 2018. < <https://www.npmjs.com/package/body-parser> >

DOUGWILSON; HACKSPARROW; JASNELL; MIKEAL. Express. 2018. < <https://www.npmjs.com/package/express> >

DURÃES, Ramon. WCF – Entendendo o que é um endpoint. 2018.

< <http://devbrasil.net/group/wcf/page/wcf-entendendo-o-que-um-endpoint> >

ECHESSA, Joyce. Building a Facebook Messenger chat bot with Node and Heroku. 2017. < <https://www.sitepoint.com/building-facebook-chat-bot-node-heroku/> >

FACEBOOK FOR DEVELOPERS. Facebook for Developers. 2018.

< <https://developers.facebook.com> >

FACEBOOK FOR DEVELOPERS. Tokens de Acesso. 2018.

< [https://developers.facebook.com/docs/facebook-login/access-tokens/?locale=pt\\_BR](https://developers.facebook.com/docs/facebook-login/access-tokens/?locale=pt_BR) >

FELLITI, Guilherme. Adeus, humanos: O que os robôs podem fazer por você (ou no seu lugar). 2014.

< <https://epocanegocios.globo.com/Informacao/Visao/noticia/2014/05/adeus-humanos.html> >

FIGUEROA, Pilar. How to create your very own Facebook Messenger bot with Dialogflow and Node.js in just one day. 2017. < <https://medium.com/crowdbotics/how->

[to-create-your-very-own-facebook-messenger-bot-with-dialogflow-and-node-js-in-just-one-day-f5f2f5792be5](#) >

FORD, Martin. Rise of the Robots. 2015.

FRAMEWORK. Wikipedia. 2018 < <https://pt.wikipedia.org/wiki/Framework> >

FREDKSCHOTT; MIKEAL; NYLEN; SIMOV. Request. 2018. < <https://www.npmjs.com/package/request> >

GIRLIEMAC. Creating a simple Facebook messenger AI Bot with API.ai in Node.js. 2017. < <https://girliemac.com/blog/2017/01/06/facebook-apiai-bot-nodejs/> >

Já pensou em melhorar o atendimento a clientes usando chatbots?. ViaCognitiva (2017).

IBM. Getting Started with Watson Assistant. 2018.

< <https://console.bluemix.net/docs/services/conversation/getting-started.html> >

IBM. Watson Developer Cloud Documentation for Watson Assistant. 2018.

< <https://www.ibm.com/watson/developercloud/assistant/api/v1/node.html?node> >

JSON. JSON. 2018 < <https://www.json.org/json-pt.html> >

LOTUFO, Larissa. Por que ainda temos medo da tecnologia? Uma análise sobre Chatbots. 2017. < <https://www.ecommercebrasil.com.br/artigos/uma-analise-sobre-chatbots/> >

MICHELETTI, Flávio. Como funcionam os métodos GET e POST. 2016. < <http://www.devfuria.com.br/php/como-funcionam-os-metodos-get-e-post/> >

NPM. Welcome to NPM. 2018. < <https://www.npmjs.com> >

NPM. What is NPM?. 2018. < <https://docs.npmjs.com/getting-started/what-is-npm> >

PORTAL DO GOVERNO. Poupatempo tem robô virtual que facilita atendimento e agenda horários. 2017 < <http://www.saopaulo.sp.gov.br/spnoticias/poupatempo-tem-robo-virtual-que-facilita-atendimento-e-agenda-horarios/> >

REDAÇÃO CANAL TECH. O que significa dizer que um software ou produto está em versão beta? 2018. < <https://canaltech.com.br/produtos/O-que-significa-dizer-que-um-software-ou-produto-esta-em-versao-beta/> >

TURING, Alan. Computing Machinery and Intelligence. 1950