

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
COLÉGIO POLITÉCNICO DA UFSM  
CURSO DE SISTEMAS PARA INTERNET**

**UM ESTUDO SOBRE CHATBOTS E SUA  
APLICAÇÃO NO COMÉRCIO ELETRÔNICO**

**TRABALHO DE CONCLUSÃO DE CURSO II**

**Denis Bao Motta**

**Santa Maria, RS, Brasil**

**2018**

# **UM ESTUDO SOBRE CHATBOTS E SUA APLICAÇÃO NO COMÉRCIO ELETRÔNICO**

**Denis Bao Motta**

Trabalho apresentado como requisito parcial para a Conclusão do Curso de  
Sistemas para a Internet da Universidade Federal de Santa Maria.

**Orientador: Prof. Daniel Lichtnow**

**Santa Maria, RS, Brasil**

**2018**

Catálogo na fonte elaborada pela biblioteca da  
Universidade Federal de Santa Maria

A ficha catalográfica é confeccionada pela Biblioteca  
Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:

<http://www.bu.ufsc.br/design/Catalogacao.html>

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
COLÉGIO POLITÉCNICO DA UFSM  
CURSO DE SISTEMAS PARA INTERNET**

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Conclusão de  
Curso

**UM ESTUDO SOBRE CHATBOTS E SUA APLICAÇÃO NO  
COMÉRCIO ELETRÔNICO**

Elaborado por  
Denis Bao Motta

Como requisito parcial para obtenção do grau de  
**Tecnólogo em Sistemas para Internet**

**COMISSÃO EXAMINADORA:**

---

**Prof. Daniel Lichtnow, Dr.**  
(Presidente/Orientador)

---

**Prof. Marcos Alexandre Rose Silva, Dr.**  
(UFSM)

---

**Prof<sup>a</sup>. Vanessa Gindri Vieira, Dr<sup>a</sup>.**  
(UFSM)

Santa Maria, 11 de dezembro de 2018

Se é esperado que uma máquina seja infalível, não se pode esperar que ela também seja inteligente.

(Alan Turing, 1947)

## **AGRADECIMENTOS**

Agradeço a todos os professores do curso de Sistemas Para Internet que compartilharam seus conhecimentos em sala de aula e acompanharam a minha jornada enquanto universitário, especialmente ao meu orientador, Prof. Daniel Lichtnow, por todo suporte, amizade, dedicação e incentivos.

Agradeço também à minha mãe Ermeli Maria Bao, pelo constante apoio, assim como à minha companheira Anieli Golin por toda compreensão, amor e perseverança.

Por fim, agradeço aos amigos e colegas da empresa Moonbase Digital que me deram a oportunidade de aprender na prática todo o conhecimento que adquiri durante a minha graduação, assim como todo o suporte necessário nesse processo.

## RESUMO

### UM ESTUDO SOBRE CHATBOTS E SUA APLICAÇÃO NO COMÉRCIO ELETRÔNICO

AUTOR: Denis Bao Motta  
ORIENTADOR: Daniel Lichtnow

Este trabalho apresenta um estudo sobre os agentes de conversação chamados *chatbots*, abordando sua história e evolução desde seu surgimento, ao lado dos primeiros conceitos de Inteligência Artificial e das primeiras concepções imaginadas por Alan Turing de um diálogo entre humanos e máquinas, até a sua recente popularização com aplicações em diversas áreas como educação, saúde, entretenimento e, principalmente, no comércio eletrônico. Através da análise de algumas aplicações que se destacaram durante este processo evolutivo, como *Eliza*, *Jabberwacky*, *A.L.I.C.E*, *SmartChild*, e outros, é feito um estudo do funcionamento e aplicação de algumas das tecnologias que ao longo deste período agregaram novos potenciais a este tipo de ferramenta. O trabalho ainda faz uma abordagem sobre o surgimento do comércio conversacional e a utilização dos *chatbots* em aplicativos de troca de mensagens por parte de empresas que buscam atingir esse crescente mercado consumidor. É também exemplificada a criação de um destes agentes de conversação através do sistema IBM *Watson* e sua API *Conversation*, uma de muitas plataformas para criação de *chatbots* que vêm se popularizando conforme cresce o interesse das empresas em fazer uso desta solução digital. Por fim, é descrito o desenvolvimento, ferramentas e tecnologias utilizadas para a construção de um *chatbot* destinado ao comércio eletrônico e integrado a rede social *Facebook* através da plataforma *Messenger* e do sistema *Wit.ai*, capaz de exemplificar conceitos e elementos estudados ao longo deste trabalho.

**Palavras-chave:** Chatbot. Comércio Eletrônico. Plataformas de criação.

## **ABSTRACT**

### ***A STUDY ABOUT CHATBOTS AND ITS APPLICATION ON E-COMMERCE***

*AUTHOR:* Denis Bao Motta

*ADVISOR:* Daniel Lichtnow

*This paper presents a study about the conversational agents called chatbots, discussing their history and evolution since its appearance, alongside the first concepts of Artificial Intelligence and its firsts conceptions imagined by Alan Turing of a dialogue between humans and machines, until its recent popularization through applications in several areas such as education, health, entertainment and, mainly, in the electronic commerce. Through the analysis of some featured applications of this evolutionary process, such as Eliza, Jabberwacky, ALICE, SmartChild, and others, a study was hold in order to understand the operation and usability of some of these technologies that, through the years, brought new potentials to this type of tool. The paper also makes an approach on the emergence of conversational commerce and the use of chatbots in messaging applications by service and commercial companies seeking to reach this emerging consumer market. It is also exemplified in this paper the creation of one of these conversational agents through the IBM Watson's system and its API Conversation, one of the many chatbot platforms that is noticeably getting popular as it grows the interest of the companies in making use of this digital solution. Lastly, it is described the development, tools and technology used for the construction of a chatbot used for e-commerce and integrated with the social network Facebook through the Messenger platform and Wit.ai system, able to exemplify concepts and elements learned along this paper.*

**Keywords:** *Chatbot. E-commerce. Chatbot platforms.*



## LISTA DE FIGURAS

FIGURA 1 – EXEMPLO DE CONVERSAÇÃO COM O PROGRAMA <i>ELIZA</i> .....	17
FIGURA 2 – ESTRUTURA BÁSICA DE UM DOCUMENTO AIML.....	20
FIGURA 3 – EXEMPLO DE REDUNDÂNCIA EM CATEGORIAS DA LINGUAGEM AIML. ....	21
FIGURA 4 – PROCESSAMENTO DE SENTENÇA PARA SUA ESTRUTURA SINTÁTICA E FORMA LÓGICA. ....	22
FIGURA 5 – EXEMPLOS DA TOKENIZAÇÃO LEXICAL E SENTENCIAL. ....	23
FIGURA 6 – ÁRVORES GERADAS PELO ANALISADOR SINTÁTICO LXPARSER. ....	26
FIGURA 7 – DISTINÇÃO DE CLASSES GRAMATICAIS DAS PALAVRAS “IA” E ÍA.....	28
FIGURA 8 – CONFIGURAÇÃO DA INTENÇÃO #PEDIR_PIZZA NA API CONVERSATION.....	30
FIGURA 9 – CONFIGURAÇÃO DA ENTIDADE “@SABOR” NA API CONVERSATION. ....	30
FIGURA 10 – EXEMPLIFICAÇÃO DA ORGANIZAÇÃO DE NÓS DE DIÁLOGO NA API CONVERSATION.....	31
FIGURA 11 – JANELA DE CONFIGURAÇÃO DO NÓ DE DIÁLOGO “PEDIDO DE PIZZA”.....	32
FIGURA 12 – JANELA DE CONFIGURAÇÃO DE SLOT DISPONIBILIZADA PELA API CONVERSATION. ....	33
FIGURA 13 – JANELA DE CONFIGURAÇÃO DOS <i>HANDLERS</i> DO NÓ “PEDIDO DE PIZZA”.....	34
FIGURA 14 – FLUXO DE DIÁLOGO FINAL, DEMONSTRANDO AS INTENÇÕES E ENTIDADES DETECTADAS, VARIÁVEIS DE CONTEXTO E AS RESPOSTAS FORNECIDAS PELO SISTEMA WATSON. ....	35
FIGURA 15 – EXEMPLO DE ERROS DE INTERPRETAÇÃO EM UM <i>CHATBOT</i> ORIENTADO A CONVERSAÇÃO. ....	38
FIGURA 16 – ATITUDE DE USUÁRIOS DE <i>SMARTPHONES</i> EM RELAÇÃO AO USO DE <i>CHATBOTS</i> . ....	39
FIGURA 17 – ORGANIZAÇÃO DAS TECNOLOGIAS UTILIZADAS E FLUXO DA TROCA DE INFORMAÇÕES.....	42
FIGURA 18 – ESTRUTURA BÁSICA DE UMA REQUISIÇÃO JSON PARA O ENVIO DE TEXTO PARA A PLATAFORMA <i>MESSENGER</i> . ....	42
FIGURA 19 – CONFIGURAÇÃO DA PLATAFORMA <i>MESSENGER</i> E DEFINIÇÃO DA CHAVE DE ACESSO PARA A PÁGINA DESEJADA. ....	43
FIGURA 20 – CONFIGURAÇÃO DA PLATAFORMA <i>MESSENGER</i> E DEFINIÇÕES OBRIGATÓRIAS DOS <i>WEBHOOKS</i> . .....	44
FIGURA 21 – INTERFACE PARA HABILITAÇÃO DA INTEGRAÇÃO ENTRE A PLATAFORMA <i>MESSENGER</i> E O SISTEMA <i>WIT.AI</i> . ....	44
FIGURA 22 – INTERFACE DO SISTEMA <i>WIT.AI</i> LISTANDO EXEMPLOS DE FRASES FORNECIDAS PELO DESENVOLVEDOR E A DETECÇÃO DA <i>TRAIT ENTITIE</i> “SAUDAÇÃO”.....	45
FIGURA 23 – CONTEÚDO DO CAMPO “ <i>NLP</i> ” GERADO PELO SISTEMA <i>WIT.AI</i> IDENTIFICANDO AS ENTIDADES “SAUDAÇÃO” (A) E “DESPEDIDA” (B).....	46
FIGURA 24 – EXEMPLO DE UMA REQUISIÇÃO RECEBIDA PELA API DE COMUNICAÇÃO (A) E DA RESPOSTA GERADA (B). ....	47
FIGURA 25 – MODELO DO BANCO DE DADOS DO PAINEL ADMINISTRATIVO DO PROJETO.....	48
FIGURA 26 – INTERFACE DO PAINEL ADMINISTRATIVO EXIBINDO A CONFIGURAÇÃO DO <i>POSTBACK</i> “ <i>INICIAR</i> ”. .....	49
FIGURA 27 – SEQUÊNCIA DE FLUXO DE INTERAÇÃO: “VER COMPUTADORES” > “VER DESKTOPS” > “ <i>IMAC</i> ”, CONFIGURADA ATRAVÉS DO PAINEL ADMINISTRATIVO DA APLICAÇÃO. ....	50
FIGURA 28 – SEQUÊNCIA DE ENVIO DE SUGESTÕES ATRAVÉS DA UTILIZAÇÃO DE ESTADOS PELA PLATAFORMA <i>MESSENGER</i> . ....	51
FIGURA 29 – INTERFACE DO PAINEL ADMINISTRATIVO MOSTRANDO A CONSTRUÇÃO DO MENU INTITULADO “ <i>DEFAULT</i> ”. ....	52
FIGURA 30 – SEQUÊNCIA DE USO DA ESTRUTURA DE MENU CRIADA NO PAINEL ADMINISTRATIVO DA APLICAÇÃO. ....	53
FIGURA 31- RESPOSTAS DO <i>CHATBOT</i> A DETECÇÃO DAS ENTIDADES “SAUDAÇÃO” (A) E “DESPEDIDA” (B). 53	

## LISTA DE TABELAS

TABELA 1 – RELAÇÃO DE PLATAFORMAS DE DESENVOLVIMENTO DE CHATBOTS .....	36
--	----

## LISTA DE ABREVIATURAS E SIGLAS

AIML - *Artificial Intelligence Markup Language*

API - *Application Programming Interface*

IA - *Inteligência Artificial*

JSON - *JavaScript Object Notation*

PLN - *Processamento de Linguagem Natural*

REST - *Representational State Transfer*

SETL - *Set Theory and Mathematical Logic*

SPA - *Single Page Application*

URL - *Uniform Resource Locator*

XML - *eXtensible Markup Language*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1	CONTEXTUALIZAÇÃO DO PROBLEMA .....	12
1.1.1	<i>Problema</i> .....	13
1.2	OBJETIVOS .....	13
1.2.1	<i>Objetivo Geral</i> .....	13
1.2.2	<i>Objetivos Específicos</i> .....	13
1.3	JUSTIFICATIVA .....	14
1.4	ORGANIZAÇÃO DO TRABALHO .....	14
<b>2</b>	<b>O QUE SÃO <i>CHATBOTS</i> E DE ONDE SURTIRAM.....</b>	<b>15</b>
2.1	HISTÓRIA E EVOLUÇÃO .....	16
<b>3</b>	<b>TECNOLOGIAS DE DESENVOLVIMENTO.....</b>	<b>19</b>
3.1	ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE (AIML) .....	19
3.2	PROCESSAMENTO DE LINGUAGEM NATURAL (PLN).....	22
3.2.1	<i>Construção da Forma Lógica</i> .....	22
3.2.2	<i>Pré-processamento de sentenças</i> .....	23
3.2.3	<i>Processamento Léxico</i> .....	25
3.2.4	<i>Processamento Sintático</i> .....	26
3.2.5	<i>Análise Semântica</i> .....	26
3.2.6	<i>Uso do Processamento de Linguagem Natural em Chatbots</i> .....	27
3.3	IBM <i>WATSON</i> E A API <i>CONVERSATION</i> .....	28
3.3.1	<i>Construção de um Chatbot e Exemplo de Uso</i> .....	29
3.3.2	<i>Configuração dos Nós de Diálogo</i> .....	31
3.4	OUTRAS PLATAFORMAS DE DESENVOLVIMENTO .....	36
<b>4</b>	<b>O USO DE <i>CHATBOTS</i> NO COMÉRCIO ELETRÔNICO.....</b>	<b>37</b>
4.1	DESAFIOS DA APLICAÇÃO .....	38
<b>5</b>	<b>CONSTRUÇÃO DE UM <i>CHATBOT</i> PARA E-COMMERCE NA REDE SOCIAL <i>FACEBOOK</i>.....</b>	<b>41</b>
5.1	CONFIGURAÇÃO E UTILIZAÇÃO DA PLATAFORMA <i>MESSANGER</i> E DO SISTEMA <i>WIT.AI</i> .....	42
5.2	DESENVOLVIMENTO DA API DE COMUNICAÇÃO .....	46
5.3	DESENVOLVIMENTO DO PAINEL ADMINISTRATIVO .....	47
5.3.1	<i>Gerenciamento de Postbacks</i> .....	48
5.3.2	<i>Estados de espera para captura de entradas do usuário</i> .....	50
5.3.3	<i>Gerenciamento de menus</i> .....	51
5.3.4	<i>Tomada de decisões com base no PLN</i> .....	53
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>55</b>

# 1 INTRODUÇÃO

Da mesma forma que humanos usam a linguagem para se comunicar uns com os outros, eles também anseiam por utilizar este método para se comunicarem com computadores, permitindo a eles expressarem seus interesses e dúvidas da maneira mais natural possível, através da seleção de itens, texto ou até mesmo através da fala (ZANDROZNY et al., 2000).

Essa é a motivação original por trás do surgimento dos *chatbots*, sistemas capazes de interagir com usuários através do uso de uma linguagem natural, simulada através da integração de algoritmos computacionais com um modelo de linguagem base. Essa tecnologia começou a ser explorada nos anos 60 com objetivo de testar a capacidade de programas em se passarem por humanos de forma a enganar o próprio usuário.

Entretanto, estes agentes de conversação não existem apenas para imitar a comunicação humana e entreter seus usuários. Sua gama de aplicações inclui áreas como Educação, Recuperação de Informação, *Marketing* Digital, Gerenciamento de Atendimento ao Consumidor, entre muitas outras. Seu desenvolvimento pode partir de princípios simples, baseados em regras, perguntas predefinidas e comandos fechados de forma a guiar o usuário por um conjunto de fluxogramas funcionais, mas sua construção também pode contemplar conceitos como Inteligência Artificial (IA) e *Machine Learning*.

Neste trabalho, serão abordados temas relacionados ao estudo das tecnologias envolvidas no funcionamento e desenvolvimento de *chatbots* que ao longo de seus anos de evolução agregaram cada vez mais potencial a este tipo de ferramenta, principalmente em sua crescente utilização no comércio eletrônico. Também é analisado o uso das plataformas de desenvolvimento de *chatbots* que veem se popularizando conforme cresce o interesse das empresas em fazer uso desta solução digital.

## 1.1 Contextualização do Problema

O comércio eletrônico é um segmento que vem desenvolvendo-se em todo o mundo. No Brasil, ele vem crescendo mesmo durante o período de recessão econômica, fechando o ano de 2016 com um faturamento de aproximadamente R\$ 44,4 bilhões, representando um crescimento de 7,4%, segundo o *site* Ebit<sup>1</sup>.

Diante do aumento da concorrência no segmento, as empresas precisam buscar diferenciais para atrair os clientes, bem como estratégias que ajudem no processo de retenção destes consumidores.

---

<sup>1</sup> Ebit – empresa que mede a reputação das lojas virtuais por meio de pesquisas com consumidores reais, gerando dados estratégicos e táticas para o mercado online - <https://www.ebit.com.br/>. Acessado em 21 de outubro de 2017.

Com o avanço das tecnologias e diversificação dos canais de comunicação utilizados no comércio eletrônico, cresce a demanda dos consumidores por uma interação simples e imediata. Empresas que investem nesse segmento podem ter de enfrentar o desafio de suprir essa necessidade de seus clientes com qualidade e rapidez, evitando problemas como falta de atenção e o tempo de resposta no atendimento ao cliente.

### 1.1.1 Problema

A praticidade na aquisição de produtos e serviços disponibilizada pelo crescimento do comércio eletrônico cria um novo patamar no relacionamento entre empresas e consumidores. A criação de novas frentes de atendimento de qualidade que possam acompanhar as exigências de agilidade geradas por esse segmento é fundamental para conquista e fidelização de clientes.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Este trabalho tem como objetivo realizar um estudo das técnicas envolvidas no desenvolvimento dos agentes de conversação chamados *chatbots*. Assim como, a aplicação destas técnicas e conceitos no desenvolvimento destes *softwares*, através das plataformas de criação e configuração de *chatbots* disponíveis no mercado atualmente, tendo como foco sua aplicação no comércio eletrônico.

### 1.2.2 Objetivos Específicos

Dentre os objetivos buscados pela realização deste trabalho, podem ser citados mais pontualmente os seguintes:

- Caracterizar as áreas relacionadas ao desenvolvimento de *chatbots*, especialmente o Processamento de Linguagem Natural (PLN);
- Realizar um estudo teórico das tecnologias envolvidas na construção de *chatbots* e sua utilização nos canais de comércio eletrônico;
- Aplicar os conceitos e tecnologias estudados na implementação de softwares capazes de atender as necessidades básicas de uma empresa quanto ao atendimento ao consumidor em um canal digital de comunicação.

### 1.3 Justificativa

Com a expansão do comércio eletrônico e a consequente demanda dos consumidores por informação contínua e de forma rápida, o peso do atendimento ao consumidor tornou-se ainda mais importante na construção da imagem da empresa e na consolidação da confiança de seus clientes.

O uso de tecnologias que garantam a otimização de recursos tornou-se imprescindível para garantir a sustentabilidade de qualquer organização. Diminuir o tempo de resposta no atendimento é uma necessidade tanto do cliente quanto de qualquer empresa que zele pela eficiência operacional otimização de custos.

Nesse sentido, os *chatbots* podem oferecer valiosos recursos para que o atendimento ao consumidor seja feito de maneira rápida, simples e eficaz, garantindo que grande parte das solicitações feitas pelos consumidores sejam atendidas já no primeiro contato e de forma praticamente instantânea. Sua utilização como método de comunicação entre empresa e clientes é uma importante ferramenta e pode influenciar consideravelmente o fator de satisfação dos clientes em suas interações com as empresas.

### 1.4 Organização do Trabalho

A pesquisa está dividida em seis capítulos pertinentes aos objetivos do trabalho. O capítulo 2 descreve o que são os *chatbots*, assim como a história e evolução desse tipo de *software*. Já no capítulo 3, procura-se realizar uma revisão sobre algumas das tecnologias fundamentais que podem ser utilizadas em sua construção. Nele também é explorado um estudo de caso em que um *chatbot* orientado a conversação para atendimento ao consumidor é configurado através de uma plataforma de criação, a fim de explorar o funcionamento da ferramenta e a lógica envolvida em sua construção.

No capítulo 4, é estudado o conceito de comércio conversacional, como a utilização dos agentes de conversação pode impactar o comércio eletrônico e os desafios de implementação que esse segmento pode vir a apresentar.

O capítulo 5 traz a descrição do desenvolvimento de um *chatbot* orientado a regras com aplicação no comércio eletrônico em popular canal de comunicação. No capítulo 6 são apresentadas as considerações finais.

## 2 O QUE SÃO *CHATBOTS* E DE ONDE SURGIRAM

Os *chatbots* são programas capazes de simular uma conversa, da mesma forma que um humano é capaz de conversar com outro, podendo serem utilizados para os mais diversos fins, como na educação, comércio, pesquisa, saúde, entre muitos outros. Em alguns sites e aplicativos de mensagem, a presença de *chatbots* tem sido bastante frequente como atendentes virtuais, que recepcionam e respondem perguntas de clientes em tempo real.

De modo geral, um *chatbot* depende de dois elementos estruturais básicos: um núcleo de processamento capaz de reconhecer e processar as entradas do usuário e uma base de conhecimento ou base de regras. Através destas, o *chatbot* torna-se responsável pela composição do diálogo com o usuário através da análise e associação entre termos, determinando os caminhos possíveis para cada resposta. Esta análise pode ser baseada em simples comparações entre palavras-chave ou até mesmo em técnicas da área de Inteligência Artificial (NUNES, 2012).

*Chatbots* orientados a regras possuem uma série de associações em seu núcleo de processamento que fazem a comparação entre a entrada do usuário e as informações contidas em sua base de conhecimento para a seleção da resposta a ser enviada para o usuário. Se, por exemplo, o programa perguntar ao usuário como ele está, pode-se prever duas possíveis respostas: “estou bem” direcionaria o *chatbot* a dizer algo como “eu também estou bem”; a outra possível resposta seria: “não estou bem”, que por sua vez, direcionaria o programa a perguntar os motivos da indisposição do usuário e assim por diante. Quanto mais completa é a base de conhecimento, mais convincente pode ser a ilusão de compreensão (NUNES, 2012).

Já os *chatbots* baseados em IA fazem uso de diversas técnicas, dentre elas o Processamento de Linguagem Natural (PLN), um campo que une conhecimentos de Ciência da Computação, Linguística e Psicologia, com o objetivo de produzir sistemas capazes de “entender” a linguagem humana pelo processamento das entradas do usuário em tempo real, gerando respostas mais coerentes (NUNES, 2012).

Ao longo do tempo, diferenciadas técnicas foram empregadas na criação deste tipo de *software* buscando uma melhor forma de simulação de conversação e algumas delas serão abordadas neste capítulo.



## 2.1 História e Evolução

O ano de 2016 foi apontado como o ano dos assistentes virtuais inteligentes e segundo o site *MIT Technology Review*<sup>2</sup>, as interfaces conversacionais estiveram entre as dez tecnologias mais revolucionárias desde mesmo ano. À frente do segmento estão os assistentes digitais das quatro grandes marcas: a *Siri* da *Apple*, *Cortana* da *Microsoft*, *Alexa* da *Amazon* e o Assistente do *Google*. Dentro desse mesmo contexto podem ser citados os *chatbots*, funcionando como assistentes baseados em texto que buscam desempenhar o mesmo papel de auxílio a interação do usuário, porém focados em funcionalidades mais específicas.

Apesar desta recente tendência, os *chatbots* existem há mais tempo do que muitos imaginam. Sua história iniciou em 1950, quando as primeiras concepções de um diálogo entre humanos e máquinas começaram a ser imaginadas por Alan Turing, e como essas máquinas poderiam imitar o pensamento humano, levando às primeiras definições a cerca da Inteligência Artificial e ao seu trabalho, *Computing Machinery and Intelligence*, onde propôs um teste concebido para medir a capacidade de inteligência de máquinas, conhecido posteriormente como Teste de Turing (NUNES, 2012).

O trabalho de Alan Turing abriu as portas para esse campo de pesquisa. A partir desses conceitos e ideias surgiu o que hoje é considerado o primeiro *software* do gênero, o programa *Eliza*, desenvolvido em 1966 por Joseph Weizenbaum, autor e professor da área de ciências da computação no *Massachusetts Institute of Technology* (MIT). Concebido para imitar um psicanalista, o programa usava padrões bastante simples de reconhecimento de palavras-chave construindo suas saídas com base em um modelo de estímulo-resposta, com destaque para a técnica de transformação de pronomes pessoais comuns, utilizada posteriormente em outros programas similares. Resumidamente, frases como “Diga-me o que você pensa sobre mim” são transformadas pelo software em “Você quer que eu diga o que penso sobre você? ”, criando a ilusão de entendimento (Figura 1).

Apesar de sua simplicidade e cerca de 200 linhas de código, o programa foi capaz de enganar diversos colegas não técnicos de Weizenbaum que, surpreendido pelo fato de que as pessoas realmente acreditaram que *Eliza* poderia ajudá-las a resolver seus problemas psicológicos, passou a considerar seu programa uma ameaça, resultando no encerramento da versão inicial do projeto e posterior publicação de seu trabalho, *Computer Power and Human*

---

<sup>2</sup> <https://www.technologyreview.com/s/600766/10-breakthrough-technologies-2016-conversational-interfaces>. Acessado em 22 de outubro de 2017.

*Reason* em 1972, onde descreve o chamado “Efeito *Eliza*” e a tendência humana de antropomorfizar os dispositivos tecnológicos (NUNES, 2012).

Figura 1 – Exemplo de conversação com o programa *Eliza*

```

> Hello, I am Eliza.
* Hey!
> Don't you ever say Hello?
* Hello!
> How are you today... What would you like do discuss?
* I'm depressed.
> Do you enjoy being depressed?
* No, I don't.
> Why not?
* Because I want to be happy.
> Why you want to be happy?
* Because I'm tired of being depressed.
> That is quite interesting.
* I'm not sure.

```

Fonte: <https://blog.infermedica.com/introduction-to-chatbots-in-healthcare/>

Vários outros *chatbots* inspirados nos moldes criados por *Eliza* surgiram durante os anos subsequentes, incorporando novas tecnologias e aumentando cada vez mais a qualidade dos diálogos. Implementado pelo psiquiatra Kenneth Colby em 1972, o *chatbot Parry* também teve sucesso em seu objetivo de imitar o comportamento humano, porém através de uma abordagem contrária a seu antecessor. Programado para se passar por um paciente, ao invés de seu médico, *Parry* imitava um indivíduo paranoico que questionava tudo que ouvia. O programa foi capaz de enganar tanto psicanalistas quanto cientistas da computação que foram incapazes de distinguir o *chatbot* de um humano (MAULDIN, 1994).

No ano de 1988, é iniciado o projeto do *Jabberwacky*, uma das primeiras tentativas de criar uma IA através da interação humana, somando até o ano de 2005 mais de 7 milhões de interações com humanos. Durante seus anos de desenvolvimento, foi um dos primeiros projetos a buscar a migração de um sistema baseado em texto para um completamente baseado em voz (FRYER & CARPENTER, 2006).

Em 1995, Richard S. Wallace realiza o lançamento do *chatbot* denominado *Artificial Linguistic Internet Computer Entity* (A.L.I.C.E), o primeiro robô de conversação a introduzir a utilização da linguagem *Artificial Intelligence Markup Language* (AIML) na construção de sua base de conhecimento, tecnologia que será melhor abordada no capítulo 4 deste trabalho.

Em 2001, surge o *SmarterChild*, considerado o precursor dos assistentes atuais como *Apple Siri* e o *Samsung S Voice*, foi um dos primeiros a incorporar recursos de processamento de linguagem natural.

Neste mesmo ano, no Brasil, foi lançado Sete Zoom<sup>3</sup>, desenvolvido pela empresa brasileira *Insite*<sup>4</sup>, atuava como garota propaganda da marca *CloseUp* e realizava o atendimento ao consumidor no *website* da marca. Além de incorporar recursos de IA, foi o primeiro *chatbot* a ser capaz de realizar uma conversa através de mensagens instantâneas e possuía conhecimento sobre mais de 5000 tópicos diferentes que o tornavam capaz de responder com sucesso cerca de 95% das entradas dos usuários.

A evolução se manteve constante e nos últimos anos os *chatbots* passaram a chamar a atenção das grandes empresas fazendo com que novos *softwares* começassem a chegar ao mercado. Em 2011 a *Apple* introduz o sistema *Siri*, em 2012 o *Google* lança o *Google Now*, em 2013 a *Microsoft* apresenta pela primeira vez o *Cortana* e em 2014 a *Amazon* apresenta pela primeira vez o sistema *Alexa* (MUTCHLER, 2017).

---

<sup>3</sup> <https://www.inbot.com.br/cases/setezoom/>. Acessado em 23 de outubro de 2017.

<sup>4</sup> <http://www.insite.com.br/>. Acessado em 23 de outubro de 2017.

### 3 TECNOLOGIAS DE DESENVOLVIMENTO

Assim como suas áreas de aplicação, os conceitos e tecnologias envolvidas no desenvolvimento dos *chatbots* são bastante variados, passando por conceitos mais simples que não obrigatoriamente levam em consideração o entendimento profundo da linguagem utilizada na comunicação entre o software e o usuário, até as áreas mais complexas dos estudos da IA.

A aplicação de tais conceitos no desenvolvimento deste tipo de *software*, embora não sejam sempre necessários, são importantes para a construção de *chatbots* que busquem o aprofundamento da sua capacidade em responder com coerência à interação do usuário e que para isso, precisem expandir sua base de conhecimento. Neste capítulo, serão descritas algumas dessas tecnologias, assim como um estudo de seus conceitos fundamentais e sua aplicação na construção dos robôs de conversação.

#### 3.1 Artificial Intelligence Markup Language (AIML)

A linguagem AIML foi desenvolvida pela comunidade de *software* livre *Alicebot* em conjunto com o Dr. Richard S. Wallace, desenvolvedor do *chatbot* A.L.I.C.E, durante os anos de 1995 a 2000. Seu objetivo principal era codificar os conceitos mínimos necessários para operação de um sistema de conhecimento baseado em estímulo-resposta em uma linguagem legível para humanos, de fácil aprendizagem e sem dependências com outros tipos de linguagem (NOEL, 2001).

Ela consiste em um conjunto de *tags*, baseadas na *eXtensible Markup Language* (XML), utilizadas para representar e relacionar expressões em linguagem natural, permitindo a criação de bases de conhecimento capazes de alimentar *softwares* de conversação de forma a torná-los capazes de manter um diálogo simples. Essas bases são constituídas por diferentes categorias representando diferentes áreas de conhecimento. A base oficial da linguagem AIML é constituída por aproximadamente 45.729 dessas categorias distribuídas em em torno de 50 arquivos classificados por tema ou pela estrutura formal de suas categorias (RODRIGUES e ALMEIDA, 2004).

*Chatbots* baseados nessa linguagem possuem um módulo chamado interpretador AIML, responsável por identificar a entrada do usuário na base e retornar a resposta adequada. Esses interpretadores estão disponíveis em várias linguagens de programação como por exemplo o *Program D* implementado em Java; *Program M* implementado em *Set Theory and*

*Mathematical Logic* (SETL); *Program E* implementado em PHP; *Program V* implementado em Perl e o *Program P* desenvolvido em *Delphi* (NETO et al., 2003).

Cada arquivo AIML é composto por quatro marcações base: `<aiml>`, que demarca o início e fim do documento; `<category>`, que define uma unidade de conhecimento dentro do escopo da base de conhecimento; `<pattern>`, que define a amostra para comparação com a entrada do usuário e, por fim a marcação `<template>`, que define as possíveis respostas que o sistema retornará para o usuário (Figura 2).

Figura 2 – Estrutura básica de um documento AIML.

```
<aiml version="1.0">
<category>
  <pattern>OI</pattern>
  <template>
    <random>
      <li>Oi, qual é o seu nome?</li>
      <li>Oi, meu nome é Robô</li>
      <li>Oi</li>
    </random>
  </template>
</category>
</aiml>
```

Fonte: <http://abelcorreadias.blogspot.com.br/2011/09/linguagem-aiml-aplicada-com-mecanismo.html>

Além das marcações básicas, a linguagem AIML conta com mais de 20 outras *tags* para construção da solução para entrada recebida, dentre elas, podem ser citadas as *tags* `<random>` e `<li>` utilizadas para diversificação da resposta do *chatbot* diante da entrada do usuário. O uso de curingas e variáveis em conjunto com as *tags* `<think>`, `<set>`, `<star>` e `<srar>` também podem ser utilizadas para melhorar a capacidade de entendimento do *chatbot*, tornando-o mais flexível quanto às possibilidades de escrita dos usuários e armazenamento de valores para simular sua memória.

Mesmo com esses recursos, a linguagem AIML possui algumas falhas que podem levar a ocorrência de alguns desvios no fluxo do diálogo. Por padrão, ela não leva em conta a estrutura global de uma conversação (abertura, desenvolvimento e fechamento), fazendo com que muitas sentenças sejam tratadas como desconhecidas, quando na realidade fazem referência a categorias adjacentes à outras tratadas anteriormente pelo *chatbot*. Outro grave problema é a implementação bastante verbosa resultante da comparação de palavras primitiva utilizada pela linguagem, levando o desenvolvedor a utilizar um grande número de categorias redundantes para a resolução de algumas entradas (WILCOX, 2008).

Ainda podem ser citadas a ausência de combinação de sinônimos em uma única especificação tornando necessário a utilização de termos específicos, resultando por exemplo, na necessidade de especificar categorias distintas para as palavras “acho” e “acredito”, e a impossibilidade da utilização de palavras-chaves no diálogo devido a possibilidade de não existirem categorias associadas a elas (WILCOX, 2008). Estes tipos de problema podem gerar a necessidade do uso de caracteres coringa, representados pelo caractere especial “\*”, para estabelecer a possibilidade de um tipo de entrada qualquer associada a uma entrada específica (Figura 3).

Figura 3 – Exemplo de redundância em categorias da linguagem AIML.

```

<category>
  <pattern>MOTHER</pattern>
  <template>Tell me more about your family.</template>
</category>

<category>
  <pattern>* MOTHER</pattern>
  <template><srail>MOTHER</srail></template>
</category>

<category>
  <pattern>MOTHER *</pattern>
  <template><srail>MOTHER</srail></template>
</category>

<category>
  <pattern>* MOTHER *</pattern>
  <template><srail>MOTHER</srail></template>
</category>

```

Fonte: RODRIGES e ALMEIDA, 2004.

Para contornar estes tipos de problemas, foram propostas algumas modificações estruturais, resultando em uma variação da tradicional linguagem AIML denominada iAIML, capaz de simular a intenção do diálogo do usuário. Nela, a base de conhecimento é redistribuída em 54 arquivos criados de acordo com a intenção dos pares estímulo-resposta e subdivididos em categorias de intenções típicas de abertura, desenvolvimento e fechamento (NEVES, 2004).

A iAIML faz uso de recursos compatíveis com a linguagem tradicional que permitem que o desenvolvedor estabeleça um controle com base no andamento global do diálogo através do uso de variáveis:

- *session*: responsável por identificar o posicionamento do usuário no escopo global da base de conhecimento (abertura, desenvolvimento ou fechamento);

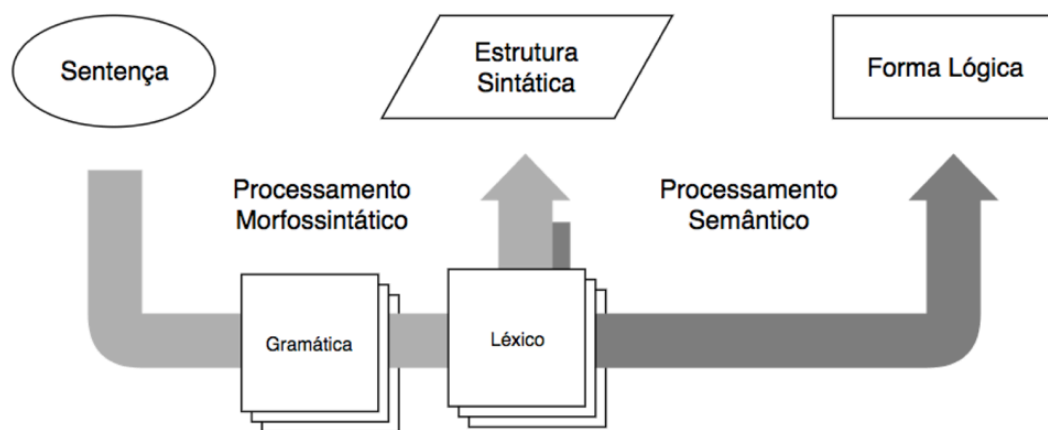
- *user\_intention*: utilizada para o controle de repetições das entradas dos usuários, avaliando suas similaridades em função da informação de intenção;
- *bot\_intention*: utilizada na tentativa de inferir a intenção do usuário nos turnos em que a mesma não é clara para o *chatbot*.

### 3.2 Processamento de Linguagem Natural (PLN)

O PLN é uma subárea da IA que estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos com o objetivo de fornecer aos computadores a capacidade de não somente entender, mas também compor textos (GONZALEZ & LIMA, 2003).

Para a construção desse entendimento são aplicadas diversificadas técnicas como as análises sintática, semântica, léxica e morfológica, com o objetivo de transformar uma sentença em sua forma lógica (Figura 4). Esse processo pode atingir níveis ainda mais complexos, como o reconhecimento de contexto, interpretação de sentidos, análise de sentimentos e até mesmo o aprendizado de conceitos a partir dos textos processados.

Figura 4 – Processamento de sentença para sua estrutura sintática e forma lógica.



Fonte: GONZALEZ & LIMA, 2003

#### 3.2.1 Construção da Forma Lógica

A representação do significado de uma sentença no PLN, independente de contexto, é obtida através de sua forma lógica, a qual codifica os possíveis sentidos de cada palavra e identifica os relacionamentos semânticos entre palavras e frases (ALLEN, 1995; GONZALEZ & LIMA, 2003).

Através do processamento morfossintático pode-se obter a estrutura sintática de uma sentença, a qual também faz uso de uma gramática para definição das regras gramaticais e da análise léxica para fornecer as demais informações necessárias, como as categorias morfológicas das palavras. A partir dessa estrutura sintática, a forma lógica da sentença é alcançada através de um processamento semântico que também faz uso das informações concedidas pela análise léxica.

### 3.2.2 Pré-processamento de sentenças

Para que a modelagem da linguagem possibilite que a máquina a interprete, são aplicadas uma série de técnicas de pré-processamento de cunho morfossintático, responsáveis pela abstração da estrutura da linguagem, filtrando a informação e identificando os itens mais relevantes. Este processo reduz o tamanho do vocabulário necessário para o entendimento da língua, além de tornar os dados menos esparsos e conseqüentemente mais convenientes para o processamento computacional.

O processamento morfossintático faz uso de uma gramática que armazena o conjunto de regras para a correta formação de palavras e sentenças de uma determinada língua. Seu papel neste processo é realizar a validação das sentenças de maneira genérica, seletiva e inteligível, aceitando um grande número de entradas válidas, reconhecendo casos problemáticos e fornecendo de forma simples o entendimento de suas regras, respectivamente (GONZALEZ & LIMA, 2003).

#### 3.2.2.1 Normalização

O processo de normalização abrange tratativas como a tokenização, que tem como objetivo separar palavras ou sentenças em unidades. A tokenização lexical marca cada palavra de uma sentença como um *token* (Figura 5A), enquanto a tokenização sentencial identifica as diferentes sentenças (Figura 5B).

Figura 5 – Exemplos da tokenização lexical e sentencial.

- A) Esta é uma sentença.  
[ 'esta' 'é' 'uma' 'sentença' '.' ]
- B) Esta é a primeira sentença. Esta é a segunda!  
[ 'Esta é a primeira sentença.' 'Esta é a segunda!' ]



A normalização vai além do processo de tokenização, passando por diferentes etapas como a transformação de letras maiúsculas para minúsculas, remoção de caracteres especiais e remoção de *tags* relacionadas a outras linguagens como HTML, Javascript, CSS, dentre outras. A partir desse processo é iniciada a estruturação do texto, que é de extrema importância para as etapas seguintes do pré-processamento que atuam sobre as unidades sentenciais e lexicais criadas durante este processo (GONZALEZ & LIMA, 2003).

### 3.2.2.2 Remoção de *Stopwords* e Numerais

A remoção dos chamados *stopwords* não é um método obrigatório no pré-processamento e deve ser aplicado de acordo com o contexto da aplicação. *Stopwords* são palavras que aparecem frequentemente nos textos e que, na maioria dos casos, não adicionam informações relevantes, como por exemplo: “a”, “de”, “o”, “da”, “que”, “e”, “do”.

Em alguns casos, a remoção destas *stopwords* pode prejudicar a interpretação do conteúdo, como por exemplo, a remoção da palavra “não” pode prejudicar a interpretação da conotação de negatividade que ela fornece a uma sentença.

Os numerais constituem outra informação passível de remoção, pois geralmente não agregam informações relevantes visto que não possuem uma carga semântica. Nesta categoria podem ser incluídos os símbolos que os acompanham, como “R\$”, “\$”, “US\$”, “kg”, “km”, entre outros.

### 3.2.2.3 Stemização e Lematização

A stemização, palavra derivada do inglês *stemming*, é um processo que consiste na redução de uma palavra ao seu radical (GONZALEZ & LIMA, 2003). Por exemplo, a palavra “meninas” se reduziria a “menin”, assim como “meninos” e “meninhos”, assim como as palavras “gato”, “gata”, “gatos” e “gatas” podem ser reduzidas para o radical “gat”.

Já a lematização consiste na redução da palavra ao seu lema, que é a transformação da palavra para sua forma no gênero masculino e singular. Por exemplo, as palavras “gato”, “gata”, “gatos” e “gatas” são todas formas do mesmo lema: “gato”. No caso de verbos, a lematização é feita através da mudança da forma nominal para o infinitivo, como por exemplo, os verbos “tiver”, “tenho”, “tinha”, “tem” são formas do mesmo lema “ter”.

A aplicação destas técnicas tem como objetivo reduzir o vocabulário, assim como abstrair o significado das palavras, agilizando o processamento e facilitando a interpretação do conteúdo.

### 3.2.3 Processamento Léxico

O termo “léxico” significa uma relação de palavras com suas categorias gramaticais e seus significados. O léxico, enquanto componente de um sistema de Processamento de Linguagem Natural, forma uma base de dados detendo informações semânticas e gramaticais sobre esses itens lexicais (SCAPINI, 1995).

Também chamado de dicionário por alguns autores, o léxico tem como papel disponibilizar o maior número de informações possíveis sobre as palavras, como por exemplo a etimologia, pronúncia, morfologia, sintaxe, definições de sentidos, entre outras (GUTHRIE et al., 1996).

Os léxicos podem ser classificados, de acordo com seu conteúdo, em: (1) convencionais, com as palavras organizadas em ordem alfabética; (2) analógicos, com os itens lexicais organizados pelo significado; (3) etimológicos, que tratam da origem das palavras; (4) morfológicos, contendo as formas flexionais dos lexemas; e (5) sinônimos e antônimos, com as listas de palavras semelhantes também com significado oposto (GONZALEZ & LIMA, 2003).

Também podem ser classificados de acordo com seu objetivo em: (1) dicionário padrão, contendo o significado das palavras; (2) thesauri, contendo o relacionamento entre as palavras; (3) bilíngues, que fazem o relacionamento entre palavras de diferentes idiomas; (4) dicionários de estilo, que orientam sobre o bom uso de regras gramaticais; e (5) dicionários de concordância, essencialmente ferramentas escolares (WILKS et al., 1996).

Como exemplo de léxico, pode-se citar o *WordNet*<sup>5</sup>, uma base de dados lexical que organiza seus termos por significado, dividindo-os em substantivos, verbos, adjetivos e advérbios. Este léxico apresenta os itens de acordo com suas definições, sentidos e relações, usando conjuntos de sinônimos para construir um relacionamento semântico básico

Com as regras gramaticais definidas e com a base de dados lexicais formada, a construção da forma lógica de uma sentença pode ser iniciada através do processamento sintático (GONZALEZ & LIMA, 2003).

---

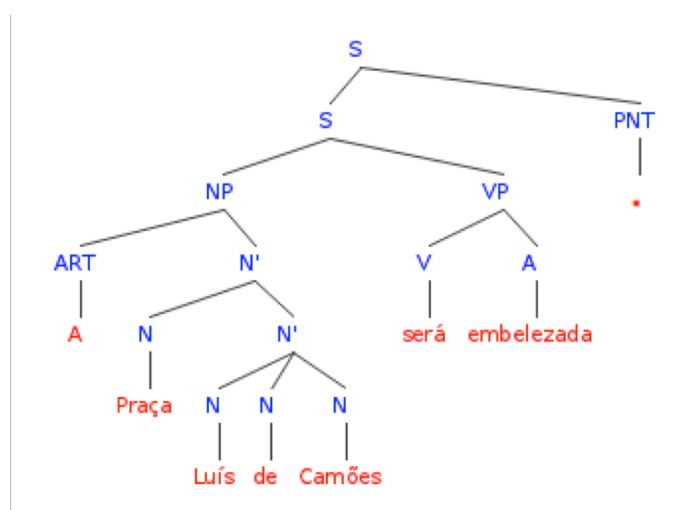
<sup>5</sup> <https://wordnet.princeton.edu/>. Acessado em 28 de outubro de 2017.

### 3.2.4 Processamento Sintático

Através da análise sintática, são avaliados os modos como as regras gramaticais podem ser agrupadas de forma a gerar uma estrutura de árvore que seja capaz de representar a estrutura sintática de uma sentença (Figura 6).

Neste processo são analisadas a constituição das palavras e os grupos que formam os elementos de expressão de uma determinada língua, e por sua vez, a constituição das frases. Essa análise, também chamada de *Parsing*, é realizada por um analisador sintático chamado *Parser* (GONZALEZ & LIMA, 2003).

Figura 6 – Árvores geradas pelo analisador sintático LXParser.



Fonte: <http://lxcenter.di.fc.ul.pt/services/pt/LXParserPT.html>

A análise sintática pode ser vista como projeção da análise semântica, ou ainda como a base utilizada para a interpretação criada durante esta. Independente do ponto de vista, o processamento sintático é indispensável para a construção do processamento semântico (NUNES, 2012).

### 3.2.5 Análise Semântica

O processamento semântico está relacionado ao significado, não só das palavras, mas também das sentenças formadas por elas. Este é um dos maiores desafios do processamento de linguagem natural e a última etapa no processo de construção da forma lógica em uma sentença (SAINT-DIZIER, 1999).

A descoberta do significado de qualquer construção em uma linguagem parte do princípio de que esse depende intimamente do significado de cada um de seus componentes e

das relações entre seus itens lexicais (ALLEN, 1995). Este processo passa por uma série de análises de cunho gramatical, considerando as diversas interações entre palavras como, por exemplo, algumas das relações paradigmáticas:

- Sinonímia: palavras de significado semelhantes, como “*distante*” e “*afastado*”;
- Antonímia: palavras de significado opostos, como “*economizar*” e “*gastar*”;
- Hiperonímia: representam o conceito geral, como “*fruta*”;
- Hiponímia: representam o conceito específico, como “*laranja*” ou “*abacate*”;
- Holonímia: representam o conceito de um todo, como “*carro*”;
- Meronímia: representam o conceito de uma parte do todo, como “*volante*”.

Com o objetivo de minimizar a complexidade deste processamento, ainda podem ser utilizadas diferentes abordagens. O conceito léxico hierárquico procura focar a análise em um léxico ao invés das regras gramaticais, fazendo com que os sentidos dos verbos sejam organizados de forma a permitir que seus significados sejam herdados de umas classes para outras (ALLEN, 1995). Já a teoria do léxico gerativo, introduz recursos que consideram a semântica das palavras isoladas e a capacidade de composição entre elas, abordando características polimórficas da linguagem, semanticalidade das expressões, uso criativo das palavras em novos contextos, entre outros (PUSTEJOVSKY, 1995).

Por fim, análise semântica de sentenças é um processo complexo que precisa enfrentar diversos obstáculos impostos pelas linguagens, variações linguísticas e ambiguidades, polimorfismos, composições, entre outros fenômenos linguísticos, fazem com que a transformação de uma sentença em sua correspondente forma lógica não seja um processo trivial.

### 3.2.6 Uso do Processamento de Linguagem Natural em *Chatbots*

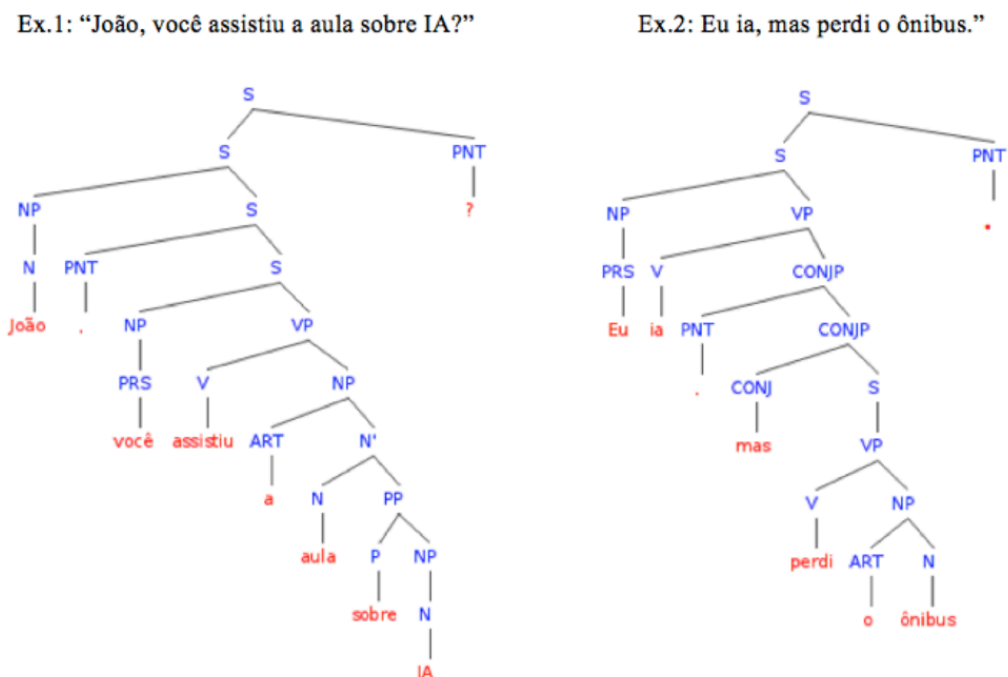
Como discutido anteriormente, a aplicação de técnicas compreendidas no campo de estudos da IA, como o PLN, nem sempre são obrigatórias ao desenvolvimento dos *softwares* de conversação. Entretanto, suas aplicações, conforme o contexto de utilização deste tipo de ferramenta, podem proporcionar um desempenho mais satisfatório em relação a experiência do usuário.

Mais especificadamente, aplicação das técnicas de PLN na análise de sentenças em *chatbots* vem ao encontro das necessidades não contempladas por métodos que apenas analisam os termos da conversação de maneira independente, como a ambiguidade, que pode

causar erros de interpretação por parte do *chatbot*, ocasionando respostas fora de contexto que acabam por distorcer o fluxo da conversação.

Um exemplo desse tipo de situação é descrito em Lichtnow (2006), onde é demonstrada a ambiguidade da utilização dos termos “*ia*”, verbo e “*IA*”, abreviatura de inteligência artificial. Neste caso, a utilização do PLN seria capaz de fornecer uma distinção entre os termos segundo suas respectivas categorias gramaticais através da análise sintática dos termos (Figura 7).

Figura 7 – Distinção de classes gramaticais das palavras “*ia*” e *IA*.



Fonte: <http://lxcenter.di.fc.ul.pt/services/pt/LXParserPT.html>

### 3.3 IBM *Watson* e a *API Conversation*

Com o crescimento das plataformas de comércio eletrônico e redes sociais e o subsequente crescimento da utilização de *chatbots* para o atendimento de usuários por parte das empresas que investem nestes segmentos, surge a demanda por plataformas de desenvolvimento que sejam capazes de criar *chatbots* de qualidade e de maneira acessível e rápida.

O IBM *Watson*<sup>6</sup> é um sistema de programação cognitiva que começou a ser desenvolvido pela IBM em 2007. Através da aplicação de variadas técnicas como:

<sup>6</sup> <https://www.ibm.com/watson/>. Acessado em 28 de outubro de 2017.

computação cognitiva, algoritmos complexos de inteligência artificial baseados em redes neurais, *deep learning* e avançadas técnicas de processamento de linguagem natural, ele é capaz de interpretar dados não estruturados originados da *web* em qualquer formato, seja texto, imagem ou vídeo, utilizando essas informações para construção e expansão de sua base de conhecimentos.

Para ser capaz de interpretar sentenças complexas em linguagem natural em um curto período de processamento, o sistema foi implementado utilizando uma arquitetura chamada *DeepQA*, também desenvolvida pela IBM. Através dessa, o sistema busca elaborar múltiplas interpretações para uma sentença, gerando várias possíveis respostas, ou hipóteses, a partir das quais coleta evidências que possam justificá-las e por fim, valida estas evidências para determinar se elas apoiam ou refutam uma determinada hipótese (LALLY e FODOR, 2011).

Entre as funcionalidades do sistema *Watson* está a plataforma de construção de *chatbots* chamada *Conversation*<sup>7</sup>, disponibilizada através de uma API (*Application Programming Interfaces*) através da qual desenvolvedores podem usufruir das vantagens de processamento oferecidas pelo sistema *Watson* na criação de *chatbots* personalizáveis capazes de entender diferentes linguagens naturais.

### 3.3.1 Construção de um *Chatbot* e Exemplo de Uso

Afim de conhecer um pouco mais a fundo as funcionalidades disponibilizadas pela API *Conversation* do sistema *Watson*, será explorado o exemplo de uso dessa plataforma na criação de um *chatbot* para atendimento de pedidos em uma pizzeria.

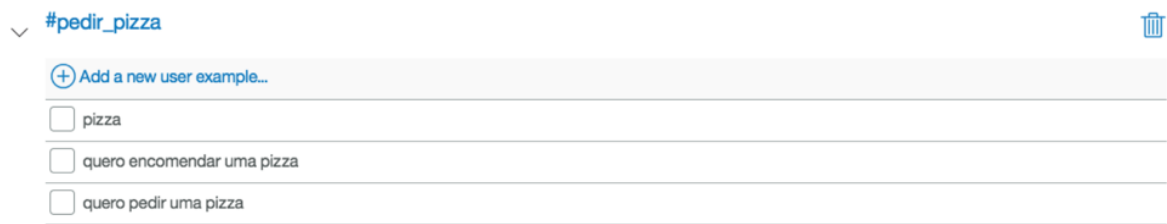
Fundamentalmente, a API *Conversation* separa a criação do *chatbot* em três conceitos fundamentais: *Intents*, *Entities* e *Dialog*, através dos quais a construção do fluxo da conversação é configurada pelo desenvolvedor, tornando-se a base de onde o sistema *Watson* retira as informações para criar o seu entendimento da conversa.

As *Intents*, ou intenções, estão relacionadas a detecção do propósito do usuário, que é o primeiro passo para prover uma resposta adequada. Elas são configuradas através de exemplos de frases fornecidas pelo desenvolvedor, que possam ser utilizadas para identificar um determinado propósito ou ideia. A Figura 8 mostra uma intenção, identificada pelo caractere especial “#” e nomeada “pedir\_pizza”, juntamente com três exemplos de frases que definem a intenção de um usuário em realizar um pedido de pizza.

---

<sup>7</sup> <https://www.ibm.com/watson/services/conversation/>. Acessado em 28 de outubro de 2017.

Figura 8 – Configuração da Intenção #pedir\_pizza na API Conversation.

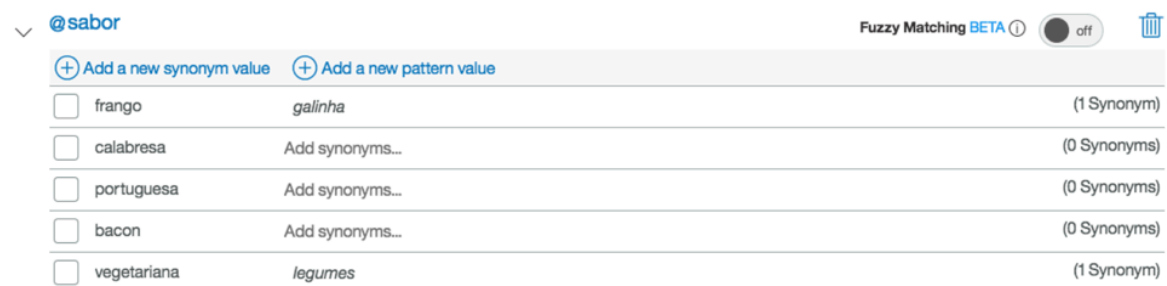


Fonte: Elaborada pelo autor.

Esses exemplos de frases são analisados pela ferramenta que, através da aplicação de técnicas de Processamento de Linguagem Natural, gera automaticamente uma variedade de possíveis entradas baseadas nos exemplos originais que quando detectadas, também possam ser relacionadas a intenção criada. Por exemplo, se o usuário digitasse a entrada: “*gostaria de fazer o pedido de uma pizza*”, o sistema a relacionaria com sucesso a intenção “#pedir\_pizza”.

As *Entities*, ou entidades, representam os possíveis núcleos ou ações da intenção desejada pelo usuário e se destinam a identificar propósitos específicos. Fundamentalmente, elas são representadas através de palavras-chaves e sinônimos que irão compor a significância da entidade. Na Figura 9 está representada uma entidade, identificada pelo caractere especial “@” e nomeada “sabor”.

Figura 9 – Configuração da Entidade “@sabor” na API Conversation.



Fonte: Elaborada pelo autor.

Através dessa configuração, a plataforma é capaz de identificar na entrada do usuário a entidade referenciada e utilizar essa informação como guia para fluxo de diálogo. Em nosso exemplo, o *chatbot* já é capaz de detectar a intenção do usuário em realizar um pedido de pizza. Mas e quanto ao tamanho e sabores? Com o uso da entidade “@sabor”, ao serem detectadas as palavras “frango”, “calabresa”, “portuguesa”, “bacon”, “vegetariana”, ou qualquer um de seus sinônimos, o sistema será capaz de compreender que o usuário está se

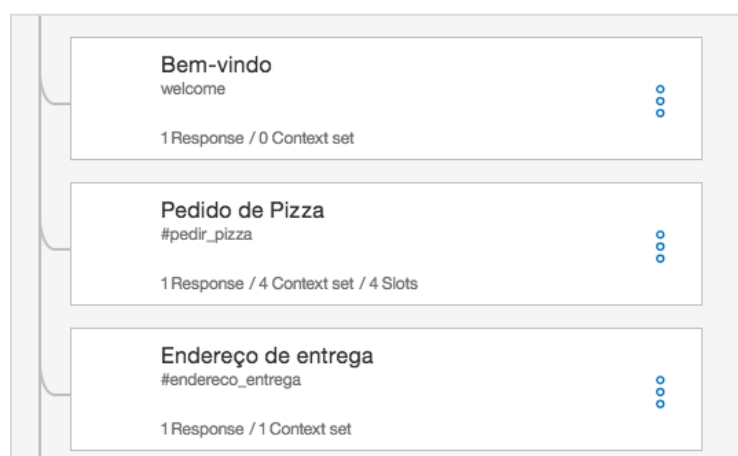
referindo a um tipo de sabor e, a partir deste conhecimento, guiar a conversação para o seu próximo passo.

Ainda na criação das Entidades, o *Watson* disponibiliza a ferramenta *Fuzzy Matching*, que proporciona uma verificação gramatical do texto inserido pelo usuário para comparação com as palavras-chave correspondentes às entidades, fazendo com que o desenvolvedor não precise se preocupar em adicionar diferentes tipos de erros de digitação como sinônimos na descrição da mesma.

Por fim, o conceito *Dialog*, ou diálogo, permite que sejam criadas as regras que irão definir o fluxo da conversação. Esse processo é realizado pela criação de nós que obedecem a condições específicas definidas pelo desenvolvedor e que por fim, repassam o fluxo da conversa para o próximo nó adequado. Eles podem ser divididos em sub-níveis onde cada nó filho processa mais a fundo a entrada detectada pelo nó pai.

Na Figura 10 é mostrada a organização de nós utilizada no exemplo de *chatbot* para atendimento de pedidos em uma pizzaria. O nó “Bem-Vindo” armazena possíveis respostas a serem dadas quando o usuário aciona o *chatbot*, ação definida pela condição denominada “*welcome*”. O nó “Pedido de Pizza” armazena as regras a serem seguidas quando a intenção “*#pedir\_pizza*” é detectada pelo sistema. O nó “Endereço de entrega” define as ações a serem tomadas ao ser detectada a intenção “*#endereco\_entrega*”, que por sua vez, é responsável por identificar a ideia de que o usuário está informando seu endereço.

Figura 10 – Exemplificação da organização de nós de diálogo na API Conversation



Fonte: Elaborada pelo autor.

### 3.3.2 Configuração dos Nós de Diálogo

A configuração dos nós de diálogo é que definirão o rumo da conversação. Essas configurações são implementadas pela plataforma *Watson* através de uma série de condições



de entrada, a partir das quais diferentes ações podem ser tomadas. Essa análise leva em conta a detecção, ou não detecção, da Intenção do usuário, possíveis Entidades presentes no texto, ou ainda outras condições definidas pelo desenvolvedor (Figura 11).

Figura 11 – Janela de configuração do nó de diálogo “Pedido de Pizza”.

**Pedido de Pizza** Customize ✕

If bot recognizes:  
#pedir\_pizza ⊖ ⊕

Then check for: Manage handlers

	Check for	Save it as	If not present, ask	Type		
1	@tamanho	\$tamanho	Temos pizzas tamanh	Required	⚙️	🗑️
2	@sabor	\$sabor1	Você pode pedir 3 sab	Required	⚙️	🗑️
3	@sabor	\$sabor2	Qual é o segundo sab	Required	⚙️	🗑️
4	@sabor	\$sabor3	E o terceiro sabor?	Required	⚙️	🗑️

⊕ Add slot

Then respond with: ⋮

1. Qual é o nome da rua para entrega? ⊖

Add a variation to this response

And finally

Wait for user input ▾

Fonte: Elaborada pelo autor.

Dentro do nó de diálogo “Pedido de Pizza”, primeiramente é definida a condição de entrada. Em nosso exemplo, essa condição é a detecção no texto do usuário da intenção “#pedir\_pizza”, exemplificada na Figura 8. Neste ponto, é utilizado outro recurso da ferramenta, chamado *Slots*, que permite a coleta de múltiplas informações em um único nó de diálogo implementando condições de entrada para cada um dos campos criados. Em nosso exemplo, essas informações serão o tamanho da pizza e os três sabores a serem escolhidos.

Após a detecção da intenção, o sistema passa a procurar pelos campos definidos nos *slots*, neste caso, a entidade “@tamanho”. Ao identificar a entidade esperada, seu valor é armazenado em uma variável de contexto para posterior utilização, mas caso ela não for detectada, o sistema permite a configuração de uma mensagem de retorno para que se possa guiar o usuário a fornecer a informação necessária. Esse processo é repetido sequencialmente para cada *slot* configurado, definindo o fluxo da conversa.

Na intenção de enriquecer o diálogo ou até mesmo preparar o *chatbot* para uma resposta inesperada do usuário, cada etapa deste fluxo pode ser configurada individualmente para responder com uma determinada ação conforme o que for detectado na entrada do usuário para um determinado *slot*, formando um novo nível de análise (Figura 12).

Figura 12 – Janela de configuração de Slot disponibilizada pela API Conversation.

### Configure slot 1 ⋮

---

**Check for:**  
@tamanho

**Save it as:**  
\$tamanho

---

**If \$tamanho is not present then ask:** Slot is required ⓘ

Temos pizzas tamanho família, grande, média ou broto. De qual tamanho você gostaria?

---

**When user responds, if @tamanho is...**

**Found:**

	If bot recognizes	Respond with	
1	@tamanho:família	Sua fome não vai resistir a nossa pi:	⚙️ 🗑️

[+ Add a response](#)

---

**Not found:**

	If bot recognizes	Respond with	
1	Enter an intent, entity or context vari	Enter a response...	⚙️ 🗑️

[+ Add a response](#)

Cancel
Save

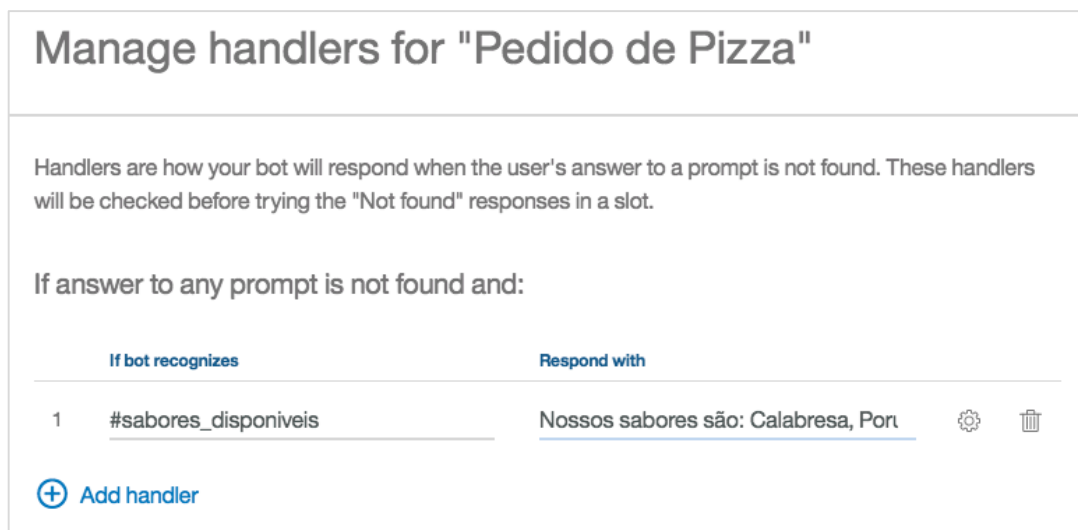
Fonte: Elaborada pelo autor.

Com essa análise, o *chatbot* pode adicionar informações e responder perguntas do usuário que fujam do contexto atual do fluxo de diálogo e imediatamente retomá-lo após este processo. Em nosso exemplo, se for detectado o pedido do usuário pelo tamanho família, o *chatbot* responderá com uma frase específica para essa situação e então continuará a execução da checagem definida pelo *slot2*.

A ferramenta ainda disponibiliza o gerenciamento das entradas do usuário de maneira global dentro do contexto criado pelo nó. Os *handlers*, podem ser configurados com condições que serão verificadas quando uma entrada do usuário não corresponder as condições esperadas por nenhum dos *slots* do nó atual, retomando o fluxo da conversação logo após o seu processamento.

A Figura 13 exemplifica um *handler* criado para o nó “Pedido de Pizza”. Com esta configuração, se o usuário demonstrar a intenção “#sabores\_disponiveis”, quando na verdade o *chatbot* está aguardando uma entrada correspondente a uma das entidades configuradas nos *slots*, ele saberá que deve retornar ao usuário a lista de sabores disponível para o pedido. Isso permite que a aplicação consiga trabalhar com trocas de contexto sem quebrar o fluxo de diálogo do nó atual, permitindo que o fluxo seja retomado após o envio da resposta adequada.

Figura 13 – Janela de configuração dos *handlers* do nó “Pedido de Pizza”.



Fonte: Elaborada pelo autor.

Em nosso exemplo, após exaurir essas checagens, o nó apresenta um campo para configuração da resposta final dada ao usuário e a ação que deverá ser feita no momento que o fluxo de diálogo sair deste nó. Entretanto, assim como os *slots*, a plataforma ainda disponibiliza outras modificações customizáveis nesta estrutura geral de análise do nó, que podem ser usadas conforme a necessidade do desenvolvedor.

No fim do processamento do nó “Pedido de Pizza”, o *chatbot* terá armazenado em suas variáveis de contexto toda a informação referente ao pedido realizado pelo cliente. A resposta dada ao final deste nó guia o usuário para que informe seu endereço. Nesta etapa o *chatbot* irá detectar a intenção “#endereco\_entrega”, configurada previamente, reconhecendo que o texto informado pelo usuário diz respeito a seu endereço. Essa informação é capturada em uma variável de contexto, completando a lista de informações necessária e concluindo o fluxo de diálogo para a realização do pedido do usuário, como demonstrado na Figura 14.

Figura 14 – Fluxo de diálogo final, demonstrando as intenções e entidades detectadas, variáveis de contexto e as respostas fornecidas pelo sistema Watson.

The figure displays two panels of a chatbot interface. The left panel shows the initial interaction: a greeting, a selection of pizza size (family), and the first flavor (calabresa). The right panel shows the final steps: selecting the second flavor (vegetarian), the third flavor (bacon), providing the delivery address (Rua Floriano Peixoto, 1234), and saying thank you. Each step shows the detected intent and entity, and the corresponding context variable.

**Left Panel:**

- Olá. Como posso te ajudar?
- Gostaria de pedir uma pizza
- #pedir\_pizza
- Temos pizzas tamanho família, grande, média ou broto. De qual tamanho você gostaria?
- tamanho família, por favor
- Irrelevant
- @tamanho:familia
- Sua fome não vai resistir a nossa pizza tamanho família!
- Você pode pedir 3 sabores. Qual é o primeiro sabor?
- calabresa
- Irrelevant
- @sabor:calabresa
- Qual é o segundo sabor?
- não tenho certeza...
- Irrelevant
- Quem sabe uma vegetariana? É importante se manter saudável :)

**Right Panel:**

- pode ser a vegetariana
- Irrelevant
- @sabor:vegetariana
- E o terceiro sabor?
- bacon
- Irrelevant
- @sabor:bacon
- Qual é o nome da rua para entrega?
- rua floriano peixoto, 1234
- #endereco\_entrega
- Ok! Seu pedido é uma pizza tamanho família nos sabores: calabresa, vegetariana e bacon, para ser entregue no endereço: rua floriano peixoto, 1234.
- obrigado!
- #retorno\_positivo
- Eu que agradeço!

Enter something to test your bot  
Use the up key for most recent

Fonte: Elaborada pelo autor.

### 3.4 Outras plataformas de desenvolvimento

Na medida em que a aplicação dos *chatbots* no relacionamento com os usuários se populariza, crescem o número de plataformas e ferramentas de desenvolvimento que visam simplificar a criação deste tipo de agente de conversação. A Tabela 1 apresenta um levantamento de alguma dessas ferramentas e suas características básicas.

Tabela 1 – Relação de plataformas de desenvolvimento de *chatbots*

Plataformas	Características Básicas	Licença	Acesso
Twyla	Integrado com diferenciadas plataformas <i>Helpdesk</i> e Live Chats.	Planos mensais	<a href="https://www.twylahelps.com/">https://www.twylahelps.com/</a>
Wit.ai	Permite a definição de entidades, intenções, contextos e ações através do uso de PLN.	Gratuito	<a href="https://wit.ai/">https://wit.ai/</a>
Microsoft Bot	Compreende intenções do usuário e permite incorporações de outros serviços, como PLN, voz e mecanismo de pesquisa.	<i>Open Source</i>	<a href="https://docs.botframework.com/en-us/">https://docs.botframework.com/en-us/</a>
ManyChat	Possui funcionalidades básicas como envio de conteúdo, agendamento de postagens, respostas automáticas e outros.	Gratuito	<a href="https://manychat.com/">https://manychat.com/</a>
rasa NLU	Permite a criação e detecção de intenções e entidades nas entradas dos usuários.	<i>Open Source</i>	<a href="https://rasa.ai/">https://rasa.ai/</a>
ChatFuel	Organiza a construção de diálogos em blocos que podem ser associados para geração de mensagens.	Gratuito	<a href="https://chatfuel.com/">https://chatfuel.com/</a>
Pandorabots	Permite a integração de serviços de IA e PLN à uma aplicação própria além da utilização de recursos AIML.	Diferentes planos	<a href="https://www.pandorabots.com/">https://www.pandorabots.com/</a>

Fonte: <https://chatbotsjournal.com/25-chatbot-platforms-a-comparative-table-aeefc932eaff>

## 4 O USO DE *CHATBOTS* NO COMÉRCIO ELETRÔNICO

O crescimento da internet e da utilização dos *smartphones* transformaram a maneira como as pessoas interagem entre si e também com empresas, estimulando a expansão do comércio eletrônico e promovendo múltiplas oportunidades para que estas empresas expandam suas atividades comerciais e criem novas formas de interações com consumidores que não eram possíveis anteriormente (EEUWEN, 2017).

O mesmo crescimento pode ser observado no número de usuários de aplicativos de mensagem, que rapidamente vem ultrapassando o número mensal de usuários de redes sociais (EEUWEN, 2017). Em resposta a esse movimento, empresas de diversas áreas passaram a investir na disponibilização de serviços através destas ferramentas de conversação, fazendo-se valer da familiaridade de seus consumidores com este tipo de comunicação baseado na troca de mensagens de texto.

Embora os *chatbots* já sejam aplicados em diversas áreas, sua utilização nestes ambientes de troca de mensagens com o objetivo de auxiliar o atendimento ao consumidor é relativamente recente. O uso deste tipo de tecnologia com objetivos comerciais faz parte de uma área de desenvolvimento chamada de comércio conversacional, a qual busca oportunizar ao consumidor conveniência, personalização e assistência em processos de tomada de decisão através da integração de diferentes elementos e ações de *marketing* ao funcionamento dos *chatbots*. (EEUWEN, 2017).

Não é difícil de entender como a utilização desses agentes de conversação, dentro desse contexto, pode ser atrativo para as empresas. Além de ajudar com o problema da demanda dos usuários por uma interação rápida e na redução de custos, sua aplicação pode estender-se a recomendações de produtos, atualizações de serviço, personalização de atendimento, chamadas a ações onde o cliente pode visualizar produtos em oferta diretamente na interface do chat, além de muitas outras soluções que podem acelerar e simplificar o processo de compra. Indo além, *chatbots* com elementos mais complexos podem até mesmo reconhecer as intenções dos clientes, filtrando a oferta de produtos e serviços de maneira personalizada com base nas escolhas e preferencias prévias do consumidor (CONSTINE, 2016).

Várias aplicações para os mecanismos de troca de mensagens já estão sendo propostos e até mesmo explorados. Sua utilização para pagamentos *online* poderá torná-los uma ferramenta de comércio unificada, permitindo uma experiência completa de compra sem que o usuário saia da interface da aplicação (SHOPIFY, 2017). Como exemplo, podemos citar a

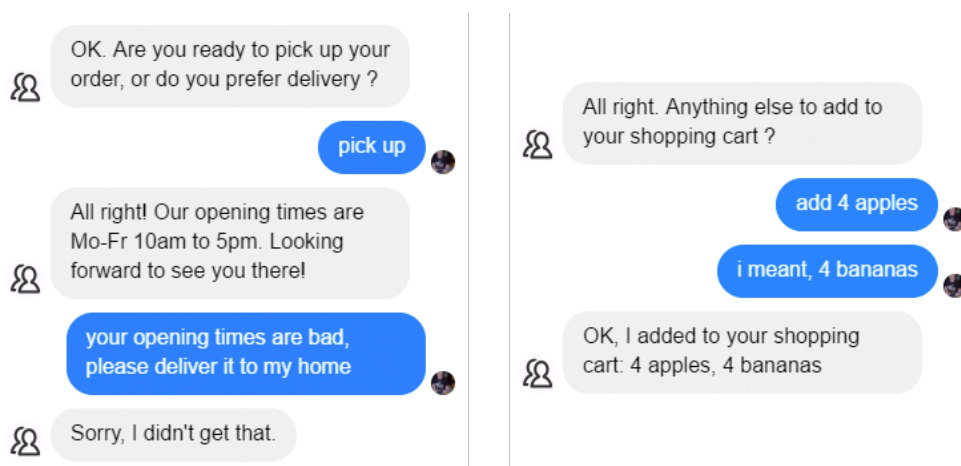
ferramenta *WeChat*<sup>8</sup>, um popular serviço de mensagens que já integra uma variada gama de serviços, como transferências monetárias, encomenda de refeições, compra de ingressos de cinema e passagens aéreas, além de outras.

A crescente aplicação desta tecnologia, juntamente com o apoio de nomes de grande influência na indústria como Chris Messina<sup>9</sup>, Satya Nadella<sup>10</sup> e Mark Zuckerberg<sup>11</sup> levam muitos a considerar que ela possa causar um grande impacto na forma como nos relacionamos, não somente com diferentes dispositivos e *softwares*, mas também com empresas e marcas.

#### 4.1 Desafios da aplicação

A maior barreira a ser superada por esse tipo de tecnologia quando aplicada ao contexto do comércio eletrônico, é a conquista da confiança de seus usuários. *Chatbots* orientados a conversação fazem uso de tecnologias baseadas em IA como o PLN, para analisar as entradas do usuário para formular respostas que possam convence-lo de que está interagindo com outro ser humano. Ao atuar de forma transparente, mesmo pequenos erros de interpretação ou falhas de contextualização por parte da ferramenta podem levar a desvios de diálogo e consequentes quebras do fluxo conversacional (BECKENKAMP, 2017). Esse tipo de problema pode vir a ocasionar estranheza, perda da confiança e frustração do cliente, que pode tender ao abandono do processo de compra (Figura 15).

Figura 15 – Exemplo de erros de interpretação em um *chatbot* orientado a conversação.



Fonte: <https://chatbotsmagazine.com/3-things-your-chatbot-fails-at-but-shouldnt-1be87e81f6f7>

<sup>8</sup> <https://www.wechat.com/pt/>. Acessado em 12 de outubro de 2017.

<sup>9</sup> <https://goo.gl/EG10bN>. Acessado em 30 de outubro de 2017.

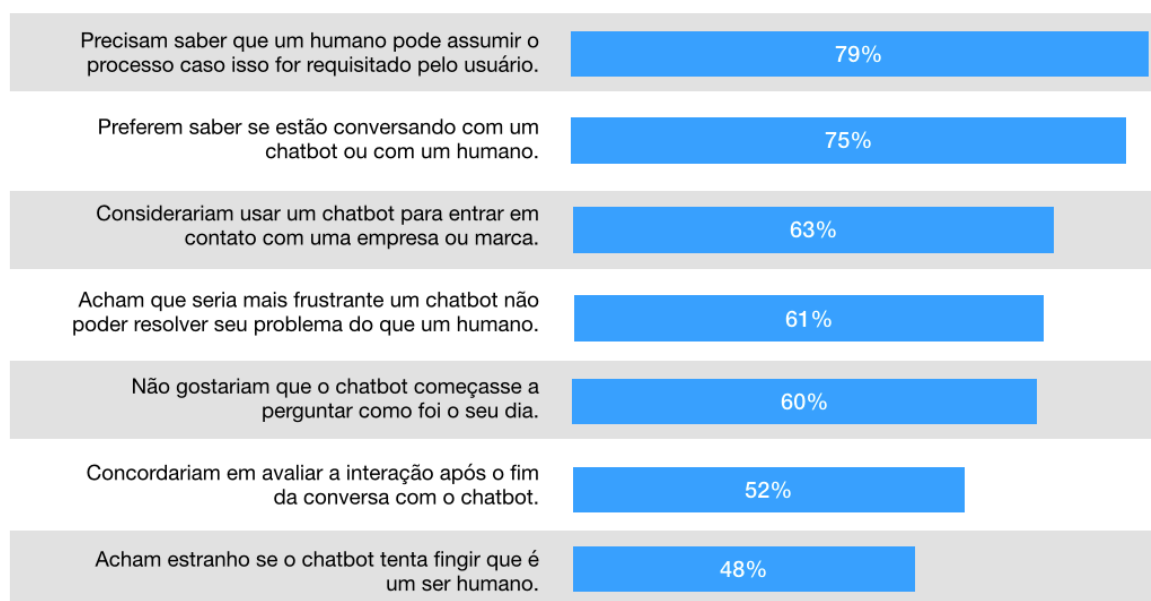
<sup>10</sup> <http://www.businessinsider.com.au/microsoft-to-announce-chatbots-2016-3>. Acessado em 30 de outubro de 2017.

<sup>11</sup> <https://www.wsj.com/articles/facebook-hopes-chatbots-can-solve-app-overload-1460930220>. Acessado em 30 de outubro de 2017.

Quando a interação com um *chatbot* tem como desfecho um investimento monetário, como é o caso do comércio eletrônico, a confiança do usuário na ferramenta e a transparência do processo tendem a ser mais relevantes do que o sucesso do software em se passar por um ser humano. A objetividade e simplicidade da troca de informações pode tornar-se fundamental para o sucesso da ferramenta em cumprir seu objetivo dentro desse segmento de mercado. Questões como, não tentar persuadir o usuário que ele está se comunicando com um ser humano e manter a interação simples e direta sem se desviar do objetivo proposto, podem resultar em uma experiência mais satisfatória.

Estas observações foram feitas por uma pesquisa realizada pelo site *Mindshare*<sup>12</sup> e publicada pelo site *Business Insider UK*<sup>13</sup> em entrevista com 1000 usuários de smartphones em relação a suas atitudes em relação aos *chatbots* quando utilizados por instituições financeiras (Figura 16).

Figura 16 – Atitude de usuários de *smartphones* em relação ao uso de *chatbots*.



Fonte: Mindshare, adaptado de: <http://uk.businessinsider.com/savings-chatbot-digit-debuts-on-messenger-2017-1>

A utilização de *chatbots* orientados a regras pode oferecer essa simplificação na forma de interação fazendo uso de elementos como botões e menus como forma de receber as entradas do usuário, não se limitando apenas a entradas de texto que podem depender de diferentes tecnologias para serem corretamente interpretadas. Este tipo de agente de

<sup>12</sup> <https://www.mindshareworld.com/>

<sup>13</sup> <http://uk.businessinsider.com/>



conversa o apresenta menores desafios de desenvolvimento quando comparados a *chatbots* orientados a conversa o e pode apresentar grandes vantagens quando aplicados no ambiente do com rcio eletr nico, principalmente se considerarmos os problemas de estabilidade, escalabilidade e flexibilidade originados pelo alto n vel de complexidade da linguagem humana (RAHMAN et al, 2017).

## 5 CONSTRUÇÃO DE UM *CHATBOT* PARA E-COMMERCE NA REDE SOCIAL *FACEBOOK*

A etapa final deste trabalho consistiu na aplicação dos conhecimentos adquiridos e ferramentas analisadas para o desenvolvimento de um *chatbot* orientado a regras, capaz de utilizar elementos gráficos para guiar fluxo da interação e o PLN para análise de texto.

A construção da lógica de funcionamento do *chatbot*, seus elementos e fluxo de interação, foram projetados de forma a serem gerenciados através de um *software* referenciado como Painel Administrativo implementado com a utilização dos *frameworks* de desenvolvimento *Laravel*<sup>14</sup> e *Vue.js*<sup>15</sup> em parceria com a empresa *Moonbase Digital*<sup>16</sup>, com o objetivo de produzir um *software* que possibilitasse a construção de diferentes *chatbots* em uma interface simples e unificada, manipulando dados através de métodos genéricos capazes de extrair suas informações de um banco de dados próprio.

Considerando esse cenário, a utilização da plataforma *Messenger*<sup>17</sup> para desenvolvimento de *chatbots* na rede social *Facebook* se mostrou a melhor opção, pois permite a utilização de variados elementos estruturais, além de possuir um sistema de PLN integrado, o *Wit.ai*<sup>18</sup>. O acesso a esses serviços foi obtido através de uma API de comunicação que atua como uma interface para troca de informações entre o Painel Administrativo e a plataforma *Messenger*.

Na Figura 17 é apresentada a arquitetura geral da solução desenvolvida. Nela, uma entrada do usuário é recebida pela plataforma *Messenger* que, por sua vez, pode fazer uso dos recursos de PLN do sistema *Wit.ai* para análise das entradas do usuário. Uma requisição contendo essas informações é encaminhada para a API de comunicação que as torna disponíveis para o Painel Administrativo, no qual as regras e a base de conhecimento do *chatbot* podem ser criadas e a partir das quais são selecionadas as respostas a serem fornecidas ao usuário. Essas informações são recuperadas do banco de dados da aplicação e então repassadas para a API de comunicação para construção e envio da resposta nos padrões exigidos pela plataforma *Messenger*, que se responsabiliza pela construção dos elementos visuais e exibição das informações para o usuário. Nas próximas seções deste capítulo cada um dos componentes dessa arquitetura é detalhado.

---

<sup>14</sup> <https://laravel.com/>

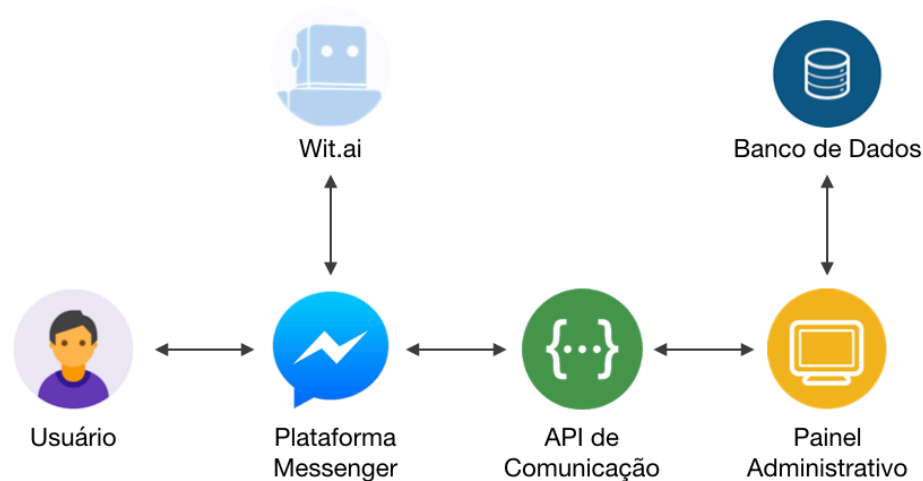
<sup>15</sup> <https://vuejs.org/>

<sup>16</sup> <http://www.moonbase.digital>

<sup>17</sup> <https://developers.facebook.com/docs/messenger-platform/>

<sup>18</sup> <https://wit.ai/>

Figura 17 – Organização das tecnologias utilizadas e fluxo da troca de informações.



Fonte: Elaborada pelo autor.

### 5.1 Configuração e utilização da plataforma *Messenger* e do sistema *Wit.ai*

A plataforma *Messenger* é uma ferramenta formada por um conjunto de APIs REST (*Representational State Transfer*) disponibilizada pela rede social *Facebook* para o desenvolvimento de *chatbots* em seu aplicativo de mensagens instantâneas, o *Messenger*<sup>19</sup>. A comunicação com a plataforma é realizada através de requisições autenticadas por um *token* de acesso individual, as quais possuem seu corpo formado por uma estrutura em formato JSON (*JavaScript Object Notation*), exemplificada na Figura 18, contendo as propriedades exigidas pela ferramenta para a manipulação de mensagens e transferência das informações. A partir dessas informações a plataforma *Messenger* pode oferecer uma série de personalizações, como modelos de mensagens, menus, tipos de conteúdos, entre outras (FACEBOOK, 2018).

Figura 18 – Estrutura básica de uma requisição JSON para o envio de texto para a plataforma *Messenger*.

```

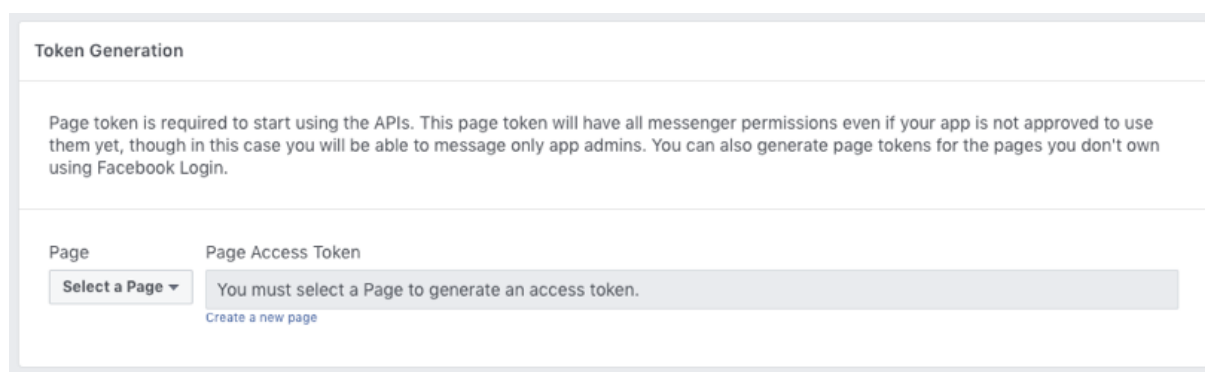
{
  "recipient": {
    "id": "<PSID>"
  },
  "message": {
    "text": "hello, world!"
  }
}
  
```

Fonte: <https://developers.facebook.com/docs/messenger-platform/send-messages>

<sup>19</sup> <https://www.messenger.com>

A habilitação da plataforma *Messenger* é realizada através da área para desenvolvedores da rede social, o *Facebook for Developers*<sup>20</sup>, onde a primeira etapa de configuração é a seleção de uma página da rede social onde esse serviço será utilizado, para a qual será gerada uma chave de acesso única, utilizada para validação da troca de mensagens entre a plataforma *Messenger* e uma aplicação externa (Figura 19).

Figura 19 – Configuração da plataforma *Messenger* e definição da chave de acesso para a página desejada.



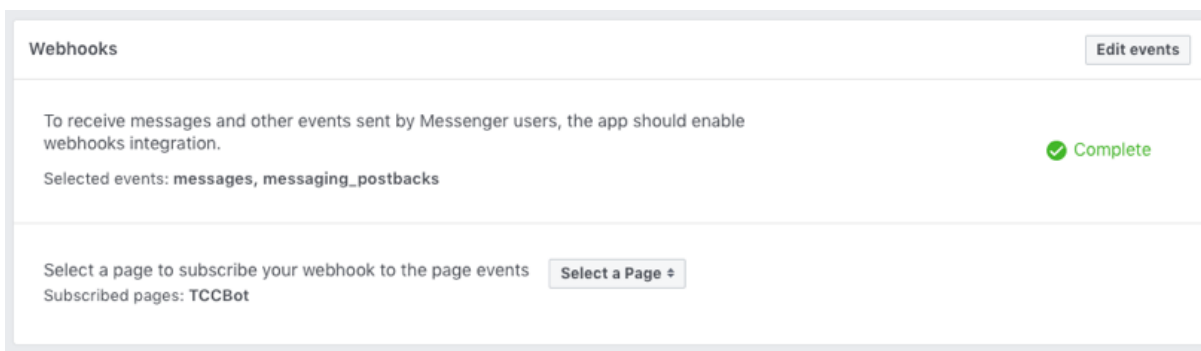
Fonte: <https://developers.facebook.com>

A segunda etapa é a configuração de um *webhook* que aplica as políticas do serviço quanto a forma de autenticação, recebimento e envio das informações. Nele é especificada a URL (*Uniform Resource Locator*) com a qual será realizada a comunicação, que deve corresponder ao endereço do servidor onde a API de comunicação está hospedada contendo os métodos de autenticação exigidos pela plataforma. Ainda é necessário que sejam definidos os tipos de eventos que terão permissão para serem utilizados. Neste trabalho foram utilizados o evento “*messages*”, que gerencia o recebimento de mensagens contendo texto simples ou anexos como imagens, vídeos, áudios ou arquivos, e o evento “*messages\_postbacks*”, responsável pelo gerenciamento de ações originadas pelo acionamento de botões ou itens de menus que possam vir a ser utilizados nos diferentes modelos de mensagem exibidos na janela do aplicativo de troca de mensagem. A figura 20 mostra a configuração dos *webhooks* para a plataforma *Messenger*.

---

<sup>20</sup> <https://developers.facebook.com>

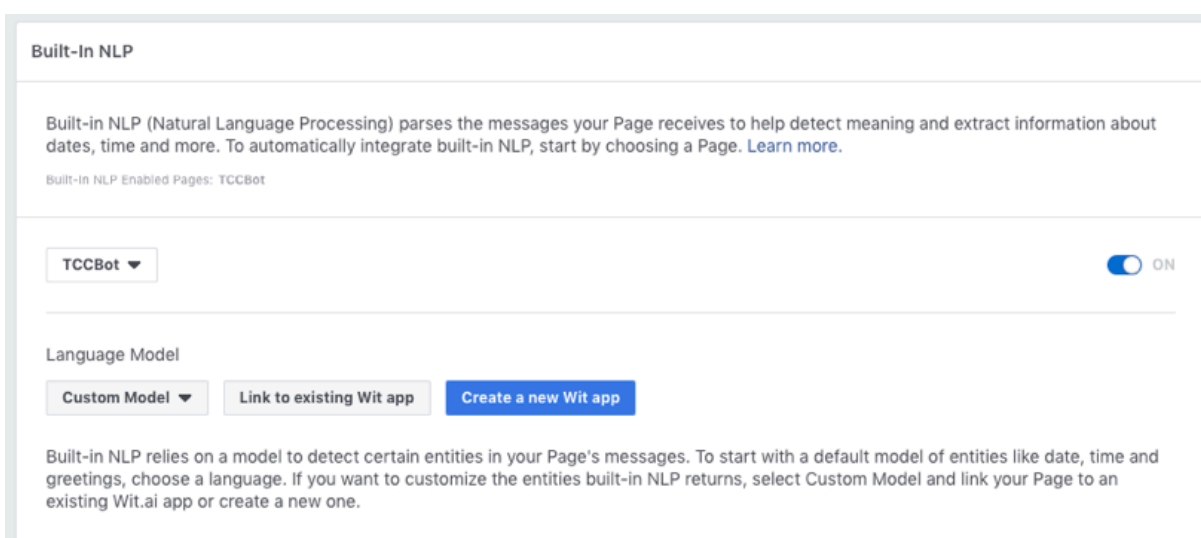
Figura 20 – Configuração da plataforma *Messenger* e definições obrigatórias dos *webhooks*.



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

Uma terceira configuração possível de ser realizada é a habilitação do item *Build-In NLP* (Figura 21). Embora não seja obrigatória para o correto funcionamento do serviço, ela é a responsável pela incorporação do PLN oferecido pelo sistema *Wit.ai*. Nesse item é possível selecionar a página da rede social para qual será ativado o serviço, assim como o modelo de processamento de linguagem a ser utilizado. O serviço recursos básicos do PLN para uma série de diferentes idiomas, e também permite a seleção de um modelo customizado, onde o desenvolvedor pode utilizar a interface do sistema *Wit.ai* para criar as próprias regras de processamento.

Figura 21 – Interface para habilitação da integração entre a Plataforma Messenger e o sistema *Wit.ai*.



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

Funcionando de maneira similar ao sistema *Watson* e sua *API Conversation*, descritas no item 3.3 deste trabalho, o *Wit.ai* faz uso de técnicas de PLN para interpretar entradas do

usuário através da análise das *Entities*, ou entidades, formadas pela detecção de palavras chave específicas presentes na entrada do usuário e das *Trait Entities*, formadas a partir do reconhecimento de sentenças completas através de comparações de similaridade com exemplos fornecidos pelo desenvolvedor, buscando com isso a caracterização da intenção implícita na entrada fornecida pelo usuário (Figura 22).

Figura 22 – Interface do sistema *Wit.ai* listando exemplos de frases fornecidas pelo desenvolvedor e a detecção da *Trait Entity* “saudação”.

**Samples**

Filter by:

Search through your samples.

Text

“ bom dia, como vai você? ” x

bom dia bot, tudo bem?

+ Add a new entity

📊 🗑️

“ boa tarde bot, tudo certo? ” x

“ bom dia bot, como vai você? ” x

Fonte: Elaborada pelo autor.

Juntamente com a análise de entidades e intenções, a plataforma ainda é capaz de fornecer um campo numérico intitulado “*confidence*”, que representa o nível de confiança de sistema na interpretação realizada. Esse valor pode variar entre 0 e 1, sendo que quanto mais próximo de 1, maior é a confiança da plataforma na interpretação fornecida. O cálculo é obtido através de uma série de comparações realizadas pela plataforma, tanto com diferentes exemplos fornecidos pelo desenvolvedor, quanto com a base de conhecimento da ferramenta, sendo que quanto mais exemplos e entidades forem configuradas pelo desenvolvedor, mais credibilidade poderá ser atribuída ao nível de confiança detectado. Os resultados encontrados pela análise realizada no sistema *Wit.ai* são incorporados ao corpo da requisição JSON enviada pela plataforma *Messenger* para a API de comunicação em um campo intitulado “*nlp*”, conforme exemplifica a Figura 23.

Figura 23 – Conteúdo do campo “*nlp*” gerado pelo sistema *Wit.ai* identificando as entidades “saudação” (A) e “despedida” (B).

```

A) {
  "entities":{
    "intent":[
      {
        "confidence":0.99760695671936,
        "value":"saudacao"
      }
    ]
  },
  "msg_id":"1V3WaZYAcNqDDMHqP"
}

B) {
  "entities":{
    "intent":[
      {
        "confidence":1,
        "value":"despedida"
      }
    ]
  },
  "msg_id":"FF3WaZYAcNqDDMHqP"
}

```

Fonte: Elaborada pelo autor.

## 5.2 Desenvolvimento da API de comunicação

A API de comunicação, desenvolvida na linguagem de programação PHP, tem como propósito conter as classes e métodos responsáveis tanto pela validação das requisições quanto pelo processamento e construção de suas respostas, seguindo o formato e regras estabelecidas pela plataforma *Messenger* em sua documentação, possibilitando assim a troca de informações entre a plataforma e o Painel Administrativo desenvolvido.

A Figura 24A, mostra o exemplo de uma requisição recebida pela API de comunicação através do acionamento de um botão na janela de conversação do *chat*, identificado como alvo da ação o *Postback* de nome “*macbook-pro*”, definido no campo “*payload*”. Ao receber essa informação, o Painel Administrativo recupera os dados relacionados a esse nó de *Postback* de seu banco de dados para a construção da resposta (Figura 24B), especificando no campo “*type*” a utilização do modelo de mensagem “*generic*”, assim como a lista de produtos correspondentes a opção selecionada. Para fins demonstrativos, foram utilizados produtos de uma marca de produtos eletrônicos disponíveis no website da empresa. Alguns campos foram abreviados ou omitido para melhor visualização.

O uso dessa forma de organização, separando os métodos de comunicação do restante do código do Painel Administrativo, permite que futuras atualizações da plataforma *Messenger* possam ser incorporadas ao projeto mais facilmente, sem exigir alterações significativas no código interno do Painel Administrativo.

Figura 24 – Exemplo de uma requisição recebida pela API de comunicação (A) e da resposta gerada (B).

```

A) {
  "object": "page",
  "entry": [
    {
      "id": "587869614980640",
      "time": 1542226857052,
      "messaging": [
        {
          "recipient": {
            "id": "587869614980640"
          },
          "timestamp": 1542226857052,
          "sender": {
            "id": "1925108800913910"
          },
          "postback": {
            "payload": "macbook-pro",
            "title": "Macbook Pro"
          }
        }
      ]
    }
  ]
}

B) {
  "id": 371,
  "type": "generic",
  "message": "Escolha a sua configuração preferida...",
  "template": 1,
  "postback_id": 141,
  "created_at": "2018-11-12 11:23:11",
  "updated_at": "2018-11-12 11:23:11",
  "products": [
    {
      "id": 241,
      "title": "Macbook Pro 13",
      "subtitle": "13 polegadas",
      "image_url": "12fa670834dac859.jpeg",
      "url": "https://www.apple.com/macbook-pro/",
      "created_at": "2018-11-12 10:52:09",
      "updated_at": "2018-11-12 11:30:38",
      "image_full_url": "https://s3-sa-east-1...59.jpeg",
      "pivot": {
        "message_id": 371,
        "product_id": 241
      }
    }
  ],
  { },
  { }
}

```

Fonte: Elaborada pelo autor.

### 5.3 Desenvolvimento do Painel Administrativo

A construção do Painel Administrativo da aplicação levou em consideração o objetivo de criar um software de configuração de *chatbots*, com recursos semelhantes ao sistema *Watson* descrito no item 3.3, capaz de gerenciar o fluxo de interação com o usuário por meio de regras de relacionamento entre seus elementos, assim como através de texto livre.

Para isso, o projeto foi arquitetado em torno da criação, gerenciamento e relacionamento entre conjuntos de eventos chamados *Postbacks*, responsáveis por conter as mensagens a serem enviadas ao usuário perante uma determinada ação. Para cada mensagem é possível definir um modelo e seus possíveis elementos, podendo esses serem *links* para endereços externos ou chamadas para outros *Postbacks*.

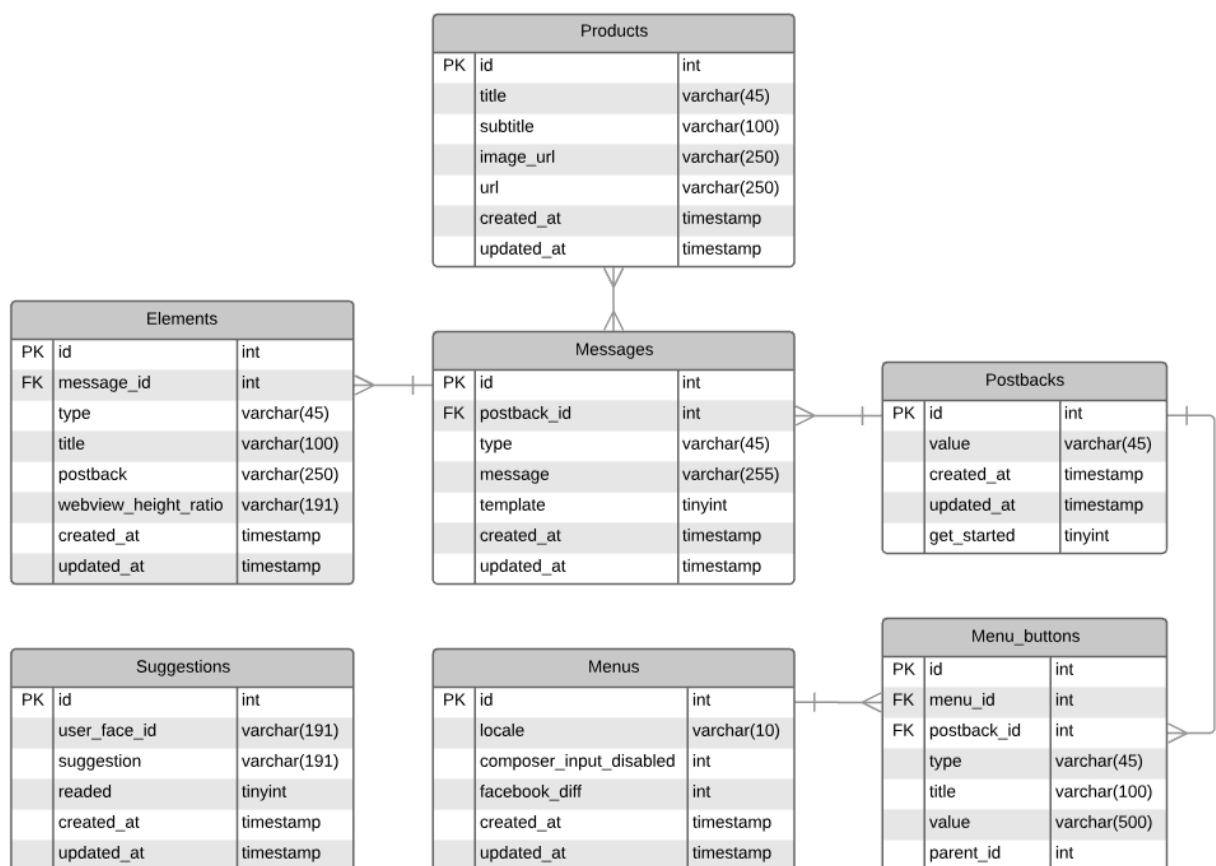
O Painel Administrativo ainda conta com o gerenciamento de menus, mais uma funcionalidade fornecida pela plataforma *Messenger*, que podem ou não serem exibidos ao lado do campo de entrada de texto na janela do aplicativo de troca de mensagens da rede



social. Esse e os demais componentes do Painel Administrativo serão descritos em detalhes nas próximas seções desse trabalho.

Ainda, para atender aos propósitos deste trabalho, também foi incluído no Painel Administrativo o gerenciamento do cadastro de produtos, embora a obtenção desse tipo de dados possa ser realizada diretamente em uma base de dados externa. A lógica organizacional, da aplicação como um todo, pode ser melhor visualizada através do modelo lógico do banco de dados da aplicação (Figura 25).

Figura 25 – Modelo do banco de dados do Painel Administrativo do projeto.



Fonte: Elaborada pelo autor.

### 5.3.1 Gerenciamento de *Postbacks*

Os *Postback* são utilizados no Painel Administrativo para encapsular uma ou mais mensagens que serão enviadas para o usuário de forma a facilitar a criação da lógica envolvida no funcionamento do *chatbot*. Quando uma requisição recebida especifica um determinado *Postback* cadastrado no banco de dados da aplicação, as mensagens nele relacionadas são enviadas sequencialmente pela API de comunicação.

Mensagens anexadas a um *Postback* podem conter anexos como textos, imagens, áudio, vídeo, arquivos ou ainda modelos de mensagem, como botões, *sliders* e listas. A Figura 26 traz como exemplo o *Postback* “iniciar” e suas mensagens. A primeira sendo uma mensagem de texto simples e a segunda, utilizando um modelo de botões que quando acionados na aplicação de troca de mensagens realizam uma chamada aos seus respectivos elementos, neste caso, os *Postbacks* de nomes “*iphone-modelos*”, “*watch-modelos*” e “*comp-modelos*”. O resultado de como essas mensagens são exibidas na tela do aplicativo de troca de mensagens pode ser vista na Figura 27A.

Figura 26 – Interface do Painel Administrativo exibindo a configuração do *Postback* “iniciar”.

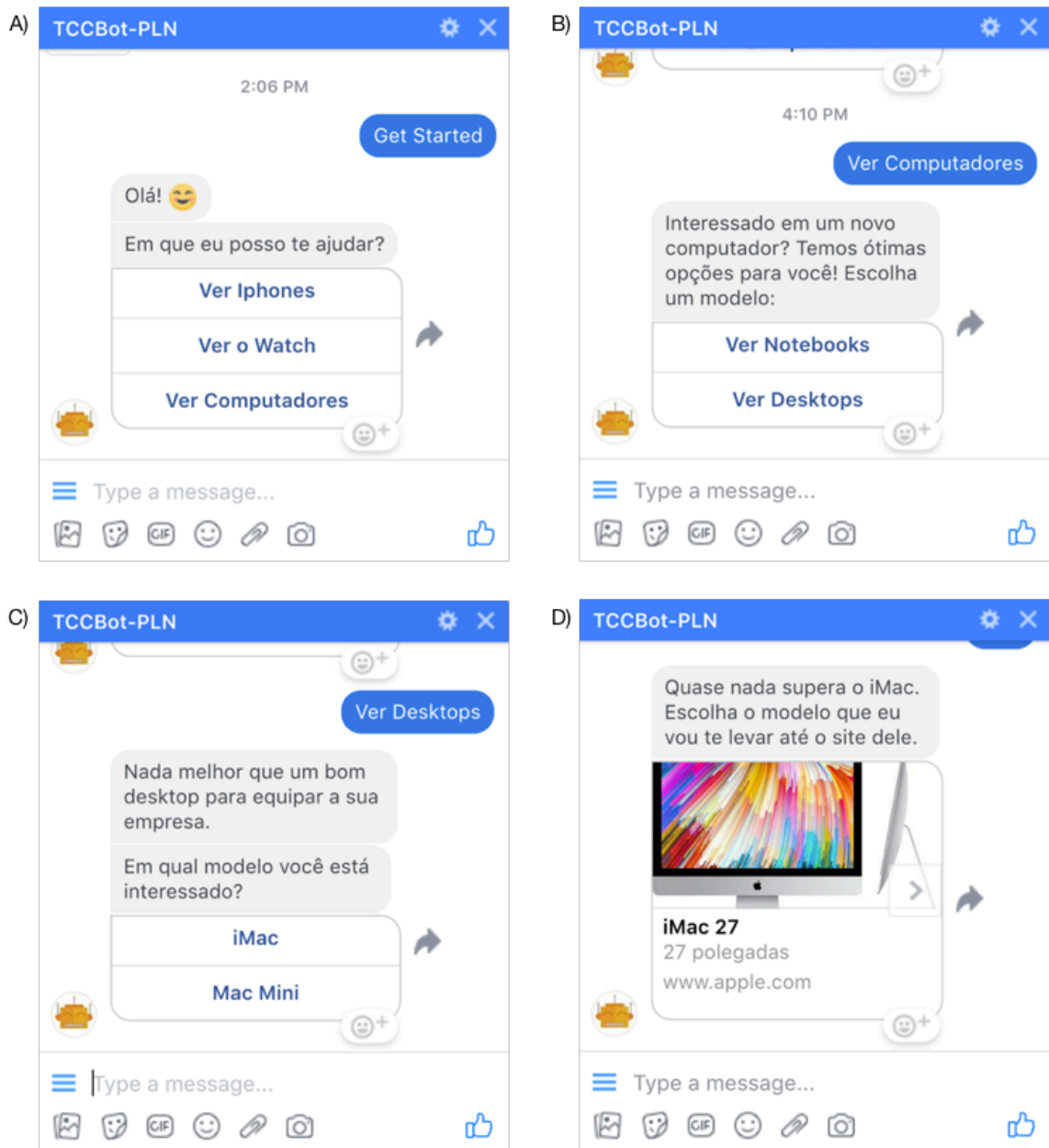
The screenshot displays the configuration interface for a 'Postback: iniciar'. At the top, there are four action buttons: 'VOLTAR' (green), 'DESLIGAR BOTÃO COMEÇAR' (green), 'EDITAR' (blue), and 'REMOVER' (red). Below this, two message configurations are shown. The first configuration is labeled 'text' and contains the message 'Mensagem: Olá! 😊' with 'EDITAR' and 'REMOVER' buttons below it. The second configuration is labeled 'buttons' and contains the message 'Mensagem: Em que eu posso te ajudar?'. Underneath this message, there are three buttons: 'Ver Iphones - postback: iphone-modelos', 'Ver o Watch - postback: watch-modelos', and 'Ver Computadores - postback: comp-modelos'. At the bottom of the configuration area, there is a 'Novo Botão' section with a dropdown menu 'Escolha o tipo de botão', input fields for 'Título do botão' and 'Destino destino?', and a green '+' button. 'EDITAR' and 'REMOVER' buttons are also present at the bottom of this section.

Fonte: Elaborada pelo autor.

A criação desse relacionamento entre *Postbacks* torna possível a construção de um fluxo de interação controlado, reduzindo a possibilidade de desvios acidentais que normalmente poderiam ocorrer em um *chatbot* orientado a conversação. A Figura 27 mostra um dos possíveis fluxos do *chatbot* configurado através do Painel Administrativo da aplicação, finalizado com a exibição de um modelo de mensagem do tipo *slider*, contendo a

listagem dos produtos cadastrados no banco de dados da aplicação, que quando acessados levam o usuário ao *website* do produto.

Figura 27 – Sequência de fluxo de interação: “Ver Computadores” > “Ver Desktops” > “iMac”, configurada através do Painel Administrativo da aplicação.



Fonte: Elaborada pelo do autor.

### 5.3.2 Estados de espera para captura de entradas do usuário

A captura de entradas do usuário é um recurso bastante importante para *chatbots* que pretendem disponibilizar funcionalidades mais personalizadas. Nesse trabalho, seu uso foi

exemplificado através de uma opção na qual o usuário pode enviar sua opinião sobre o serviço oferecido.

A plataforma *Messenger* pode gerenciar esse tipo de recurso através da aplicação de três estados: *Start*, *Check* e *Stop*, monitorados através de métodos distintos fornecidos pela própria plataforma. O estado *Start* inicia o processo de captura, permitindo o envio de mensagens de instrução para o usuário. Na sequência, é acionado o estado *Check*, que faz com que a plataforma permaneça aguardando uma entrada do usuário (Figura 28A). O estado só é modificado após a ação do usuário ser concluída, sendo então acionado o estado *Stop*, que por sua vez encerra o processo permitindo a exibição de uma nova mensagem, instruindo o usuário do término da ação (Figura 28B). O texto informado é recebido pela API de comunicação e então pode ser salvo no banco de dados da aplicação, sendo disponibilizado para o desenvolvedor em uma seção do Painel Administrativo.

Figura 28 – Sequência de envio de sugestões através da utilização de estados pela plataforma *Messenger*.



Fonte: Elaborada pelo autor.

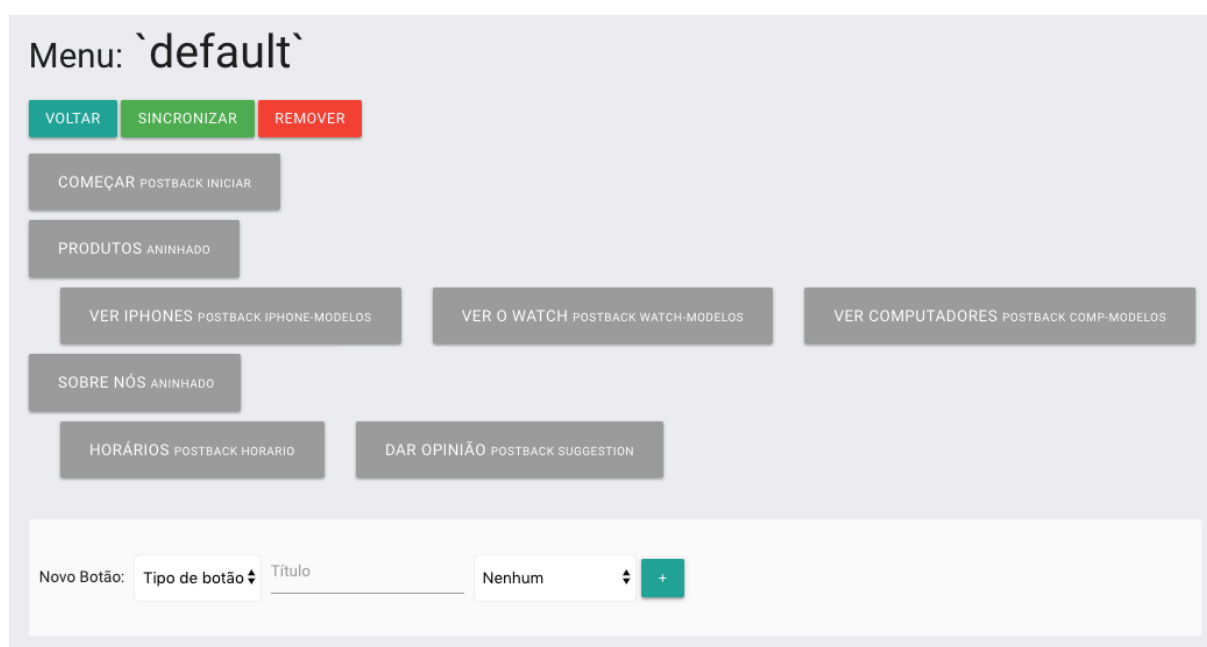
### 5.3.3 Gerenciamento de menus

Outra personalização proporcionada pela plataforma *Messenger*, capaz de facilitar a usabilidade de um *chatbot*, é o uso dos menus. Seu papel é permitir que o usuário acesse um ponto específico dentro do fluxo de interação sem a necessidade de percorrê-lo por completo. A configuração dessa funcionalidade é proporcionada pela plataforma através da execução de métodos específicos descritos em sua documentação, que impõe restrições quanto ao nível de

profundidade dos menus criados e ao número de itens em cada nível a fim de não prejudicar a usabilidade quando em ambientes *mobile*, permitindo no máximo 2 níveis de hierarquia, com no máximo 3 itens cada.

O Painel Administrativo gerencia a criação de múltiplos menus através de uma seção exclusiva, onde o desenvolvedor é capaz de configurar organizações independentes, especificando a linguagem a ser utilizada e se ele deverá permitir ou não o uso do campo de entrada de texto no *chat*. A estrutura de cada menu é construída através do cadastramento de itens de forma hierárquica e seu acionamento pode levar o usuário a link externo, ou ainda a execução de um *Postback*, retomando o fluxo de interação (Figura 29).

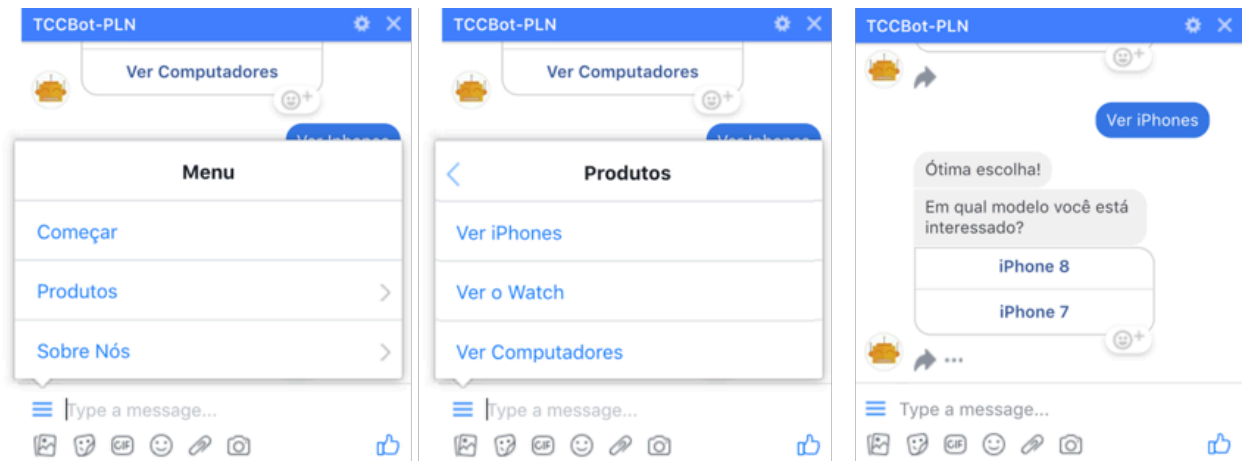
Figura 29 – Interface do Painel Administrativo mostrando a construção do menu intitulado “*default*”.



Fonte: Elaborada pelo autor.

O cadastro dos menus, assim como os itens que os compõe, são mantidos no banco de dados da aplicação e enviados para a plataforma *Messenger* através de um botão de sincronização que executa o método de envio contido na API de comunicação, fazendo a ativação e construção dessa estrutura na janela da aplicação de troca de mensagens da rede social. O resultado obtido pode ser visualizado na Figura 30, que mostra a sequência de interação do item de menu “*Produtos*”, que quando acionado leva à exibição do segundo nível de itens, onde pode ser selecionada a opção “*Ver iPhones*”, que por sua vez, leva ao envio de um modelo mensagem de botões.

Figura 30 – Sequência de uso da estrutura de menu criada no Painel Administrativo da aplicação.



Fonte: Elaborada pelo autor.

#### 5.3.4 Tomada de decisões com base no PLN

Neste trabalho, o tratamento das informações fornecidas pelo sistema *Wit.ai* foi realizado diretamente no código fonte do Painel Administrativo através de uma série de verificações condicionais que, ao detectarem uma determinada entrada, retornam a resposta de texto adequada, armazenada no próprio código.

A Figura 31 mostra exemplos de como o *chatbot* desenvolvido interage com usuário quando as entidades “saudação” e “despedida”, reconhecidas pelo sistema *Wit.ai*, são processadas pelo Painel Administrativo.

Figura 31- Respostas do *chatbot* a detecção das entidades “saudação” (A) e “despedida” (B).



Fonte: Elaborada pelo autor.

É importante ressaltar que outras formas de realizar essa análise e resposta a interpretações são possíveis, inclusive através da própria interface do sistema *Wit.ai*. Entretanto, com a opção tomada de centralizar das configurações do *chatbot* em uma única ferramenta, essa foi a melhor forma encontrada para aplicação do processo de tomada de decisões relacionadas ao PLN de modo a exemplificar de forma clara o uso dessa funcionalidade. Em um plano de desenvolvimento futuro para o Painel Administrativo, seria possível integrá-lo diretamente com o sistema *Wit.ai*, que já disponibiliza ferramentas para esse fim. Com isso, o cadastramento das *Entities*, *Trait Entites* e das respostas a serem retornadas poderá ser feito diretamente na interface do painel e persistidos em seu banco de dados, simplificando o processo de verificação condicional e centralizando ainda mais as funcionalidades do sistema.

## 6 CONCLUSÃO

Embora a interação entre humanos e computadores seja uma área pesquisada há muitos anos, ela continua sendo um tema complexo, tornando o objetivo de criação de um agente de conversação capaz de se comunicar através de uma linguagem natural, de forma coerente e dominando uma ampla gama de tópicos, um desafio.

Os *chatbots* já percorreram um longo caminho desde o seu surgimento até sua recente popularização. Ao longo desse período, vários projetos contribuíram para o desenvolvimento dessa área de pesquisa, incorporando novas tecnologias e impulsionando sua aplicação muito além de seu objetivo inicial, possibilitando sua utilização nas mais diferentes áreas e mudando a forma como os usuários interagem e utilizam serviços digitais.

A API *Conversation* disponibilizada pelo sistema *Watson*, assim como outras plataformas do gênero, pode oferecer uma série de vantagens, principalmente no que se refere ao PLN através de uma interface intuitiva e de fácil entendimento. Entretanto, sua utilização para construção de *chatbots* orientados a conversação requer um alto nível de planejamento para que se atinja um nível de qualidade satisfatório. Essa dificuldade pode ser constatada na construção do *chatbot* realizada através dessa ferramenta, quando a tentativa de manter um fluxo de diálogo foi facilmente frustrada por falhas lógicas no fluxo de conversação, principalmente na tentativa de prever possíveis mudanças de contexto. Mesmo com esse tipo de ferramenta de desenvolvimento, disponibilizar essa abertura para que o usuário decida o rumo da conversa pode representar um grande risco, já que falhas no fluxo do diálogo tendem a ser um fator determinante para o sucesso da aplicação dos *chatbots* no comércio eletrônico.

Por outro lado, *chatbots* orientado a regras e capazes de fazer uso de elementos gráficos para criação de um fluxo de interação restrito e focado se mostraram ser uma opção mais interessante, principalmente quando sua aplicação é voltada a esse segmento do mercado. Por representar um desafio de desenvolvimento menor, se comparado a complexa lógica envolvida na construção de *chatbots* orientados a conversação, essa forma de aplicação da tecnologia passa a ser uma opção de mais fácil acesso ao grande público e promete um maior sucesso em sua aplicação, a medida que sua menor complexidade leva a diminuição da probabilidade de erros no fluxo da interação.

Ao que se refere a experiência de desenvolvimento, o uso da plataforma *Messenger* representou uma facilidade devido a sua disponibilidade de recursos para auxílio a interação do usuário e simplificada integração, tanto com o serviço de troca de mensagens da rede social *Facebook*, quanto com o sistema *Wit.ai*, mostrando-se uma escolha acertada para o tipo



de projeto desenvolvido. Por possuir uma documentação bastante acessível, a plataforma *Messenger* simplificou o processo de desenvolvimento da API de comunicação, necessária para sua integração com o Painel Administrativo, através de variados exemplos que ilustram como o desenvolvedor pode utilizar os recursos oferecidos.

Ao manter o fluxo da conversa restrito, a utilização de *chatbots* como um guia de acesso à produtos e serviços, mesmo representando uma simplificação de suas capacidades, pode ser amplamente explorada a medida que pode proporcionar excelentes resultados.

## REFERÊNCIAS

- ALLEN, J. **Natural Language Understanding**. Redwood City, CA: The Benjamin/Cummings Pub. Co, 1995.
- BECKENKAMP, M. **Você não precisa de Inteligência Artificial para criar um Chatbot**. Bots Brasil, 2017. Disponível em: <https://medium.com/botsbrasil/voc%C3%AA-n%C3%A3o-precisa-de-intelig%C3%Aancia-artificial-para-criar-um-chatbot-fd1c2bcbbeba>. Acesso em 12 de novembro de 2018.
- CONSTINE, J. **Facebook launches Messenger platform with chatbots**. 2016. Disponível em: <https://techcrunch.com/2016/04/12/agents-on-messenger/>. Acesso em 24 de novembro de 2017.
- DALE, R. **The return of Chatbots**. Natural Language Engineering, Cambridge University, n. 22, v. 5, p. 811–817, 2016.
- EEUWEN, M. **Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers**. Dissertação de Mestrado. University of Twente. 2017
- FACEBOOK. **Messenger Platform Docs**. Disponível em: <https://developers.facebook.com/docs/messenger-platform>. Acesso em 13 de novembro de 2018.
- FRYER, L.; CARPENTER, R. **Bots as language learning tools**. Language Learning & Technology, 2006.
- GONZALEZ, M.; LIMA, V. **Recuperação de informação e processamento da linguagem natural**. In: XXIII Congresso da Sociedade Brasileira de Computação. 2003.
- GUTHRIE, L. et al. **The role of lexicons in natural language processing**. Communications of the ACM, v. 39, n. 1, p. 63-72, 1996.
- LALLY, A.; FODOR, P. **Natural language processing with prolog in the IBM watson system**. The Association for Logic Programming (ALP) Newsletter, 2011.
- LICHTNOW, D. **Identificação de assuntos em chats**. Universidade Católica de Pelotas, 2006.
- MAULDIN, M. **Chatterbots, Tnymuds, and the Turing Test Entering the Loebner Prize Competition**. Carnegie Mellon University Center for Machine Translation. Association for the Advancement of Artificial Intelligence, p. 16-21, 1994.
- MUTCHLER A. **A short history of voice revolution**. Voicebot.ai. 2017. Disponível em: <https://www.voicebot.ai/2017/07/14/timeline-voice-assistants-short-history-voice-revolution/>. Acesso em 25 de novembro de 107.
- NETO et al. **Chatterbot em AIML para o Curso de Ciência da Computação**. Centro de Ciências Tecnológicas da Terra e do Mar. Universidade do Vale do Itajaí, 2003.
- NEVES, A. **Iaiml: Um mecanismo para o tratamento de intenção em chatterbots**. Tese de Doutorado, Centro de Informática, UFPE, 2004.
- NOEL, B. **Artificial Intelligence Markup Language (AIML)**. A.L.I.C.E. AI Foundation Working Draft. 2001. Disponível em: <http://www.alicebot.org/TR/2001/WD-aiml/>. Acesso em 21 de outubro de 2017.
- NUNES, F. **Chatbots e Mimetismo: uma conversa entre humanos, robôs e artistas**. Proceedings of 6th International Conference on Digital Arts, ARTECH, 2012.
- PUSTEJOVSKY, J. **The Generative Lexicon**. Cambridge: The MIT Press, 1995.
- RAHMAN et al. **Programming challenges of chatbot: Current and future prospective**. Humanitarian Technology Conference (R10-HTC), IEEE Region 10, 2017.

RODRIGUES, J.; ALMEIDA, I. **Análise dos métodos atuais de aquisição do conhecimento para construção e treinamento de Chatterbots**, Universidade Federal Rural de Pernambuco, 2004.

SAINT-DIZIER, P. **Polymorphic Behavior of Word-senses**. *Linguística Computacional: Investigação Fundamental e Aplicações*. Lisboa: Edições Colibri, p.29-56, 1999.

SCAPINI, I. K. **Relações entre Itens Lexicais**. Poersch, JM; Wertheimer, AMC; Ouro, MEP; Ludwig, EM, p. 393-429, 1995.

SHAWAR, B.; ATWELL, E. **A Comparison Between Alice and Elizabeth Chatbot System**. School of Computing. University of Leeds. 2002.

SHOPIFY. **Conversational Commerce Definition - What is Conversational Commerce**. Disponível em: <https://www.shopify.com/encyclopedia/conversationalcommerce>. Acesso em 24 de novembro de 2017.

SZYMCZAK, A. **Introduction to chatbots in healthcare**. Disponível em: <https://blog.infermedica.com/introduction-to-chatbots-in-healthcare/>. Acesso em 15 de outubro de 2017.

TURING, A. **Computing machinery and intelligence**. [S.l.: s.n., 1950]. Disponível em: <http://loebner.net/Prizef/TuringArticle.html>. Acesso em 15 de outubro de 2017.

ZANDROZNY et al. **Natural language dialogue for personalized interaction**. *Communications of the ACM*, n. 43, v. 8, p. 116–120, 2000.

WALLACE, R. **From Eliza to A.L.I.C.E**, The A.L.I.C.E AI Foundation. Disponível em <http://www.alicebot.org/articles/wallace/eliza.html>. Acesso em 15 de outubro de 2017.

WILKS, Y.; SLATOR, M.; GUTHRIE, L. **Electric words: dictionaries, computers, and meanings**. MIT Press, 1996.

WILCOX, B. **Beyond aiml**, *Chatbots 102*. 2008. Disponível em: [http://www.gamasutra.com/view/feature/3761/beyond\\_aiml\\_chatbots\\_102.php?print=1](http://www.gamasutra.com/view/feature/3761/beyond_aiml_chatbots_102.php?print=1). Acesso em 22 de outubro de 2017.

