

CENTRO UNIVERSITÁRIO DE ITAJUBÁ - FEPI

Curso de Sistemas de Informação

Lucas Souza Nogueira Moreira

## **CHATBOT DE ATENDIMENTO AUTOMATIZADO**

**Itajubá  
2022**

Lucas Souza Nogueira Moreira

## **CHATBOT DE ATENDIMENTO AUTOMATIZADO**

Trabalho de Conclusão de Curso apresentado  
ao curso de Sistemas de Informação do Centro  
Universitário de Itajubá - FEPI como requisito  
parcial para obtenção do título de Bacharel em  
Sistemas de Informação.

**Orientador: Prof. Dr. Gabriel Cirac Mendes  
Souza**

**Itajubá  
2022**

Em sessão às 20 horas do dia 08 de Dezembro de 2022 o acadêmico LUCAS SOUZA NOGUEIRA MOREIRA apresentou o Trabalho de Conclusão de Curso, intitulado “*Chatbot de Atendimento Automatizado*” como requisito para conclusão do Curso de Sistemas de Informação do Centro Universitário de Itajubá – FEPI perante a Banca Examinadora. Depois de todas as considerações feitas, o candidato foi considerado:

Aprovado

Aprovado com Restrições

Reprovado

LUCAS SOUZA NOGUEIRA MOREIRA

Acadêmico

Prof. Dr. Gabriel Cirac Mendes Souza

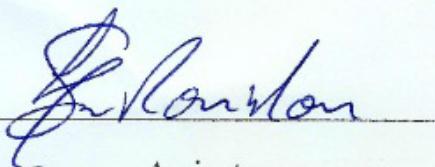
Orientador

Prof. Dr. Márcio de Oliveira

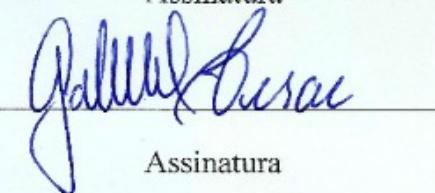
Arguidor 1

Prof. Me. Leandro Duarte Pereira

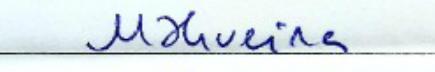
Arguidor 2



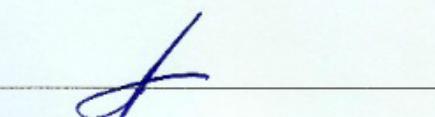
Assinatura



Assinatura



Assinatura



Assinatura

---

MOREIRA, Lucas.

CHATBOT DE ATENDIMENTO AUTOMATIZADO. Lucas Souza Nogueira  
Moreira, Itajubá, 2022, **59** p.

Orientador: Prof. Dr. Gabriel Cirac Mendes Souza

Trabalho de Conclusão de Curso (Graduação), Sistemas de Informação  
Centro Universitário de Itajubá - FEPI.

1. Chatbot. 2. Inteligência Artificial. 3. Linguagem Natural. Atendimento ao Cliente.
  - I. CIRAC, Gabriel. II. Centro Universitário de Itajubá - FEPI. III. CHATBOT DE ATENDIMENTO AUTOMATIZADO. Trabalho de Conclusão de Curso.
-

# Agradecimentos

Em primeiro lugar devo os meus agradecimentos aos meus pais, a minha esposa, e a minha tia. Que fizeram parte de todas essa minha trajetória, por todo o apoio e ajuda, e que graças a confiança deles em mim pude concluir mais essa etapa.

À empresa Aptiv e a todos meus colegas de trabalho, onde fui recebido muito bem, e onde pude me desenvolver como programador de sistemas, e também as várias oportunidades que me foram dadas e que fizeram parte do meu crescimento tanto como pessoa quanto como profissional.

Entretanto, a estrada pra chegar até aqui, foi longa e difícil, e agora eu vejo o quão complicado é mensurar a importância das pessoas que estiveram ao meu redor, e que me auxiliaram nessa caminhada, mas de uma coisa é certa, cada um desses colegas, professores, amigos e familiares fizeram toda a diferença em cada etapa dessa trajetória, e por conta disso não posso expressar de outra forma se não um grande, obrigado.

*“Em algum lugar,  
algo incrível está esperando  
para ser descoberto”  
(Carl Sagan)*

# Resumo

O desenvolvimento tecnológico vem crescendo cada vez mais rápido ao longo dos anos e tecnologias que antes não eram viáveis devido a diversas limitações estão cada vez mais presentes em nosso cotidiano. Graças a isso, muitas empresas estão se adaptando ao modelo de vendas online, por ser uma alternativa menos burocrática e com alto retorno sobre o investimento. Este trabalho propõe o estudo e desenvolvimento de um serviço automatizado de *chatbot*, para empresas interagirem e realizarem vendas através da plataforma de mensagens instantâneas do *WhatsApp*, permitindo que os clientes realizem compras através da conversa de forma ágil e sem contato e interação humana. A metodologia utilizará ferramentas de ponta disponíveis no mercado para implementar a comunicação com a plataforma de mensagens e interpretar a linguagem natural. Portanto, abstrairá da conversa, palavras-chave e contextos necessários para apresentar as melhores respostas em cada cenário e pergunta. Dado o estado atual do mercado, soluções como estas são essenciais para melhorar a proximidade da empresa com os seus clientes e tornam-se uma estratégia fundamental para garantir a fidelização e efetuar uma venda. Os resultados da investigação sugerem a viabilidade do aplicativo, pois o protótipo proporcionou uma comunicação assertiva em nossos contextos de uso criados.

**Palavras-chaves:** Chatbot. Linguagem Natural. Atendimento ao Cliente. Programação. Inteligência Artificial.

# Abstract

Technological development has been growing faster and faster over the years and technologies that were previously not viable due to various limitations are becoming more present in our daily lives. Thanks to this, many companies are adapting to the online sales model, as it is a less bureaucratic alternative with a high return on investment. This work proposes the study and development of an automated service *chatbot*, for companies interact and make sales through the instant messaging platform of *WhatsApp*, allowing customers to carry out purchases through the conversation in an agile way and without contact and human interaction. The methodology will use state-of-the-art tools available on the market to implement the communication with the messaging platform and to interpret natural language. Therefore, it will abstract from the conversation, keywords, and contexts necessary to present the best answers in each scenario and question. Given the current state of the market, solutions like these are essential to improve the company's proximity to its customers and become a fundamental strategy to guarantee the loyalty and make a sale. The investigation results suggest the viability of the application, as the prototype provided assertive communication in our created usage contexts.

**Key-words:** Chatbot. Natural Language. Customer service. Programming. Artificial intelligence.

# Listas de ilustrações

Figura 1 – Conversa do programa "ELIZA" com um humano . . . . .	17
Figura 2 – Conversa com atendimento da Domino's Pizza via WhatsApp . . . . .	20
Figura 3 – Fluxo de execução do agente . . . . .	24
Figura 4 – Fluxo de <i>intent</i> e resposta ao usuário . . . . .	25
Figura 5 – Conceitos Inteligência artificial . . . . .	27
Figura 6 – Esquema projeto de aprendizado de máquina . . . . .	29
Figura 7 – Processo de desenvolvimento incremental . . . . .	31
Figura 8 – Ciclo da execução do <i>Scrum</i> . . . . .	31
Figura 9 – Caso de uso: usuário . . . . .	36
Figura 10 – Caso de uso: gestor . . . . .	37
Figura 11 – Quadro de prioridade método MoSCoW . . . . .	38
Figura 12 – Wireframe do produto, parte 1 . . . . .	39
Figura 13 – Wireframe do produto, parte 2 . . . . .	40
Figura 14 – MVC (Model View Control) . . . . .	41
Figura 15 – Fonte: Elaborado pelo autor . . . . .	41
Figura 16 – Função de decodificação . . . . .	42
Figura 17 – Função de decodificação . . . . .	43
Figura 18 – Mind map estrutura do <i>chatbot</i> . . . . .	44
Figura 19 – <i>Intent</i> utilizadas na função do Carrinho . . . . .	44
Figura 20 – Exemplo da construção de cada <i>Intent</i> . . . . .	45
Figura 21 – Código que informa os produtos no carrinho do cliente . . . . .	46
Figura 22 – Exemplo do mensagem mostrando os produtos no carrinho . . . . .	47
Figura 23 – Exemplo de iteração mostrando as categorias e produtos . . . . .	47
Figura 24 – Exemplo das informações dos produtos no banco de dados . . . . .	48
Figura 25 – Código que realiza a consulta no <i>Firebase</i> . . . . .	49
Figura 26 – Exemplo das informações dos pedido no banco de dados . . . . .	50
Figura 27 – Resultado obtido, 1 . . . . .	51
Figura 28 – Resultado obtido, 2 . . . . .	52
Figura 29 – Resultado obtido, 3 . . . . .	53

# **Lista de tabelas**

Tabela 1 – Empresa que venderam pela internet nos últimos 12 meses, por tipo de canal online em que ocorreu a venda . . . . .	19
Tabela 2 – Empresa que venderam pela internet nos últimos 12 meses, por tipo de canal online em que ocorreu a venda . . . . .	34
Tabela 3 – Levantamento de regra de negócio . . . . .	34
Tabela 4 – Levantamento de requisitos não funcionais . . . . .	35
Tabela 5 – Tabela de especificação . . . . .	55

# **Lista de abreviaturas e siglas**

API	Application Programming Interface
IA	Inteligência Artificial
JSON	JavaScript Object Notation
NoSQL	Not Only SQL
SQL	Standard Query Language
PLN	Processamento de Linguagem Natural
RF	Requisito Funcional
RFN	Requisito Não Funcional
RN	Regra de Negócio
GPS	Sistema de Posicionamento Global
MVC	Model View Control
MVP	Minimum Viable Product

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Objetivo	15
1.2	Objetivo específicos	15
1.3	Motivação	15
1.4	Metodologia de pesquisa	15
1.5	Organização do estudo	16
<b>2</b>	<b>Revisão da literatura</b>	<b>17</b>
2.1	História dos chatterbot	17
2.2	Atendimento ao cliente	18
2.3	Tendências de mudança	18
2.3.1	Aplicações Atuais e Estado da Arte	20
<b>3</b>	<b>Referencial Teórico</b>	<b>22</b>
3.1	Processamento de Linguagem Natural	22
3.2	API WhatsApp	22
3.3	Twilio	23
3.4	Introdução ao Dialogflow	23
3.4.1	Agente	24
3.4.2	Intents	24
3.4.3	Fulfillments	26
3.5	Firebase	26
3.6	Inteligência Artificial	27
3.7	Aprendizado de Máquina	28
<b>4</b>	<b>Metodologia</b>	<b>30</b>
4.1	Pesquisa Exploratória	30
4.2	Método Ágil de Desenvolvimento	30
<b>5</b>	<b>Desenvolvimento da Solução</b>	<b>33</b>
5.1	Levantamento de Requisitos de Software	33
5.2	Requisitos Funcionais	33
5.3	Regra de Negócio	34
5.4	Requisitos Não-funcionais	35
5.5	Caso de Uso	35
5.6	MoSCoW	37
5.7	Prototipagem de Software	38
5.8	Ferramentas Utilizadas	40
5.9	Configuração da API	41
5.10	Criação do <i>Chatbot</i>	43

5.11	Banco de Dados . . . . .	47
<b>6</b>	<b>Resultados . . . . .</b>	<b>51</b>
6.1	Discussão . . . . .	53
6.1.1	Precificação do projeto . . . . .	54
6.2	Desenvolvimentos futuros . . . . .	55
<b>7</b>	<b>Considerações Finais . . . . .</b>	<b>57</b>
	<b>Referências . . . . .</b>	<b>58</b>

# 1 Introdução

Com o crescimento contínuo das tecnologias nos diversos setores da indústria e comércio, cada vez mais esse recurso está acessível ao público geral através dos diversos meios. Os *smartphone*, por exemplo revolucionaram a forma de acesso à internet trazendo computação e comunicação próximas das pessoas e do seu dia a dia.

A comunicação sempre foi uma das áreas mais estudadas pelos pesquisadores da tecnologia e o interesse por compreender e replicar a linguagem humana vem de muito tempo como, por exemplo, o trabalho de Alan Turing, que em 1950 publicou o “*Computing Machinery and Intelligence*” que trouxe os conceitos que ficaram conhecidos como teste de Turing. De forma geral, esse teste verifica a capacidade operacional de um computador para replicar a inteligência e comportamento humano ([TURING; HAUGELAND, 1950](#)).

Um dos primeiros softwares de processamento de linguagem natural e simulação de diálogos foi criado por Joseph Weizenbaum em 1966 e denominado ELIZA, que examinava perguntas feitas ao sistema, extraia as palavras chaves e retornava uma sentença de resposta ([WEIZENBAUM, 1966](#)). O programa fazia o seu trabalho com tanta eficiência que foi considerado o primeiro sistema a passar pelo teste de Turing. O software ELIZA foi um grande trabalho da área de inteligência artificial e é a base fundamental do funcionamento dos *chatbots* utilizados como referência até os dias de hoje.

Já no Brasil, nos últimos anos, o comercio eletrônico vem sendo uma das vertentes da internet que segue em grande expansão, mesmo após mais de vinte anos, desde que as primeiras plataformas de venda começaram a surgirem no território nacional. Principalmente durante o período de restrições devido o vírus da *covid-19* iniciados em 2020, o avanço do consumo online de serviços e compras aumentou exponencialmente, segundo pesquisas da CNDL/SPC Brasil. Somente entre o período de 2020 a 2021, 91% dos pesquisados realizaram algum tipo de compra pela internet, o que movimento em torno de R\$ 87,4 bilhões de reais no Brasil ([BARBOSA, 2021](#)), e mesmo os pequenos comércios locais tiveram que se adaptar as necessidades e a essas tendências tecnológicas.

Graças a isso, novas formas de consumo e atendimento online surgiram com o intuito de facilitar, agilizar e oferecer uma experiência de compra melhor e mais flexível para os usuários. Desde as opções mais simples, que acabaram sendo utilizadas para a interação com o cliente, até vendas, estão adotando os aplicativos de mensagens de texto instantânea como o *Whatsapp*, *Telegram*, *Facebook Messenger*. Para agilizar o processo de atendimento ao cliente nessas plataformas, empresas de grande porte optaram pela estratégia de utilizar *chatbots*, que estimulam e otimizam as conversas com os clientes. Neste sentido, esta investigação se propõe a criar através de um estudo de caso um *chat-*

*bot* utilizando tecnologias como o *WhatsApp* e ferramentas disponíveis na *Google Cloud Platform*. Com isso, espera-se desenvolver e fixar as habilidades adquiridas ao longo do curso, além de propor uma solução prática e de fácil implementação. Finalmente, este trabalho irá fazer uma revisão da literatura para levantar o estado da arte sobre o tópico e principais tecnologias.

## 1.1 Objetivo

O objetivo desse trabalho é realizar um estudo e desenvolvimento de uma solução de *chatbot*, para realizar o atendimento automatizado através do aplicativo *WhatsApp*, e que simule uma conversação de venda e atendimento de uma lanchonete.

## 1.2 Objetivo específicos

Os objetivos específicos do desenvolvimento desse trabalho foram:

- Simular uma experiência de conversa e atendimento;
- Desenvolver um mínimo produto viável de uma solução de vendas por mensagem;
- Analisar os custos associados a aplicação da solução em um negócio real;
- Utilizar ferramentas tecnológicas em nuvem de alto nível, que auxiliem no desenvolvimento desse tipo de solução;

## 1.3 Motivação

Com a aumento da acessibilidade a tecnologia e crescimento do comércio eletrônico, o atendimento se tornou uma estratégia para garantir a fidelidade do cliente e concluir uma venda. Para isso, ferramentas que otimizem esse trabalho são imprescindíveis para aumentar a eficiência e diminuir os custos.

Com esse propósito, os *chatbots* oferecem os recursos e ferramentas necessárias para realizar essa tarefa, aumentando a interação da empresa e consequentemente deixando o cliente satisfeito.

## 1.4 Metodologia de pesquisa

Para realizar a investigação, o trabalho adotou uma análise e pesquisa exploratória. Para tanto, foram selecionados diversos artigos e documentos sobre a aplicação de um sistema de atendimento automatizado em um negócio e as formas de desenvolvimento

possíveis. Além disso, o método de pesquisa experimental realizou um estudo controlado sobre os dados levantados, para comprovar os principais achados das pesquisas analisadas anteriormente, com o intuito de melhorar a experiência do cliente ao utilizar a solução desenvolvida. Finalmente, para organizar e controlar o desenvolvimento do *chatbot* este trabalho adotou a metodologia ágil *SCRUM*.

## 1.5 Organização do estudo

Para descrever o estudo, a pesquisa foi estruturada em formato de tópicos, descritos abaixo:

- Tópico 2 – Revisão da literatura, que é um levantamento da história dos *chatbots* e uma apresentação da utilização deles no mercado.
- Tópico 3 – Referencial teórico, onde é descrito todo o estudo realizado e conhecimentos adquiridos para a construção do sistema e explicação de cada ferramenta utilizada.
- Tópico 4 – Metodologia, onde será apresentado as metodologias utilizadas para desenvolver a solução.
- Tópico 5 - Desenvolvimento da solução, descrição do processo de desenvolvimento e exemplo da construção do sistema.
- Tópico 6 – Resultados, é a demonstração do produto final desenvolvido durante o projeto.
- Tópico 7 – Considerações finais, traz a conclusão referente a pesquisa realizada, a especificação do uso da aplicação e os desenvolvimentos futuros.

## 2 Revisão da literatura

## 2.1 História dos chatterbot

Os *chatbots* ou também conhecidos como *chatterbot*, termo que tem origem etimológica da combinação de palavras *chatter* significando a pessoa que conversa e a palavra *bot* que é um diminutivo para a palavra *robot*, palavra essa que foi criada em 1994 por Michael Mauldin na *Twelfth National Conference on Artificial Intelligence* para descrever os robôs de conversa (MAULDIN, 1994).

Estes são um tipo de sistemas computacionais que de forma geral tentam reproduzir a conversação entre pessoas, interagindo por meio de uma interface com o indivíduo, podendo ser de forma textual ou por voz (SOFTIĆ et al., 2021) e, normalmente, tem o objetivo de responder as perguntas realizadas pelo usuário de forma que ele não consiga distinguir a diferença entre um *chatbot* ou uma pessoa real. Na Figura 1 é possível observar uma captura de tela da interação entre um usuário e o *chatbot* ELIZA.

Figura 1 – Conversa do programa "ELIZA" com um humano

File Edit Options Buffers Tools Help

✖️ 🗑️ ✎ ↺ ✖️ 🗑️ 🔍 🗃 🗃 ?

I am the psychotherapist. Please, describe your problems.

Each time you are finished talking, type RET twice.

I have a terrible headache.

Maybe you should consult a doctor of medicine, I am a psychiatrist.

Okay, I'll do that then.

Can you elaborate on that?

What are you talking about? No I cannot elaborate. See you later.

Why do you say that?

Now you're just talking nonsense!

What makes you believe now I am just talking nonsense?

You are not making any sense.

Perhaps your plans have something to do with this.

-u:\*\* \*doctor\* (Doctor Fill)--L1--Top-----

<sup>1</sup>Fonte: Página sobre ELIZA no Wikipedia

Os *chatbots*, tal qual várias outras aplicações em tecnológicas, são ferramentas desenvolvidas com o intuito de aumentar produtividade, agilizar atividades e facilitar processos manuais. Também são largamente utilizados como forma de estudo sobre a aplicação de inteligência artificial e interpretação de linguagem natural.

Com crescimento do comércio eletrônico e aumento na demanda do atendimento ao cliente, ocorreu também um aumento na utilização de *chatbots* em empresas, visto que o sistema garante diversos ganhos e permite o retorno rápido e aumento de produtividade. Porém, a qualidade do *chatbot* impacta diretamente nestes ganhos, pois se o sistema não apresentar respostas claras, ou se sua detecção de linguagem for falha e não conseguir solucionar os questionamentos dos usuários, o programa se torna inviável, assim, a tecnologia ou inteligência artificial utilizada deve se adaptar às condições de cada empresa que emprega *chatbot*, pois deste modo permite obter um conhecimento mais aprimorado sobre seus clientes.

## 2.2 Atendimento ao cliente

O atendimento ao cliente pode ser simplificado, onde todo o contato realizado antes, durante e depois, com o intuito de vender um produto ou serviço, possa ser feito de forma automatizada. Entretanto, o processo não se resume somente em realizar uma venda, mas também em toda a experiência de compra. Além disso, é preciso entender as exigências e as vontades de cada tipo de perfil. Quando um cliente procura determinado atendimento, significa que ele confia na sua empresa e a mesma deve satisfazê-lo. Pesquisas da *Salesforce* ([MUICAHY, 2020](#)), sugerem que 80% dos indivíduos entrevistados consideram que a experiência oferecida por uma marca é tão importante quanto a qualidade dos produtos e serviços que ela oferece.

## 2.3 Tendências de mudança

Dada a importância do atendimento ao cliente e a respectiva a experiência oferecida, diversas empresas estão aderindo a novos métodos, visando trazer clientes para o seu negócio. O comércio digital vem sendo uma das principais opções para aqueles que desejam alcançar uma quantidade maior de pessoas e pesquisas demonstram que durante o período de 2019 a 2021, a proporção de empresas que comercializaram qualquer tipo de produto ou serviço por meio de vendas online cresceu de 57% para 73% ([NIC.BR, 2022](#)), segundo a mesma pesquisa, entre as empresas que venderam pela internet, o método mais utilizado foram os aplicativos de mensagens como o *WhatsApp*, sendo este o mais utilizado pelas pequenas empresas. A Tabela 1 apresenta os dados summarizados coletados pela pesquisa citada.

<sup>1</sup> Disponível em: <<https://pt.wikipedia.org/wiki/ELIZA>>. Acesso 25 mar. 2022.

Tabela 1 – Empresa que venderam pela internet nos últimos 12 meses, por tipo de canal online em que ocorreu a venda

	Percentual(%)	Mensagem de WhatsApp ou chat do Facebook	Website da empresa	Sites de vendas, como Mercado Livre, OLX, Submarino, etc.	Redes sociais, como Facebook, Instagram, etc.	App
	Total	78	30	23	39	31
PORTE	De 10 a 49 pessoas	80	28	22	40	31
	De 50 a 249 pessoas	67	38	26	28	33
	De 250 ou mais pessoas	54	55	28	36	39
REGIÃO	Norte	85	20	13	45	35
	Nordeste	79	28	24	51	38
	Sudeste	75	31	24	31	29
	Sul	82	33	21	43	34
	Centro-Oeste	78	23	24	45	27
MERCADO DE ATUAÇÃO	Indústria de transformação	75	34	21	31	24
	Construção	66	28	18	34	19
	Comércio, reparação de veículos automotores e motocicletas	85	24	25	44	31
	Transporte, armazenagem e correio	73	31	24	22	31
	Alojamento e alimentação	81	32	22	47	61
	Informação e comunicação	70	48	18	37	28
	Atividades imobiliárias, atividades profissionais, etc...	64	39	19	30	22

Em relação a pesquisa citada como referência, o gerente do Cetic.br|NIC.br, Alexandre Barbosa avalia que:

"O uso de aplicativos de mensagens nas vendas foi a forma encontrada por muitas empresas para contornar os efeitos da pandemia e em um contexto de distanciamento social. A popularização dos pagamentos por meio do Pix também é uma tendência importante revelada pelos indicadores da TIC Empresas" (CETIC.BR, 2022).

E é nesse contexto que tecnologias como os *chatbots* estão ganhando espaço no mercado, com o intuito de maximizar a eficiência do atendimento ao cliente de forma inteligente. Além disso, essa solução evita a necessidade de uma pessoa para responder a cada mensagem recebida e consegue interpretar as perguntas do cliente de forma assertiva, buscando em sua base de conhecimento a melhor repostas possível para cada problema.

### 2.3.1 Aplicações Atuais e Estado da Arte

O cenário atual está migrando para o meio eletrônico e diversas empresas estão buscando soluções como os *chatbots*. Tais *bots* variam de acordo com o foco da empresa, mas muitos deles utilizam plataformas de mensagem eletrônica já conhecidos como *Facebook Messenger*, *WhatsApp* e *Telegram*, algumas ainda arriscam outras alternativas para implementar seu atendimento como *Amazon Alexa*, *Google Assistant* ou até por ligação telefônica.

Dentre as empresas brasileiras, uma das referências na utilização de *chatbots* em seu atendimento é o Banco Bradesco. Apelidado de Bia, o *bot* do Bradesco realiza atendimento personalizado para cada cliente, informando o estado da conta, saldo e extrato das ultimas transações.

Já a rede americana de pizzaria Domino's Pizza, famosa por ser uma das maiores empresas de alimentação, utiliza o *chatbot* como forma de atendimento via *WhatsApp* desde 2019 no Rio de Janeiro e teve expansão para outras cidades brasileiras a partir do primeiro semestre de 2020 (PAIVA, 2018), na Figura 2 é possível observar uma captura de tela de uma conversa com a atendimento da empresa.

Figura 2 – Conversa com atendimento da Domino's Pizza via WhatsApp



Fonte: Página do site mobiletime<sup>2</sup>

A empresa se Bebidas Ambev utiliza um *chatbot* interno, formulado para tirar dúvidas de seus colaboradores sobre agendamento de férias, pagamento, benefícios, suporte e apoio interno, segundo a desenvolvedora da solução, a empresa *Inbot*, o sistema teve mais de 7 mil interações no primeiro mês de implantação e tem previsão de atingir mais

<sup>2</sup> Disponível em: <<https://www.mobiletime.com.br/noticias/13/11/2019/bot-da-dominos-vende-pizza-pelo-whatsapp-no-rio-de-janeiro/>>. Acesso 30 mar. 2022.

de 80 mil interações mensais. Assim, percebe-se que há uma tendência muito grande em adotar *chatbots* em sistemas de venda.

## 3 Referencial Teórico

### 3.1 Processamento de Linguagem Natural

Também conhecido com PLN, o processamento de linguagem natural é o meio de estudo que busca permitir a interação de uma máquina e a linguagem do ser humano. Sendo linguagem natural o termo da ciência que define nossa principal e mais comum forma de comunicação, a linguagem, e os próprios idiomas que utilizamos são exemplos de língua natural. Segundo o estudo ([SILVA et al., 2007](#)), a arquitetura de um sistema de processamento de linguagem natural utiliza algumas etapas para realizar sua interpretação de forma confiável, mas nem sempre essas etapas são realizadas de forma sequencial. Como as informações de resposta são interdependentes e serão utilizadas no contexto geral da frase, pode-se definir as seguintes etapas:

- Analisador léxico: Primeiro processo, responsável por separar os componentes mais importantes das sentenças e analisar cada palavra e pontuação gramatical;
- Analisador sintático: Nesta etapa o sistema cria uma estrutura sintática que tenha o mesmo significado do texto de entrada, mas estruturada de uma forma mais simples para o computador;
- Analisador semântico: Processo da análise que busca entender os componentes da sentença através do seu significado no contexto geral da frase, analisando o significado da palavra de acordo do contexto das outras palavras do texto;
- Analisador do discurso: Na análise do discurso, busca-se entender o significado da sentença dentro do contexto geral da conversa, analisando as frases que antecedem a atual;
- Analisador pragmático: Por fim, a forma que nós expressamos verbalmente uma frase, pode atribuir significados distintos, podendo dizer que uma frase está sujeita aos aspectos pragmáticos da nossa comunicação e implicando em interpretações diferentes de uma mesma sentença.

### 3.2 API WhatsApp

Como o *WhatsApp* é a plataforma de mensagem eletrônica mais popular do Brasil e, muitas pequenas e médias empresas se adaptaram para utilizar a plataforma como

ferramenta de atendimento e de vendas dos seus produtos, a plataforma foi escolhida como principal foco do desenvolvimento desta solução de *chatbot*.

O *WhatsApp* oferece uma solução voltada para empresas de pequeno porte pelo *WhatsApp Business APP*, que é um novo aplicativo com o princípio de auxiliar as vendas e que possui algumas funcionalidades de mensagens rápidas pré cadastradas e criação de perfil comercial. Dependendo do objetivo da empresa, soluções que utilizam diretamente a *API* do *WhatsApp Bussiness* agregam mais benefícios, possibilidade manipular a ferramenta de uma forma mais ampla, podendo até mesmo integrar o aplicativo de mensagens a outros sistemas de gestão e controle.

Entretanto, a utilização ao *WhatsApp Busines API* é restrito para as empresas licenciadas e não está liberado para uso público, como alternativa para isso, alguns representante oficiais do *WhatsApp*, que são conhecidos como *Business Service Provider*, disponibilizam soluções para utilizar essa *API* sem a necessidade de possuir essa licença.

Para o desenvolvimento da solução estudada e integração com o *WhatsApp API*, foi utilizado os serviços da empresa *Twilio*, logo que essa oferece um suporte nativo ao Brasil, possui maior integração com ferramentas externas, e os custos do serviço são mais viáveis.

### 3.3 Twilio

A plataforma da *Twilio* foi a empregada como um conector do projeto, ou seja, ela é a provedora oficial de serviço que possibilita utilizar as informações da *API* do *WhatsApp*. A plataforma oferece um número de celular de teste que pode ser utilizado dentro do aplicativo de mensagens para realizar as conversas num ambiente experimental de desenvolvimento e essa ferramente é chamada *sandbox*.

Dentro da plataforma foi utilizado a ferramenta de Funções, que disponibiliza um ambiente *serverless*, ou seja, um ambiente de execução em nuvem, que consegue tratar requisições orientadas a eventos, com isso, foi possível criar um serviço de comunicação, que irá receber as solicitações de entrada, enviar para a plataforma fazer interpretação da linguagem do texto e receber como resposta um objeto no formato de *JSON*, que é necessário para realizar a conversão para a formatação utilizada pelo aplicativo do *WhatsApp*.

### 3.4 Introdução ao Dialogflow

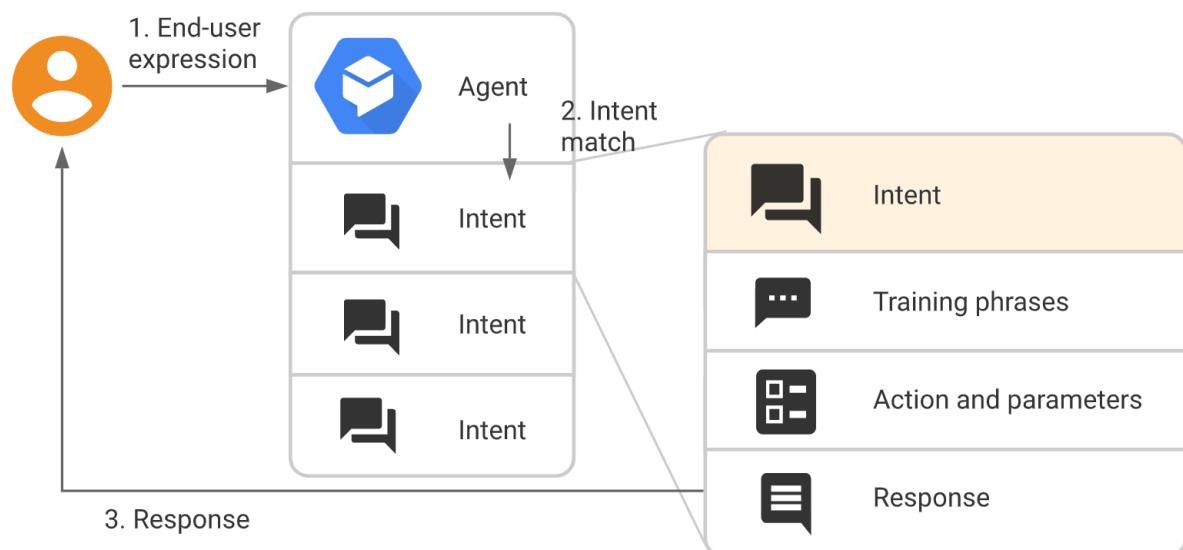
O *Dialogflow* conhecido anteriormente como *API.AI* é uma solução que foi adquirida pela *Google* para construção de conversas, sua plataforma de desenvolvimento é totalmente web e disponibilizada como uma das ferramentas da *Google Cloud Platform*.

Esta *API* evolui durante os anos por ter sido uma das principais ferramentas de desenvolvimento de aplicações para o *Google Assistant* e hoje é uma das ferramentas mais antigas e maduras disponíveis no mercado. A plataforma proporciona ao desenvolvedor a liberdade de criar uma experiência de conversação completa e, para isso, disponibiliza ferramentas para lidar com as tarefas de gestão de conversa e de entendimento, além de geração de expressões em linguagem natural (JANARTHANAM, 2017). Nos tópicos a seguir, serão abordadas as funções disponíveis na plataforma para a construção de um *chatbot*.

### 3.4.1 Agente

No *Dialogflow* o agente é o módulo responsável para processar as conversas simultâneas de cada usuário. O agente é a primeira etapa, ele recebe cada frase da conversa e utiliza processamento de linguagem natural para entender a linguagem humana e suas nuances. Além disso, utiliza frases de treinamento pré-definidas para transcrever a ação para a *intent* responsável por gerar a resposta. Na Figura 3 é possível observar o caminho que é utilizado para gerar a resposta ao usuário.

Figura 3 – Fluxo de execução do agente



Fonte: Imagem disponível na documentação do Dialogflow <sup>1</sup>

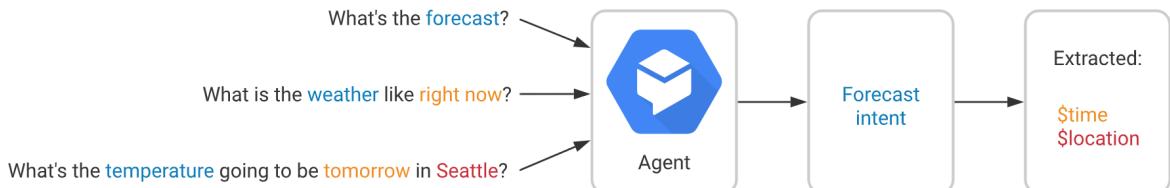
### 3.4.2 Intents

Para definir a melhor resposta, o *Dialogflow* analisa a intenção do texto diversas vezes, em seguida extrai as informações e parâmetros nela contidas e utiliza estes valores para gerar a melhor resposta para aquela frase. Cada *chatbot* irá possuir várias *intents*,

<sup>1</sup> Disponível em: <<https://cloud.google.com/dialogflow/es/docs/intents-overview>>. Acesso 30 set. 2022.

correspondendo a uma possível resposta que o sistema realizará. Na Figura 4 é possível analisar como cada mensagem é recebida pelo agente e transferida para a *intent* correta.

Figura 4 – Fluxo de *intent* e resposta ao usuário



Fonte: Imagem disponível na documentação do Dialogflow <sup>2</sup>

Toda *intent* possui alguns itens que a definem, sendo esses:

- Frases de treinamento: são as frases cadastradas que o sistema utilizará como referência para associar a *intent* que será executada, a frase cadastrada não precisa ser igual ao valor de entrada pois é utilizado *machine learning* para expandir sua lista e prever frases semelhantes.
- Ação e parâmetros: são configurados nesses campos os valores esperados que a *intent* receberá pelo texto, quando correspondido esses valores são extraídos da expressão e utilizados para realizar uma função e proporcionar uma resposta mais complexa, como no exemplo da Figura 4 onde é abstruído a localização e o tempo em que se deseja consultar a previsão do tempo.
- Resposta: as respostas são as mensagens que o chatbot irá retornar para o usuário, podendo variar dependendo de como a frase foi composta pelo usuário e pode ser configurado para responder à pergunta, solicitar mais informações ou encerrar a conversa.
- Contexto: O contexto é utilizado para dar sentido a frase e pode ser utilizado como forma de adicionar informações necessárias para responder a uma pergunta subsequente do usuário, quando configurado o sistema armazena informações de contextualização onde conversa está situada, e pode variar qual *intent* realizará a resposta, dependendo do estado atual da aplicação.
- Eventos: cada *intent* pode invocar um evento diferente, como exemplo a consulta em uma *API* que retornará a previsão do tempo, e essa configuração poderá ser feita através de uma função externa configurada dentro do módulo de *fulfillment*.

<sup>2</sup> Disponível em: <<https://cloud.google.com/dialogflow/es/docs/intents-overview>>. Acesso 30 set. 2022.

### 3.4.3 Fulfillments

Com o desenvolvimento da aplicação tornando-se mais robusta, temos a necessidade de que ela seja mais dinâmica e realize comunicações e interações com outros sistemas, como uma *API* ou um banco de dados. Para isso, é preciso desenvolver uma fonte externa de *backend* e realizar a comunicação através de um *webhook*. Essa função é chamada pela plataforma como *fulfillment*.

Em nossa aplicação o *fulfillment* foi utilizado para buscar e inserir valores na base de dados do *Firebase*, ele realiza a consulta das informações e itens cadastrados e, sempre que o cliente conclui um pedido, as informações importantes são salvas no banco de dados. Essa tarefa é realizada pelo *chatbot* durante a conversa com cada usuário e cada usuário possui sua sessão própria de iteração com o *bot*.

O *webhook* é uma tecnologia que possibilita a consulta em tempo real de dados entre duas aplicações diferentes, a funcionalidade é executada a partir de eventos e sempre terá um receptor em um emissor e realiza a atualização dos dados de forma rápida ([CALADO, 2022](#)).

O *Dialogflow* permite que a estrutura do *webhook* possa ser construída tanto de forma externa quanto dentro da própria plataforma. Com isso, não é necessário a utilização de um outro serviço para hospedar a ferramenta e o *Node.js* é utilizado para realizar essa construção, o que facilita o desenvolvimento da aplicação.

## 3.5 Firebase

O *Firebase* é um conjunto de vários serviços de hospedagem e é uma das ferramentas disponíveis na *Google Cloud Platform*. Dentre as diversas opções de hospedagem do sistema, a opção escolhida foi o *Realtime Database*, essa ferramenta permite a criação de uma base de dados em nuvem, que possibilita a consulta e inserção das informações em tempo real através de uma *API* disponibilizada pelo serviço.

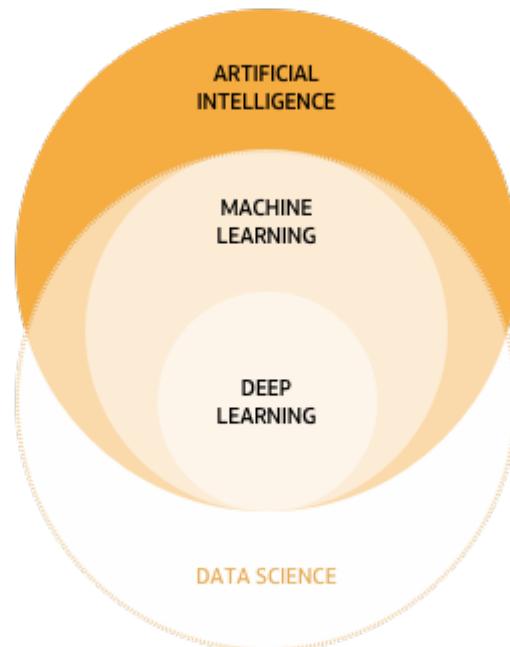
O esquema utilizado pelo *Firebase* para fazer o armazenamento dos dados é o *NoSQL*, sigla do inglês para banco de dados não relacional. A diferença desse tipo de esquema comparado com os outros tipos, é que não há a necessidade da construção de um banco de dados relacional. Os valores e dados são salvos na formato de documento *JSON*, o que quer dizer que cada campo pode ser possuir um valor escalar, contendo valores como *number*, *string*, *boolean*, *array*, ou uma lista de valores como objeto. Esse modelo de estrutura facilita a integração da base de dados com a aplicação, já que o *JSON* é um formato fácil de ler e consumir, considerando as linguagens que foram utilizada e que são baseadas em *Javascript*. Além disso, o próprio *Firebase* facilita a integração de sua *API* com esses tipos de linguagens.

### 3.6 Inteligência Artificial

A IA ou inteligência artificial representam uma classe de algoritmos e sistemas de software que visam imitar a inteligência humana, ou seja, pretendem criar alguma capacidade de compreensão, solucionar problemas do mundo real, memorizar algum tipo de fato e simular a lógica e abstração do homem; Essas são alguns exemplos de funções esperadas de uma IA.

O termo inteligência artificial com o tempo se tornou genérico, não abrangendo de forma ampla toda sua área de estudo. Assim, foram criadas subdivisões do estudo, como o *Machine Learning*, *Deep Learning* e *Data Science*, onde cada um desses tópicos estuda de forma mais aprofundada os sistemas de computação de interesse. A Figura 5 ilustra de como esses tópicos se inter-relacionam.

Figura 5 – Conceitos Inteligência artificial



Fonte: Página do site da empresa Oracle <sup>3</sup>

Na indústria, as tecnologias de inteligência artificial já vêm sendo aplicadas em diversas áreas, dentre alguns exemplos podemos citar:

- GPS: Utiliza IA para interpretar dados fornecidos pelos usuários e em algumas aplicações identificar rotas de trânsito ideal para cada usuário;

<sup>3</sup> Disponível em: <<https://www.oracle.com/br/artificial-intelligence/what-is-ai/>>. Acesso 26 mar. 2022.

- Jornalismo: Há estudos de programas que a partir dos dados fornecidos pelo usuário, o sistema escreve matérias jornalísticas de forma autônoma, onde o leitor não consegue distinguir o texto foi escrito pela máquina;
- Comercio online: Utilizam reconhecimento de padrões para identificar os interesses de cada usuário e oferecer produtos de acordo com suas preferências.

No estudo dessa pesquisa a IA funciona como um interpretador das perguntas feitas pelos usuários, visto que a inteligência artificial pode utilizar dados de áreas como psicologia e filosofia. A linguística ([TEIXEIRA, 2014](#)), que são estudos de fora do escopo da computação, mas que agregam os conceitos necessários para um bom funcionamento da IA, fornece subsídios teóricos para o processamento de linguagem natural.

### 3.7 Aprendizado de Máquina

O aprendizado de máquina, também conhecido como *Machine Learning*, é um campo de estudo da computação sobre inteligência artificial que evoluiu dos processos de aprendizado computacional e reconhecimento de padrões e pode ser definido como sendo algoritmos de predição inteligentes baseados em conjuntos de dados. Nos últimos anos, o estudo de aprendizado de máquina tem avançado consideravelmente, ao ponto que a interpretação da base de conhecimento, extração de dados, entendimento semântico e habilidade de definir padrões chega a ser mais acurada que de próprios humanos ([NICHOLS; CHAN; BAKER, 2019](#)).

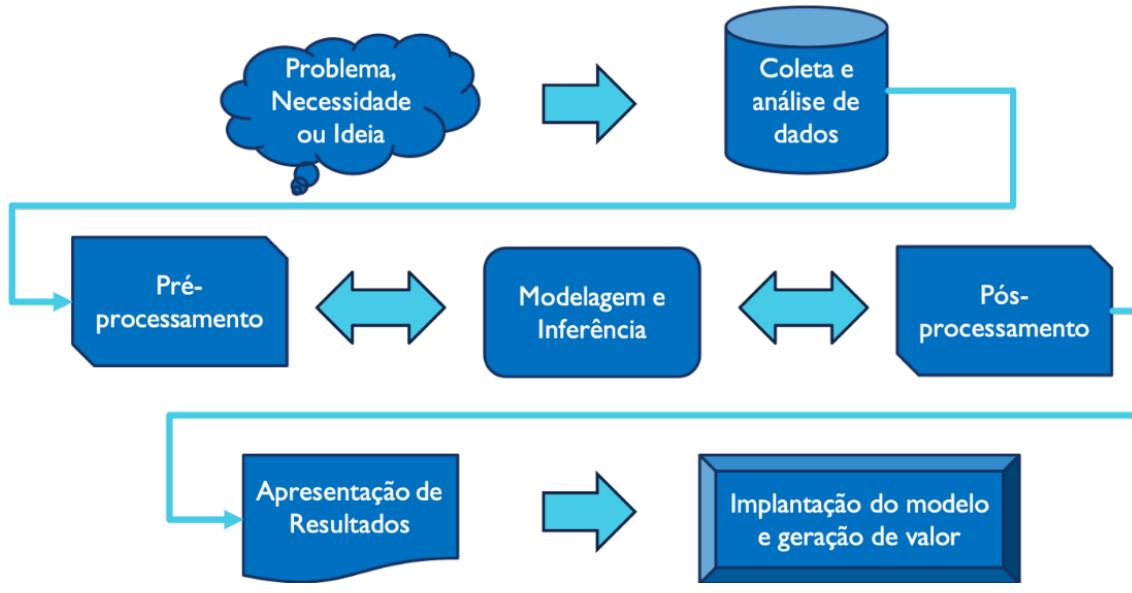
Dentre as principais técnicas de aprendizado de máquina, pode-se citar duas subdivisões. O aprendizado supervisionado, que faz fazer previsões de respostas com base em modelos montados por pares ordenados, ou seja, entrada e saída desejada, que nada mais são que uma amostragem das características e informações rotuladas de um determinado problema e a solução e saídas esperadas. Já o segundo tipo é conhecido como aprendizado não-supervisionado, conceito onde algoritmos realizam diversos testes de segmentação e análise de padrões para definir respostas de forma automática, sem um rotulo nem resposta preestabelecido, portanto, não temos a saída estimada. Além das técnicas de aprendizado, para aplicarmos os conhecimentos na prática existem sete etapas de desenvolvimento ([GONZÁLES, 2018](#)), como pode ser observado na Figura 6.

- Coleta de dados: após definir a aplicação, precisamos criar o *dataset*, que é a base de conhecimento que a nossa inteligência artificial usará como referência para fazer suas análises;
- Preparação: deve-se avaliar os dados coletados e verificar a amplitude das informações obtidas, se os dados forem muito uniformes ou tendenciosos isso irá afetar

diretamente os resultados de saída, sendo necessário realizar a normalização dos dados. Além disso, nesta etapa é feita divisão da nossa base de dados em duas amostras, uma será utilizada para o treinamento, e a outra avaliação;

- Definição dos modelos: o modelo escolhido dependerá diretamente do objetivo proposto do projeto, esse modelo deve responder qual padrão de dados resultarão na resposta esperada, os modelos mais utilizados são de classificação, agrupamento e regressão;
- Treinamento: o treinamento pode ser considerado como sendo um pilar fundamental para o aprendizado de máquina, pois além de preparar nossa inteligência, ele aprimora constantemente as habilidades de previsão, melhorando os resultados;
- Avaliação: irá constatar os resultados obtidos pelo treinamento, comparando parte dos dados separados previamente com a resposta do modelo treinado. Isso garante que máquina está aprendendo a estrutura do problema e não somente memorizando respostas anteriores;
- Aprimoramento dos parâmetros: após os resultados das etapas anteriores a reavaliação dos parâmetros é necessária para garantir a acuracidade do modelo e conseguir formas de melhorar o desempenho do aprendizado de máquina do modelo criado;
- Predição: a etapa de predição consiste na implementação da máquina em uma aplicação, para verificar de fato sua atuação e resolver o problema proposto anteriormente.

Figura 6 – Esquema projeto de aprendizado de máquina



Fonte: ([ESCOVEDO; KOSHIYAMA, 2020](#))

## 4 Metodologia

### 4.1 Pesquisa Exploratória

A pesquisa exploratória é um estudo que busca adquirir os conhecimentos necessários a investigação e proporciona uma visão ampla acerca do tema abordado. A partir das informações exploradas será possível definir as questões mais importantes sobre o tema, que orientarão o desenvolvimento do trabalho durante todas as etapas do estudo (RAUPP; BEUREN, 2006).

Sabendo disso, o presente trabalho realizou um levantamento sobre o cenário do mercado atual e foi comprovado que há um crescente aumento nos estudos e buscas relacionados a *chatbots* nos últimos anos. Tal incremento está se mostrando uma ferramenta muito promissora dentre as tecnologias utilizadas hoje em dia. Além disso, as maiores empresas de software estão apostando em soluções que compreendem a linguagem natural, sendo chamados de assistentes virtuais pessoais, alguns exemplos são a *Microsoft Cortana*, *Apple Siri*, *Google Assistant*, *IBM Watson* e a *Amazon Alexa* (ADAMOPOULOU; MOUSSIADES, 2020).

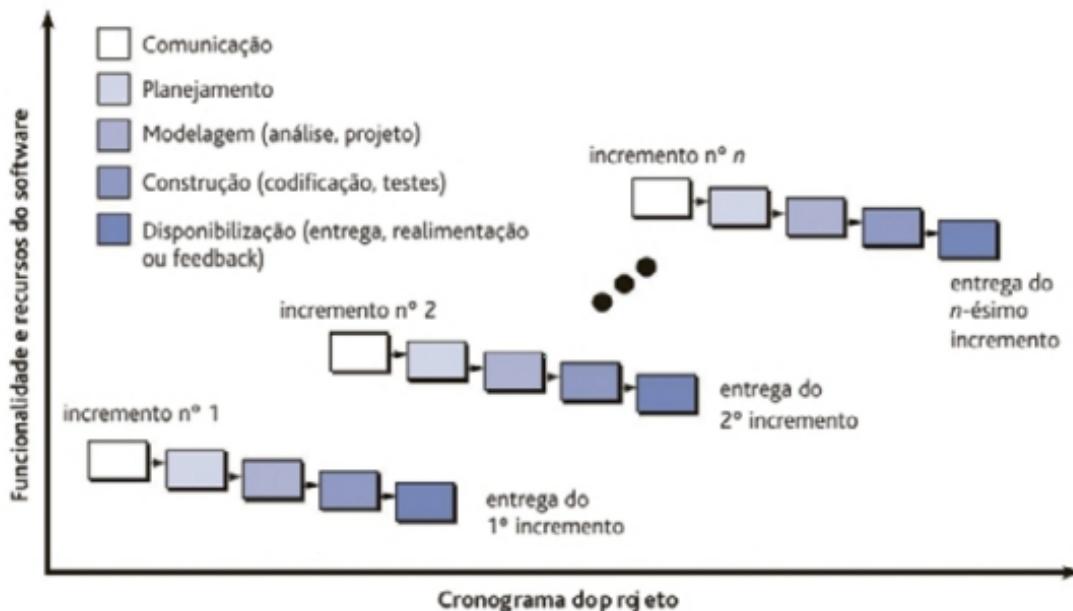
Buscamos também as ferramentas existentes no mercado para o desenvolvimento de um *chatbot*, foi utilizado o livro “*Hands-On Chatbots and Conversational UI Development*“ como referência, que proporcionou um entendimento dos conceitos, ferramentas e técnicas que são utilizadas para realizar o desenvolvimento dentre as plataformas apresentadas *Chatfuel*, *Dialogflow*, *Twillio*, *Alexa Skills* e *Microsoft Bot Framework*.

Dentre todas as ferramentas apresentadas, a escolhida para ser implementada no projeto é o *Dialogflow*, que é uma plataforma disponibilizada pela Google, de fácil integração com os bancos de dados, e de baixo custo. Com isso, pode-se dizer que a pesquisa exploratória, através da leitura de artigos e trabalhos do gênero, foi crucial para a definição das tecnologias mais relevantes para implementação do projeto.

### 4.2 Método Ágil de Desenvolvimento

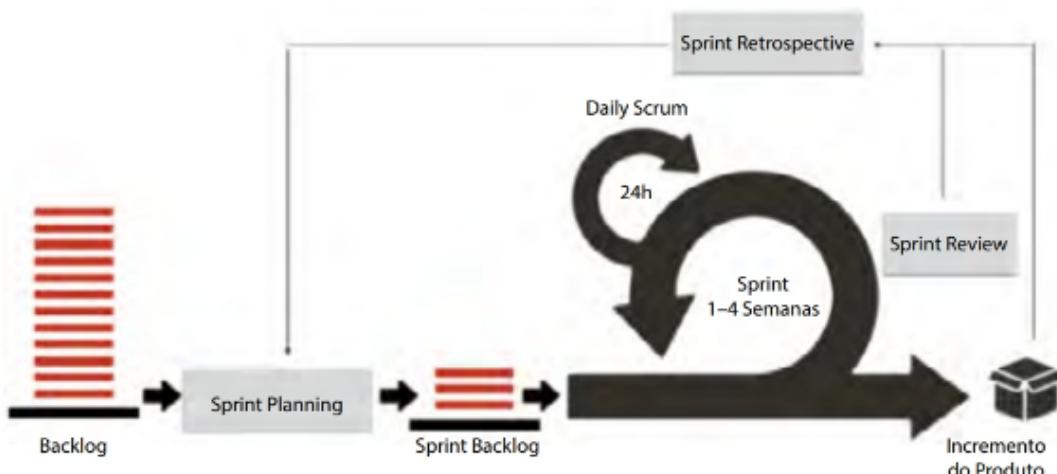
Para realizar o gerenciamento do projeto foi utilizado o *framework Scrum*, que é um dos métodos de desenvolvimento ágil mais utilizados na atualidade, dentre os seus princípios e valores está o processo de desenvolvimento incremental, que pode ser definido como uma série de funcionalidades que serão criadas dentro do projeto e desenvolvidas de forma gradual, onde cada nova função realizada agrupa valores e complementa o projeto, como pode ser analisado na Figura 8 (MASCHIETTO LUIS G.; MORAES, 2021).

Figura 7 – Processo de desenvolvimento incremental



Fonte: (ESCOVEDO; KOSHIYAMA, 2020)

O modelo *Scrum* baseia seus objetivos através de eventos, os quais tem o intuito de criar uma regularidade e previsibilidade. A primeira tarefa para se definir um projeto utilizando *Scrum* é a definição do *backlog*, que consiste em uma lista de todas as atividades necessárias para atingir a conclusão do projeto. Após essa etapa, as atividades são quebradas em entregas menores chamadas de *sprints*, que tem um tempo máximo de duração e um objetivo para ser alcançado, uma nova *sprint* se inicia assim que a anterior estiver finalizada (SUTHERLAND; SCHWABER, 2013). O desenvolvimento se dá de forma contínua até a finalização do projeto, o modelo do ciclo de desenvolvimento pode ser visualizado na Figura 8 abaixo.

Figura 8 – Ciclo da execução do *Scrum*

Fonte: (ESCOVEDO, 2020)

Considerando os princípios de organização do *Scrum* citados, o escopo geral do projeto foi definido em um *backlog* do produto, em seguida divido em ciclos de *sprints* que seguem as etapas de desenvolvimento de software. As etapas definidas para cada *sprint* estão listadas da seguinte forma:

- *Sprint 1* – Busca de referência;
- *Sprint 2* – Levantamento dos requisitos de software;
- *Sprint 3* – Prototipagem de software;
- *Sprint 4* – Definição das ferramentas a serem utilizadas;
- *Sprint 5* – Desenvolvimento da iteração entre as ferramentas;
- *Sprint 6* – Desenvolvimento do projeto simplificado;
- *Sprint 7* – Desenvolvimento e comunicação com o banco de dados;
- *Sprint 8* – Conclusão.

# 5 Desenvolvimento da Solução

## 5.1 Levantamento de Requisitos de Software

O levantamento de requisitos é o primeiro e um dos principais passos no desenvolvimento de software, é nesta etapa que será abstraído da solução idealizada as funções que deverão ser implementadas no sistema, cada um dos requisitos de software representa uma propriedade ou característica que é esperada para satisfazer a ideia inicial do projeto. O produto final desse levantamento será uma lista de funcionalidades que serão utilizadas pelo time de desenvolvimento no decorrer de sua criação. Por isso, o levantamento deve ter o envolvimento de todas as partes interessadas no projeto e deve tentar abranger o máximo cenários possíveis, para compreender corretamente cada proposta e evitar erros durante o desenvolvimento da solução.

Os requisitos do projeto estudado foram definidos com o intuito de estabelecer as funções de um *chatbot* completo e o método de levantamento utilizado para definir os requisitos foi o *In Loco*, expressão do latim que significa “no próprio lugar”, que é uma técnica que consiste em analisar o cenário real de um ambiente em que o sistema será utilizado e identificar os processos que sistema deverá possuir para solucionar o problema apresentado.

## 5.2 Requisitos Funcionais

Os requisitos funcionais representam funcionalidades que definem como o sistema deve funcionar e se comportar em determinadas situações. Os requisitos funcionais estão sumarizados na Tabela 2.

Tabela 2 – Empresa que venderam pela internet nos últimos 12 meses, por tipo de canal online em que ocorreu a venda

Identificação	Requisito	Descrição
RF001	Consultar Categoria Produto	O sistema deverá listar todos os produtos, dividindo-os em categorias para melhorar visualização do usuário.
RF002	Consultar Informações Produto	O usuário poderá consultar cada produto individualmente, e após selecionar o item ele deve mostrar informações do mesmo como ingrediente, valor etc.
RF003	Adicionar no Carrinho	O usuário poderá adicionar cada produto visualizado em uma lista de produtos.
RF004	Comprar vários produtos	O usuário poderá visualizar os produtos adicionados na lista e realizar a compra dos produtos em um único pedido.
RF005	Informações de compra	Ao realizar a compra o usuário deverá informar os dados necessários para concluir o pedido, como endereço, tipo de pagamento etc.
RF006	Visualizar status do pedido	O usuário poderá visualizar as compras realizadas e as informações referentes aos pedidos como status, endereço de entrega, tipo de pagamento etc.
RF007	Salvar dados do usuário	O sistema deverá salvar automaticamente as informações do usuário para que nos próximos acessos não seja necessário preencher os dados novamente.

### 5.3 Regra de Negócio

As regras de negócio levantadas pela metodologia estão descritas na Tabela 3.

Tabela 3 – Levantamento de regra de negócio

Identificação	Requisito	Descrição
RN001	Utilizar o Whatsapp	O sistema deverá receber e interagir com o aplicativo de mensagens do Whatsapp.
RN002	Salvar informações no banco de dados	O sistema deverá salvar as informações essenciais no banco de dados para poder consultar e inserir dados sem precisar alterar o código.

## 5.4 Requisitos Não-funcionais

Os requisitos não funcionais são as características específicas que o sistema deve possuir. Os requisitos não funcionais estão sumarizados na Tabela 4.

Tabela 4 – Levantamento de requisitos não funcionais

Identificação	Requisito	Descrição
RNF001	Reconhecer perguntas erradas	O sistema deverá reconhecer as perguntas que não correspondem a nenhuma função e deverá usar uma resposta pré-definida dependendo do contexto do diálogo.
RNF002	Entender linguagem	O sistema deverá entender a linguagem utilizada pelo usuário e reconhecer sinônimos e linguajar do usuário.
RNF003	Responsividade	O sistema deverá ser responsivo, ou seja capacidade de responder rapidamente e adequadamente as perguntas solicitadas.
RNF004	Disponibilidade e Velocidade	O sistema deverá realizar suas resposta de forma rápida e ter alta disponibilidade para uma grande quantidade de usuários ao mesmo tempo.

## 5.5 Caso de Uso

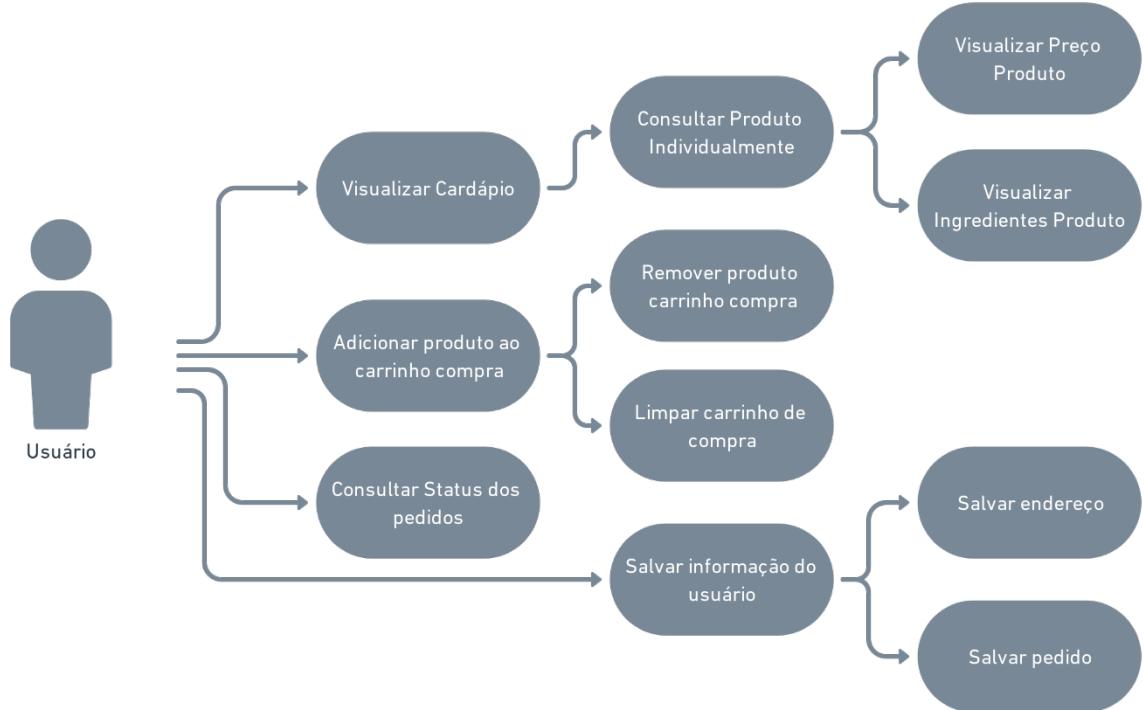
O diagrama de caso de uso, é uma representação visual que resume detalhadamente as interações e ações de cada um dos usuários em um sistema, nesse tipo de modelo, cada usuário é denominado um ator diferente, pois cada um terá uma representação diferente, no sistema estudado foi definido dois atores, o Usuário e o Gestor, e cada modelo pode ser analisado abaixo:

**O ator Usuário:** representa todo e qualquer cliente que utilizará a solução proposta e irá interagir com nosso *chatbot*, pode-se dizer que dependendo da aplicação representa a parte principal do projeto, pois caso o sistema seja difícil de utilizar ou apresente falhas que impeçam a usabilidade, o cliente poderá se recusar a utilizar a plataforma e até mesmo perder uma venda.

As ações do Usuário dentro do sistema são muito próximas as de uma plataforma de *ecommerce*, tentando replicar as mesmas funcionalidades utilizando somente as interações através de mensagens de texto com o *chatbot*. Considerando esse cenários, as funções desse ator se resumem a funções de consulta, como visualizar os produtos disponíveis, consultar ingredientes dos produtos, consultar pedidos realizados e o status do pedido, e as funções de compra que salvam os dados escolhidos pelo cliente na base de dados, que são: realizar

pedido, salvar produto carrinho e salvar endereço. Na Figura 9 temos o diagrama do caso de uso do usuário.

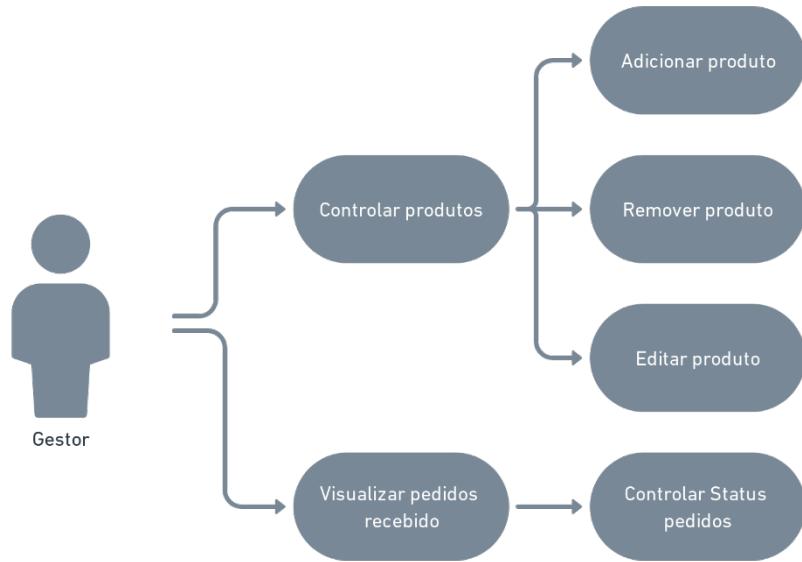
Figura 9 – Caso de uso: usuário



Fonte: Elaborado pelo autor

**Gestor:** o papel deste ator é de controlar e gerenciar as informações do sistema, no nosso aplicativo esse ator terá somente acesso a base de dados, onde são salvos os produtos disponibilizados pela empresa e onde fica armazenada as informações dos clientes e pedidos que são feitos através do *chatbot*. Esse gerenciamento poderia ser feito através de uma aplicação de controle, o que facilitaria a manutenção e traria uma independência e autonomia para realizar as mudanças sem a necessidade de alterar as informações diretamente no banco de dados. Como o desenvolvimento dessa solução não está no escopo do projeto estudado, essas funções não foram implementadas. Na Figura 10 o caso de uso do gestor da aplicação.

Figura 10 – Caso de uso: gestor



Fonte: Elaborado pelo autor

## 5.6 MoSCoW

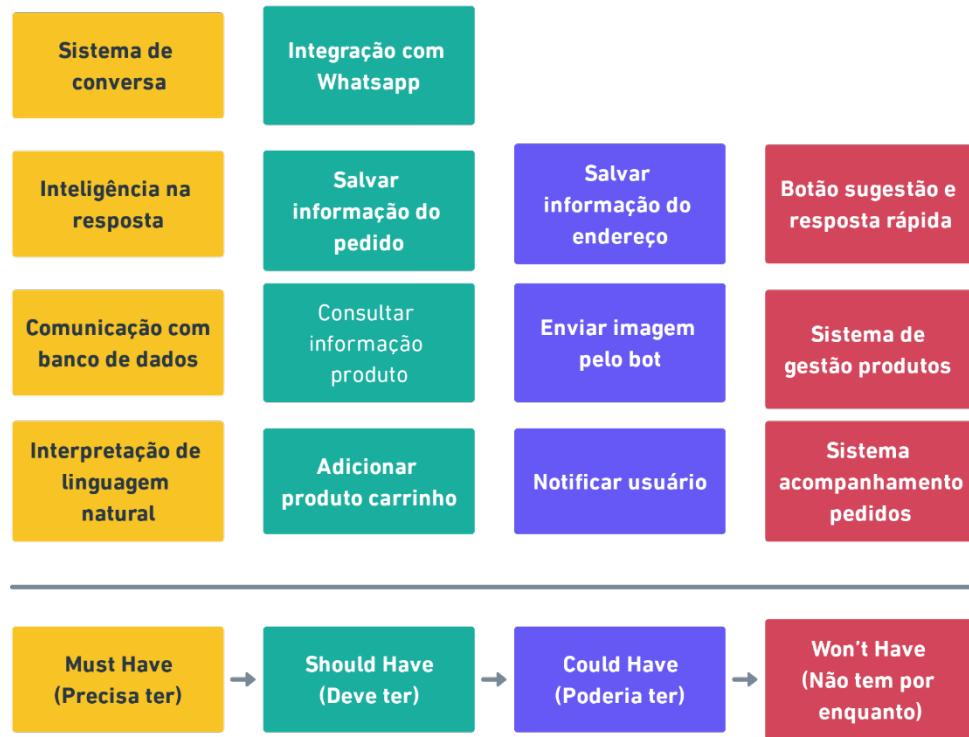
O método *MoSCoW* de priorização de projeto é um modelo desenvolvido por Dai Clegg enquanto trabalhava para a *Oracle*, seu objetivo era criar um *framework* que ajudasse a priorizar as tarefas de desenvolvimento de forma organizada e controlar a expectativa esperado do projeto desenvolvido.

O significado do acrônimo da palavra *MoSCoW* representa quatro categorias de funcionalidades de um projeto:

- Precisa ter (*Must have*), função mandatória para o projeto, pois o mesmo não funcionará caso não possua essas ferramentas;
- Deve ter (*Should have*), partes importantes, mas que não são vitais para o projeto, porém adicionam um valor significativo para o contexto geral/
- Poderia ter (*Could have*), funções que seriam um diferencial, mas não trazem um impacto tão grande para o projeto;
- Não tem por enquanto (*Won't have*), são ferramentas que não são prioridade para o desenvolvimento atual do projeto.

A partir dessa classificação podemos criar um quadro exemplificando as principais funções, dispostas de acordo com suas categorias. Para o projeto desenvolvido, o modelo pode ser visto através da Figura 11:

Figura 11 – Quadro de prioridade método MoSCoW



Fonte: Elaborado pelo autor

## 5.7 Prototipagem de Software

A prototipagem é a etapa onde é estabelecido um modelo de referência do produto, mostrando um esboço para guiar o desenvolvimento do projeto. Para realizar esse protótipo foi utilizado a ideia de textitwireframe, que se resume em um modelo da interface capaz de definir e seu layout e elementos principais. Na Figura 12 temos o protótipo do inicio da conversa.

Figura 12 – Wireframe do produto, parte 1



Fonte: Elaborado pelo autor

A importância do desenvolvimento de um protótipo do produto é conseguir abstrair através de uma representação visual um modelo do projeto, nesse estudo foi utilizado o *WhatsApp* como interface *frontend*, devido a isso, o protótipo focou em simular uma conversa completa do cliente utilizando a plataforma e resumindo a parte gráfica e visual a um modelo básico de aplicativo de mensagem. Na Figura 13 temos a segunda parte do protótipo.

Figura 13 – Wireframe do produto, parte 2



Fonte: Elaborado pelo autor

## 5.8 Ferramentas Utilizadas

Depois de definir as necessidades da aplicação e conhecendo as melhores ferramentas para o desenvolvimento, é preciso construir a comunicação entre cada serviço que esta investigação empregou.

Como forma representativa de cada serviço utilizado e a comunicação realizada entre cada um deles, foi estruturado um modelo MVC do projeto, que é um padrão de arquitetura de software que é dividida em três componentes essenciais: *Model*, *View* e *Controller*. A camada *View* gerencia toda parte gráfica e textual da aplicação e é o único componente que o usuário tem contato. A *Controller* interpreta toda iteração realizada com a *View*, coordenando toda ação entre a *Model* e a *View* apropriadamente. Por fim, a *Model* gerencia o comportamento e os dados da aplicação (BURBECK, 1992). Este conceito está representado na Figura 15.

Figura 14 – MVC (Model View Control)

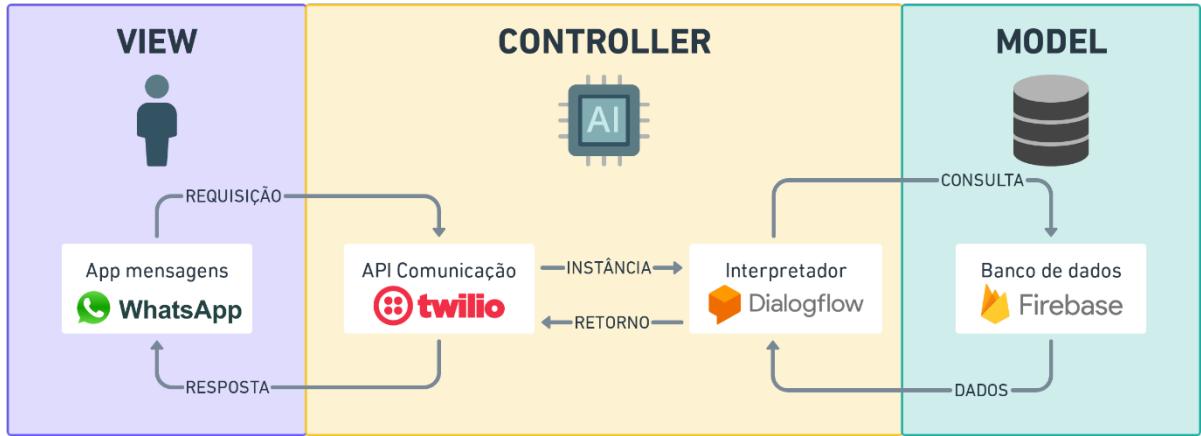


Figura 15 – Fonte: Elaborado pelo autor

Na aplicação estudada, a comunicação e interação com o usuário será feito pelo aplicativo do *WhatsApp*, portanto ele estará na camada de *View*. Na *Controller* teremos as duas aplicações responsáveis por retornar as respostas ao usuário, a primeira é a *API* do *WhatsApp*, utilizando a solução fornecida pela *Twilio*, essa ferramenta recebe todas as mensagens que o usuário envia e comunica com a segunda solução, o *Dialogflow*, que interpretará de forma inteligente as frases, linguagem e contexto das mensagens recebidas e retornará a melhor resposta possível para cada iteração. Dependendo da mensagem, se houver a necessidade de interação com o banco de dados, seja para consultar ou guardar uma informação, o *Dialogflow* se comunicará diretamente com *Firebase Realtime Database* concluindo essa a camada *Model* da solução.

## 5.9 Configuração da API

A *Twilio* oferece a opção de criar um ambiente de teste para comunicação com o *WhatsApp*, para isso foi criado um projeto dentro da plataforma disponibilizada por eles e nela foi configurado um serviço, que receberá a comunicação e respostas feitas pelo *Dialogflow*. Para isso, é necessário configurar as chaves e permissões de iteração entre as duas plataformas, deste modo as mensagens recebidas já serão redirecionadas para a aplicação da Google. Além disso, é preciso configurar uma função dentro da plataforma da *Twilio* que receberá o modelo padrão de resposta em *JSON* e converterá para uma tipagem de mensagens do *WhatsApp*. Na Figura 16 é possível verificar o código referente a comunicação da *Twilio* com o *Dialogflow* (BARNEY, 2021).

Figura 16 – Função de decodificação

```

const dialogflow = require("dialogflow");

exports.handler = async function (context, event, callback) {

    let twiml = new Twilio.twiml.MessagingResponse();
    const receivedMsg = event.Body; const userNumber = event.From;

    // Recebe a resposta do Dialogflow
    let dialogflowJsonFilePath = Runtime.getAssets()["/dialogflow.json"].path;
    const dialogflowSessionClient = new dialogflow.SessionsClient({
        keyFilename: dialogflowJsonFilePath,
    });
    const dialogflowProjectId = process.env.DIALOGFLOW_PROJECT_ID;
    const dialogflowSessionPath = dialogflowSessionClient.sessionPath(
        dialogflowProjectId, userNumber
    );
    const dialogflowRequestParams = {
        session: dialogflowSessionPath,
        queryInput: {
            text: {
                text: receivedMsg, languageCode: "pt-BR",
            },
        },
    };
    const dialogflowRequest = await dialogflowSessionClient.detectIntent(dialogflowRequestParams);
    const dialogflowResponses = dialogflowRequest[0].queryResult.fulfillmentMessages;

    // Faz a iteração de cada mensagem recebida
    for (const response of dialogflowResponses) {
        // Textos
        if (response.text) { const text = response.text.text[0]; twiml.message(text); }
        // Payloads
        if (response.payload) {
            const fields = response.payload.fields; const payloadMessage = await twiml.message();
            if (fields.mediaUrl) { const mediaUrl = fields.mediaUrl.stringValue; payloadMessage.media(mediaUrl); }
            if (fields.text) { const text = fields.text.stringValue; payloadMessage.body(text); }
        }
    }
    return callback(null, twiml);
};

```

Fonte: *Github* do autor <sup>1</sup>

Com essa etapa concluída temos o ambiente configurado, que envia as mensagens recebidas para a aplicação do *chatbot* e recebe e interpreta as respostas para serem mostradas de forma correta para o usuário, assim podemos realizar o desenvolvimento das interações que desejamos que a solução responda.

<sup>1</sup> Disponível em: <<https://github.com/MichaelBarney/Dialogflow-Twilio-Whatsapp-Conector>>. Acesso 23 out. 2022.

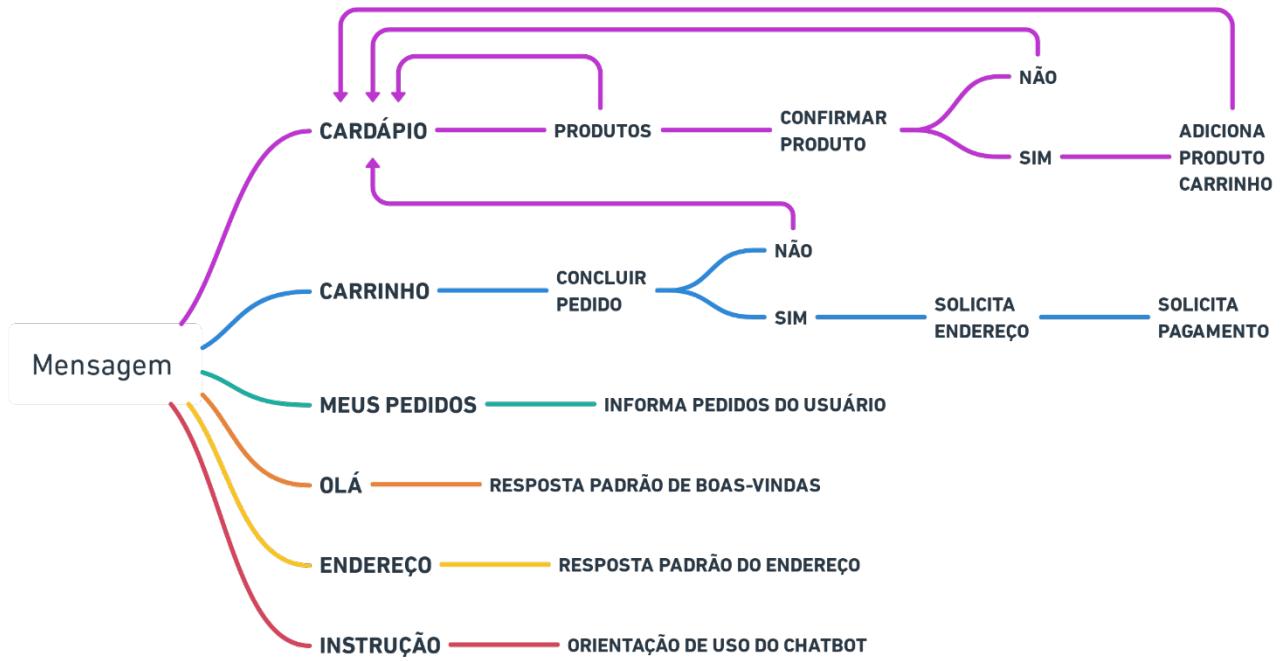
Figura 17 – Função de decodificação

```
{
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          "Olá, como você está?"
        ]
      }
    },
    {
      "text": {
        "text": [
          "\nPrazer somos a Lanchonete DEV 🍔, e estamos aqui para atender seu pedido"
        ]
      }
    },
    {
      "text": {
        "text": [
          "\n\nPara conhecer nossos produtos me peça o \"Cardápio\""
        ]
      }
    }
  ],
  "outputContexts": []
}
```

Fonte: Elaborado pelo autor

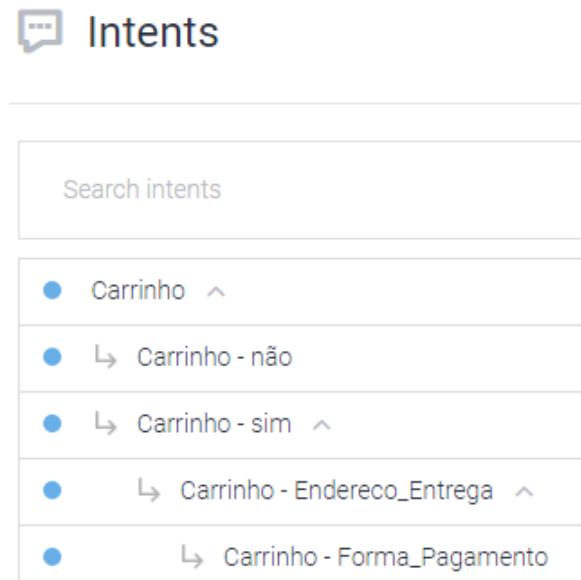
## 5.10 Criação do *Chatbot*

O *chatbot* em si é a inteligência que realizará a interpretação de perguntas feitas a um computador e responderá essas mensagens criando uma conversação entre máquina e ser humano. Em nossa aplicação a ferramenta responsável por essa interação é o *Dialogflow*. Como explicado anteriormente esse software trabalha utilizando *intents* e cada uma representa uma resposta diferente que o *chatbot* pode fornecer, dependendo da mensagem de entrada recebida. Para isso, foi desenvolvido um diagrama de mapa mental para ajudar na orientação sobre a construção das respostas do *chatbot*, como pode ser visto na Figura 18, onde cada linha colorida representa uma *intent* de entrada e o contexto da conversa com o usuário, cada junção representa uma resposta possível que o *chatbot* pode oferecer. Como pode ser observado, dependendo do andamento da conversa pode-se avançar ou retroceder em suas repostas, o objetivo é construir uma conversação mais dinâmica e próxima de uma pessoa real. O *mindmap* desenvolvido pode ser visualizado na Figura 18.

Figura 18 – Mind map estrutura do *chatbot*

Fonte: Elaborado pelo autor

Dentro da plataforma do *Dialogflow* é construído uma *intents* para cada resposta, como demonstrado a estrutura de exemplo das mensagens que listam os produtos no carrinho. Na Figura 19 pode-se observar a árvore de *intents* e como cada uma está diretamente relacionada a anterior.

Figura 19 – *Intent* utilizadas na função do Carrinho

Fonte: Elaborado pelo autor

Pode-se analisar a estrutura interna de cada uma das *intents* na Figura 20:

Figura 20 – Exemplo da construção de cada *Intent*

The screenshot shows the Dialogflow interface for creating an intent named 'Carrinho'. The top bar has a 'SAVE' button. Below it, the 'Contexts' section shows an input context 'Add input context' and an output context '2 Carrinho-followup Add output context'. The 'Events' section shows an event 'carrinho'. The 'Training phrases' section contains a search bar and a list of user expressions: 'meus itens', 'carrinho compra', 'meus pedidos', 'lista itens', and 'lista pedidos'.

Fonte: Elaborado pelo autor

- No primeiro campo temos a construção do contexto, que são como variáveis que dão sentido à conversação, em cada resposta temos dois tipo diferentes, o de entrada e o de saída, o contexto de entrada é um filtro que controla a execução da *intent*, que só ocorrerá caso o contexto de entrada exista, portanto ele define se houve um diálogo anterior que de sentido á execução dessa resposta. O contexto de saída por sua vez cria um novo contexto a conversa, permitindo a execução de novas *intents* que estejam relacionadas a está, ela também define o tempo de vida daquele contexto, que consistem em quantas respostas a partir daquele continuarão tendo o mesmo contexto na conversa. É possível ter mais de um contexto de entrada ou de saída em cada resposta.
- A segunda parte é a criação do evento, que nada mais é que a definição do nome da função que será executada pelo nosso *webhook* quando aquela *intent* for executada.

- Em terceiro, temos as frases de treinamento, essas são as palavras e expressões que serão o gatilho para a execução dessa resposta, a mensagem do usuário não precisa necessariamente corresponder exatamente a frase de treinamento da *intent*, pois o *Dialogflow* consegue abstrair a partir da pergunta do usuário semelhanças que fazem sentido as frases treinadas.

Definido a estrutura base de uma *intent* precisamos definir a função de execução dessa resposta, portanto dentro da própria plataforma do *Dialogflow* utilizamos a ferramenta de *fulfillment* que permitirá a abstração em código *Javascript* a execução da resposta, na Figura 21 temos o código que representa a função de listar os itens no carrinho do usuário.

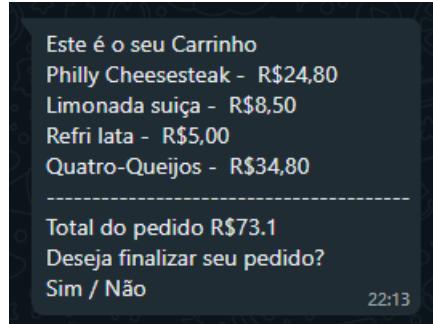
Figura 21 – Código que informa os produtos no carrinho do cliente

```
function getcarrinho(agent) {
  if(_Carrinho.length > 0){
    _Caminho = "";
    let resposta = `Este é o seu Carrinho`;
    _Total_Pedido = 0.0;
    for (const element of _Carrinho) {
      _Total_Pedido = Number(_Total_Pedido) + Number(element.Valor.replace(",","."));
      resposta = resposta + `\n` + element.Categoria+` - `+element.Produto+` R$`+element.Valor;
    }
    resposta = resposta + `\n-----\nTotal do pedido R$`+_Total_Pedido;
    resposta = resposta + `\nDeseja finalizar seu pedido?\nSim / Não`;
    agent.add(resposta);
  }
  else{
    agent.add(`Ainda não há nenhum item no seu carrinho.`);
  }
}
```

Fonte: Elaborado pelo autor

A partir da conversa estabelecida com o usuário, ele escolherá os produtos que deseja adquirir, portanto temos um objeto "*\_Carrinho*" que possui o nome e valor de cada item selecionado, após essa etapa verificamos se ele possui itens ou não, caso ele tenha itens na sua lista de produtos é executa um laço de repetição que consultará cada um dos itens, calculando a soma o valor dos produtos e a descrição de cada um. Finalmente, temos uma mensagem de retorno para o usuário, da forma exemplificada na Figura 22.

Figura 22 – Exemplo do mensagem mostrando os produtos no carrinho



Fonte: Elaborado pelo autor

## 5.11 Banco de Dados

Como apresentado no mapa mental de iterações do *chatbot*, o conjunto de resposta que mostra o cardápio da lanchonete deve ser mostrado no formato de categorias e, conforme o usuário seleciona cada um dos itens, ele irá construir a conversa até escolher o produto desejado. Para tal, é construído uma espécie de laço, onde a partir de cada item do cardápio pode-se avançar ou retroceder em seu estado, como mostrado no exemplo abaixo 23.

Figura 23 – Exemplo de iteração mostrando as categorias e produtos



Fonte: Elaborado pelo autor

Para realizar esse tipo de construção, precisamos primeiro salvar a lista de produtos em uma base de dados, que será consultada dependendo da posição atual do usuário na conversa. Primeiro, temos a lista genérica de categorias, em seguida as opções disponíveis para cada categoria e depois a informação final de ingrediente e valor do produto. Abaixo temos o exemplo do arquivo *JSON* com os dados cadastrais de uma lista de produtos. Na Figura 24 temos um exemplo de como foi construído o arquivo dos produtos.

Figura 24 – Exemplo das informações dos produtos no banco de dados



```
JSON ▾
{
  "Cardapio": {
    "Pizza": [
      {"Pizza Grande": [
        {"Quatro-Queijos": [
          {"Ingrediente": "Mussarela, Parmesão, Catupiry, Prato",
           "Valor": "45,99"}
        ]},
        {"Pizza Média": [
          {"Mussarela": [
            {"Ingrediente": "mussarela, rodelas de tomate e orégano",
             "Valor": "32,99"}
          ]},
          {"Sanduíches": [
            {"Philly Cheesesteak": [
              {"Ingrediente": "Queijo Prato, Mussarela, Fatias Carne Bovina, Pimentão, Cebola",
               "Valor": "24,80"}]}]}]}]}]
```

Fonte: Elaborado pelo autor

Com essa estrutura foi possível percorrer a lista dos produtos de forma dinâmica. Para buscar as informações pelo *Firebase Database* utilizou-se o formato de consulta por *DataSnapshot*, que é uma função nativa do serviço empregado para consultar uma cópia dos dados salvos no banco. E adotando esse tipo de estrutura é possível empregar o nome de cada categoria para consultar seus filhos e, a partir deles, consultar seus dados interno, como no exemplo da Figura 25, onde é possível consultar e listar todos os itens através do caminho Cardápio/Pizza/Pizza Grande.

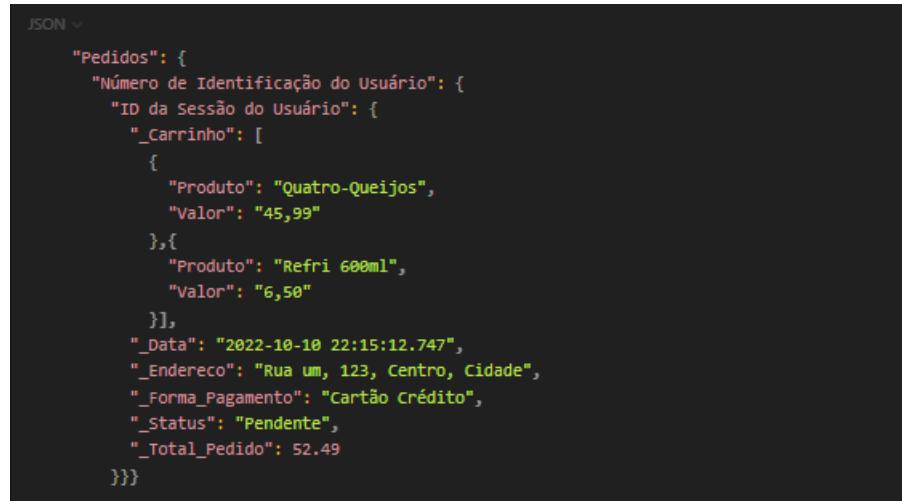
Figura 25 – Código que realiza a consulta no *Firebase*

```
let resposta = "Bom dia, Hoje temos disponível cardápio de:\n\n";
return firebaseAdmin.database()
.ref('Cardapio/Pizza/Pizza Grande')
.once('value')
.then((snapshot) => {
    let count = 1;
    snapshot.forEach(child => {
        resposta = resposta + count + " - " + child.key + "\n";
        Categorias.push([
            "index": count,
            "valor": child.key
        ]);
        count = count + 1;
    });
    agent.add(resposta);
}).catch(
    error => {
        agent.add("Erro ao retornar cardápio 😞 " + error);
});
}
```

Fonte: Elaborado pelo autor

Da mesma forma apresentada anteriormente, o *Firebase* possibilita inserir dados em seu banco, sendo possível incluir toda a estrutura de um objeto que foi instanciado dentro do código do sistema. Utilizando essa função, salvamos os dados de cada pedido dentro do banco, para cada usuário iremos armazenar o número de identificação do cliente e cada pedido realizado por ele terá um número de sessão único, onde as informações dos produtos que ele adicionou ao carrinho são armazenadas. Além disso, será salvo as informações pessoais como data, endereço, forma de pagamento, status do pedido e valor total da soma de todos os produtos, como pode ser visto na imagem abaixo [26](#).

Figura 26 – Exemplo das informações dos pedido no banco de dados



The screenshot shows a JSON editor interface with a dark theme. At the top left, there is a dropdown menu labeled "JSON". The main area displays a single JSON object representing an order. The object has the following structure:

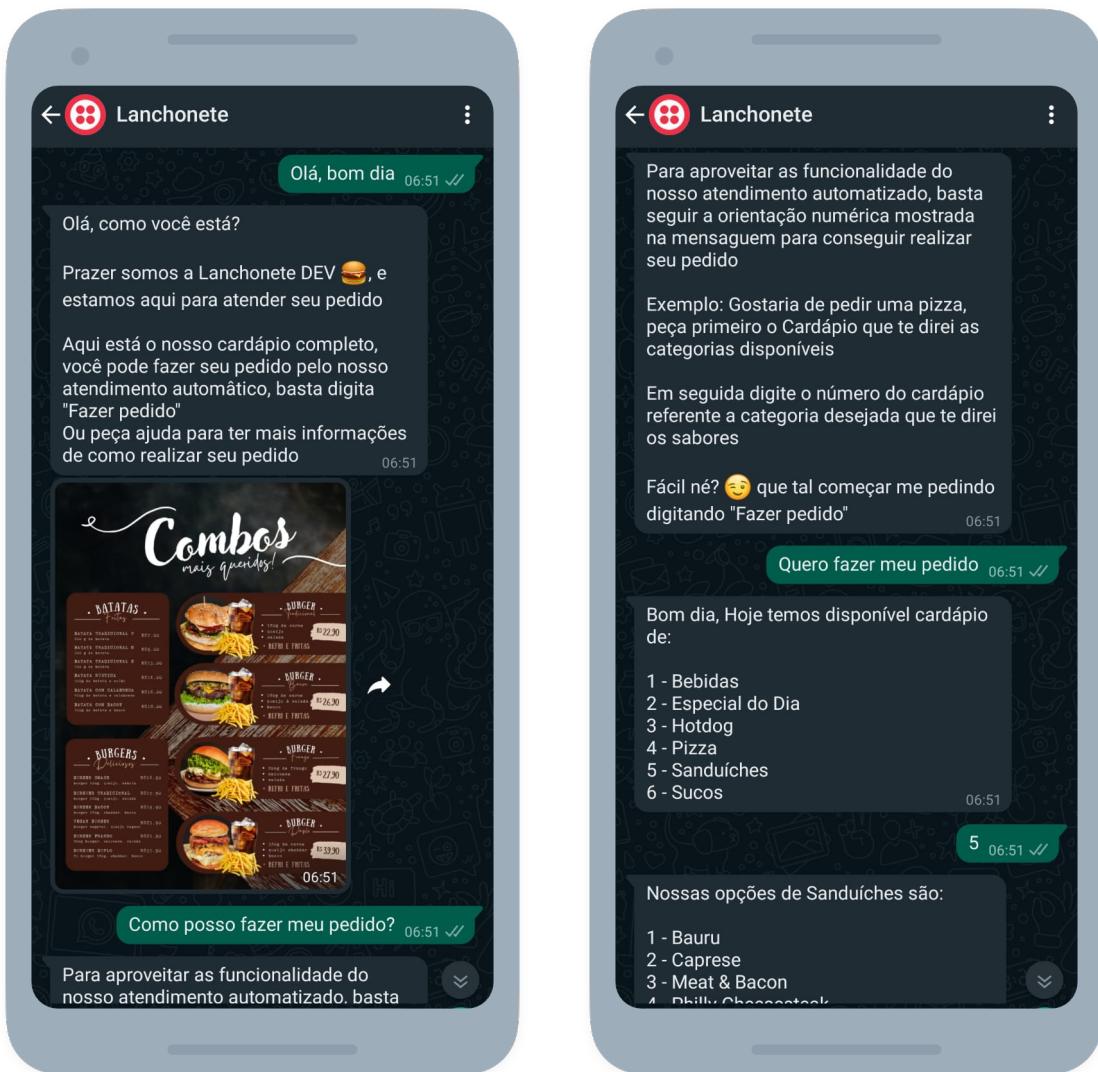
```
JSON
{
  "Pedidos": [
    {
      "Número de Identificação do Usuário": {
        "ID da Sessão do Usuário": [
          {
            "_Carrinho": [
              {
                "Produto": "Quatro-Queijos",
                "Valor": "45,99"
              },
              {
                "Produto": "Refri 600ml",
                "Valor": "6,50"
              }
            ],
            "_Data": "2022-10-10 22:15:12.747",
            "_Endereço": "Rua um, 123, Centro, Cidade",
            "_Forma_Pagamento": "Cartão Crédito",
            "_Status": "Pendente",
            "_Total_Pedido": 52.49
          }
        ]
      }
    }
  ]
}
```

Fonte: Elaborado pelo autor

## 6 Resultados

A partir dos conhecimentos adquirido no decorrer da investigação, foi possível criar um sistema de *chatbot* para atendimento de uma lanchonete, focando na construção de uma versão resumida e simplificada de um software que pode ser comercializado. Buscou-se implementar cada uma das necessidades básicas do sistema, como consulta e inserção no banco de dados e a utilização das funções disponibilizadas pelo *Dialogflow*, como o envio de imagens e uso de contexto, sempre tentando obedecer aos requisitos de software e protótipo desenvolvido anteriormente.

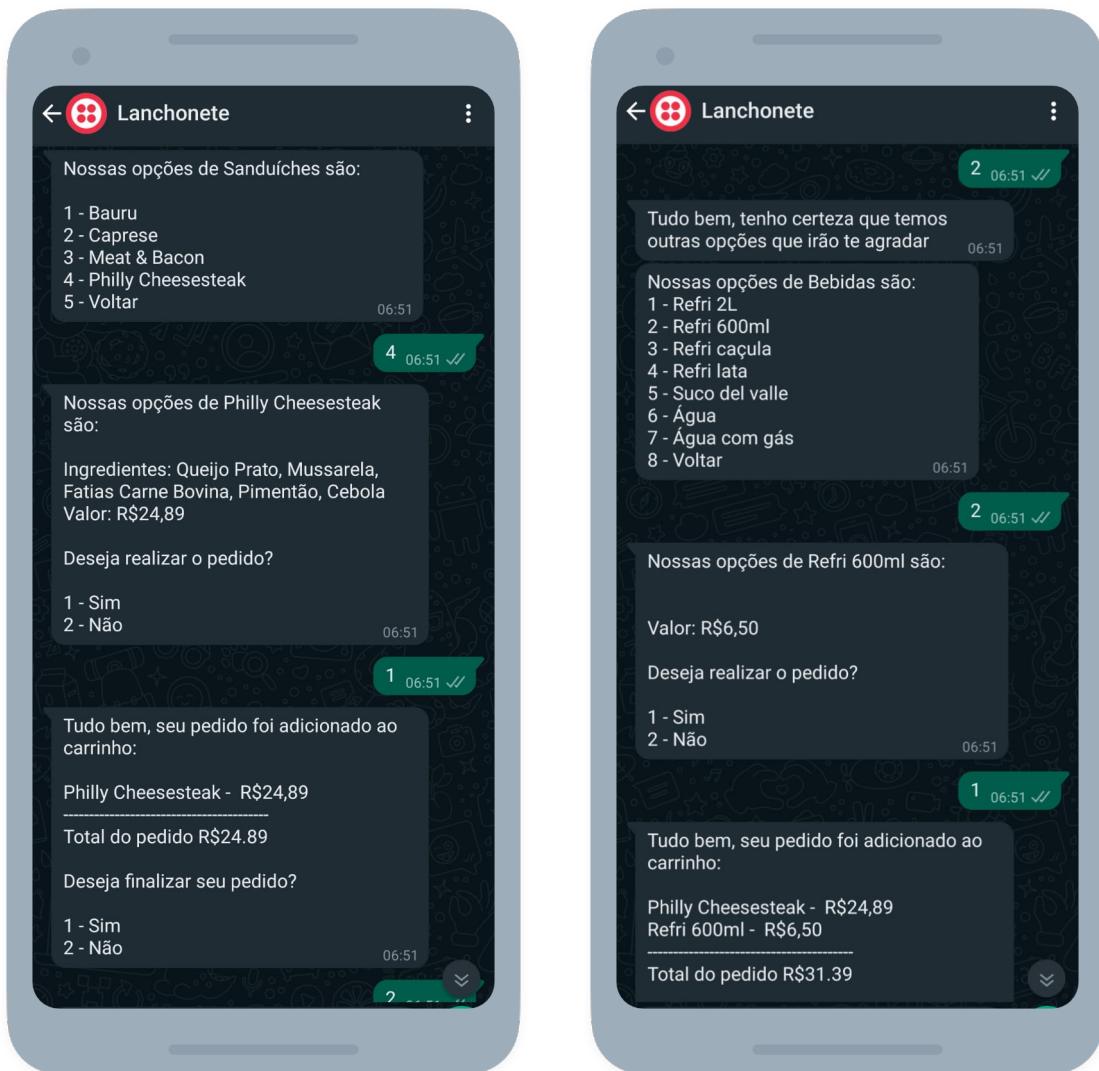
Figura 27 – Resultado obtido, 1



Fonte: Elaborado pelo autor

Para apresentar os resultados obtidos com o desenvolvimento do projeto foi construído um exemplo de conversação completa, desde a primeira iteração do usuário até a finalização do pedido, que podem ser visualizadas pela sequências de Figuras que sucedem esse capítulo. Na Figura 27 observa-se que o sistema foi desenvolvido para responder ao usuário de uma forma informal, com o objetivo de deixar o cliente mais confortável ao se comunicar com robô.

Figura 28 – Resultado obtido, 2

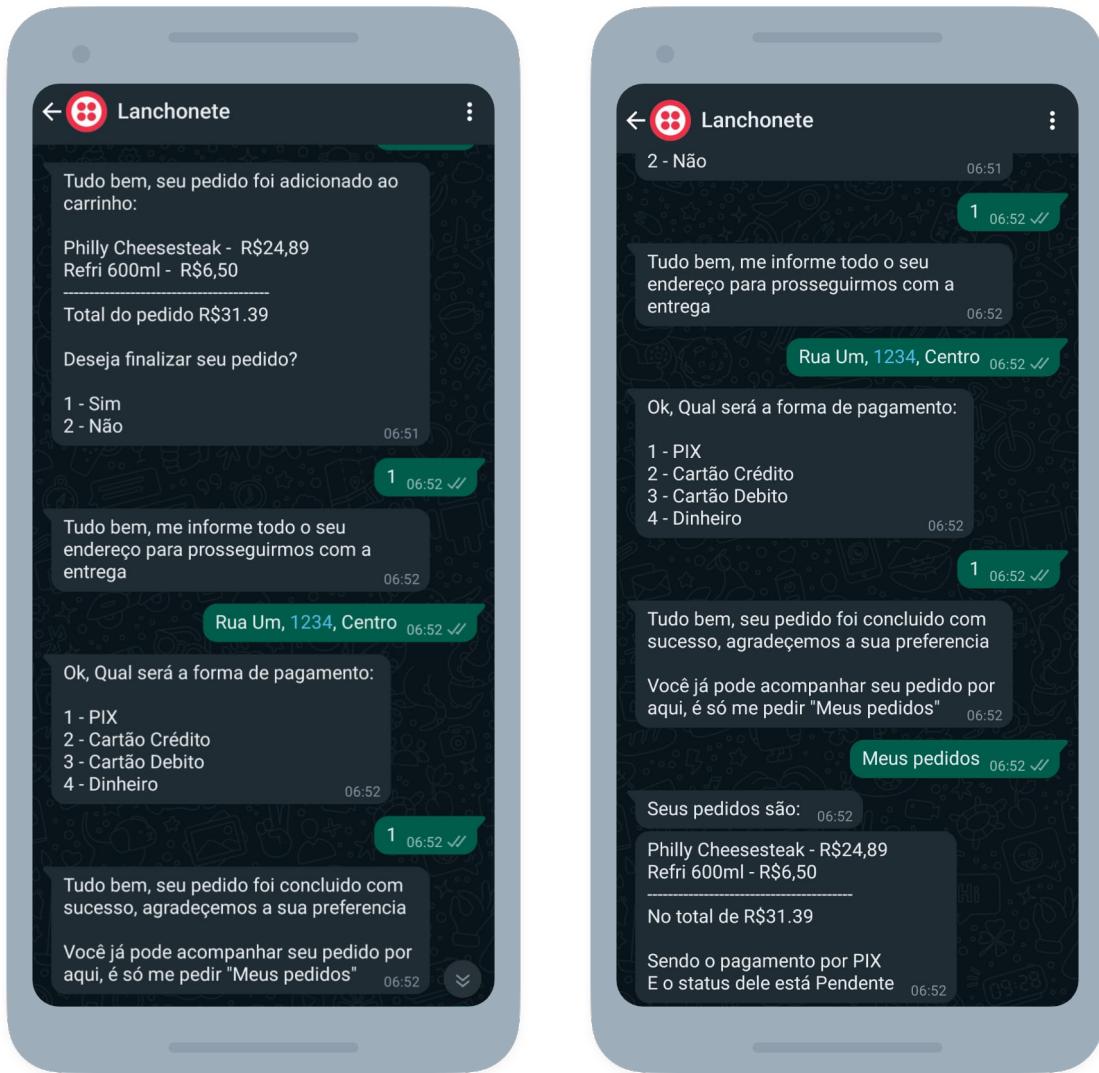


Fonte: Elaborado pelo autor

O sistema, além de orientar como deve ser feita a iteração entre o usuário e o *chatbot*, tenta ser o mais didático e amigável possível, orientando o cliente a utilizar as opções numéricas em vez de ter que digitar cada produto que deseja adquirir, além de proporcionar uma navegação dinâmica entre as opções e possibilitar avançar ou retroceder entre as categorias. Tal exemplo pode ser visto na Figura 28.

Em seguida, pode-se observar na Figura 29 algumas das funcionalidades da solução como a opção de adicionar os produtos no carrinho, que ficam salvo durante a conversa do usuário com o *bot*, a opção de concluir o pedido, que solicita ao cliente algumas informações como endereço de entrega e forma de pagamento, além da função capaz de consultar os pedidos realizados e acompanhar o estado do mesmo.

Figura 29 – Resultado obtido, 3



Fonte: Elaborado pelo autor

## 6.1 Discussão

Considerando que a presente investigação apresentou resultados satisfatórios e que existe uma demanda no mercado por esse tipo de solução, pode-se pensar em estender o protótipo para um produto dinâmico e modular, logo que o projeto desenvolvido pode ser considerado um mínimo produto viável ou MVP, e ser utilizado como um modelo inicial de

um produto comercializável. A ideia é ofertar a solução como um serviço para diferentes organizações que tenham interesse de automatizar o seu atendimento ao cliente.

### 6.1.1 Precificação do projeto

Resumidamente, temos dois tipos de custos diferentes que irão agregador no preço final do projeto, o custo de desenvolvimento e o custo tecnológico. O custo tecnológico é a somatória dos gastos em serviços e ferramentas usadas de outras em empresas como a plataforma, servidores e conectores. O custo de desenvolvimento consiste no valor cobrado pelo profissional que realizara as modificações e customizações necessárias para que o projeto fique de acordo com o solicitado pelo cliente, mais os custos de manutenção do sistema. Como estamos realizando o estudo de caso sobre o desenvolvimento do produto não iremos quantificar o valor sobre o custo de desenvolvimento, somente os gastos tecnológicos.

A plataforma da *Twilio* cobra um valor mensal de R\$24,00 para que o número do estabelecimento seja vinculado a *API* do *WhatsApp* utilizada, além disso, a plataforma realiza a cobrança em cima da quantidade de mensagens que foram realizadas, utilizando como referência as informações disponíveis no site da empresa os custos por mensagem são de US\$0,005 centavos de dólar.

Quantificando os gastos mensais de uma empresa que deseja utilizar esses sistemas de atendimento, precisaremos analisar a quantidade de mensagens e atendimentos que essa empresa realiza durante um mês, exemplificando que a empresa atenda a 500 clientes e cada um desses clientes faça em média a troca de 5 mensagens até conseguir concluir a compra, teremos assim um total de 2500 mensagens respondidas por mês, o que resultaria em torno de R\$75,00 reais em encargos de tráfego.

Já o preço de utilização do *Dialogflow* varia dependendo da necessidade do projeto, o pacote de teste oferece o uso gratuito da solução, mas é limitado a 180 solicitações por minuto. Todavia, a solução oferece um pacote pago, onde o custo é aproximadamente US\$0,002 por solicitação. Este é um valor relativamente baixo, visto que o tipo de solicitação que utilizamos na solução é somente a comunicação por texto, que é mais barata e simples que outros serviços da plataforma, como a detecção e resposta por áudio ou ligações telefônicas ([GOOGLE](#), ).

Na solução do Firebase Realtime Database, tal qual as outras aplicações disponibilizadas pelo *Google Cloud Platform*, os custos de operação do serviço variam de acordo com a utilização do sistema, existem duas opções principais, o básico e gratuito que oferece 1 GB de armazenamento total em sua base de dados, 20 mil gravações por dia, 50 mil leituras por dia e 20 mil exclusões por dia. Já o plano pago cobra por transações de corridas mensalmente, sendo US\$0,18 por cada Giga byte de armazenamento, e US\$0,026

por Giga byte utilizado em transações e transferência de dados. Graças a essas opções e a grande quantidade de informações que podem ser utilizadas, a solução desenvolvida empregou somente o plano básico, o que não acarreta custos adicionais a empresa que irá utilizar o sistema ([BOHBOT](#), ).

As informações sobre os preços dos serviços está sintetizado na Tabela 5.

Tabela 5 – Tabela de precificação

Twilio	Dialogflow	Firebase
Cobrança mensal de R\$24,00 pela utilização da API WhatsApp.	Gratuito até 180 solicitações por minuto.	Gratuito até 1GB de armazenamento, 20 mil gravações, 50 mil leituras e 20 mil exclusões por dia.
US\$0,005 por mensagem trafegada.	Aproximadamente US\$0,002 por solicitação.	US\$0,18 por GB de armazenamento, US\$0,026 por GB utilizado em transações.

## 6.2 Desenvolvimentos futuros

O trabalho desenvolvido introduziu os principais fundamentos para desenvolver um chatbot, mas ainda assim, há diversas melhorias e mudanças que podem ser implementadas ao projeto, para construir uma solução mais robusta capaz de competir diretamente com outros serviços disponíveis no mercado, como aplicativos e web-sites, dentre essas melhorias pode-se:

- Construir uma ferramenta voltada para o Gestor do negócio, onde o mesmo possa alterar e adicionar os produtos que estão listados no cardápio, sem precisar da interação do programador que implementou a ferramenta. Esta solução também pode conter um sistema de controle dos pedidos, que será utilizado para listar todos os pedidos que foram concluídos, como também para atualizar o estado de entrega do mesmo;
- Poderá vincular o sistema atual com *APIs* externas, como por exemplo a sistema de pagamento instantâneo PIX, automatizando o pagamento assim que o usuário selecionar esse método ao finalizar a compra;
- Outro exemplo seria comunicar com soluções de validação de endereço, que pode confirmar em um mapa o endereço real do cliente e até prever o tempo estimado de entrega do pedido baseado no trajeto percorrido;
- Criar funções de interação direta com os clientes que já utilizaram a plataforma, como possibilitar a escolha do mesmo endereço de entrega de uma compra anterior

ou possibilitar salvar os itens em um carrinho para finalizar a compra depois. Além de conhecer e sugerir os produtos preferidos daquele usuário, entre outras opções.

## 7 Considerações Finais

Neste trabalho, foi desenvolvido um sistema de *chatbot* automatizado para atendimento de um estabelecimento. A proposta aconteceu devido o crescimento na demanda do comércio online, que ampliou-se nos últimos anos, mudando o formato de venda, tanto dos comerciantes locais de pequenas e médias empresas, quanto nas grandes varejistas já conhecidas no mercado.

Foi necessário realizar um estudo voltado para o entendimento de uma solução de reconhecimento da linguagem natural que simulasse a iteração humana, além de conhecer as ferramentas disponíveis no mercado que conseguiram realizar as tarefas desejadas e possibilita-sem a integração do *chatbot* com a aplicação do *WhatsApp*. Neste sentido, pode-se dizer que os objetivos dos estudos iniciais do trabalho foram atingidos com sucesso.

Graças aos conhecimentos adquiridos e apresentados nesse trabalho, foi possível construir uma aplicação de *chatbot* utilizando diversas ferramentas de alto nível como o *Dialogflow*, *Twilio* e o *Firebase Realtime Database*. O desenvolvimento executado foi fundamental para entender na prática como deve ser construído uma aplicação de iteração com o usuário, capaz de responder cada uma das perguntas de forma objetiva e clara, simulando ao máximo uma conversação real.

Por fim, o protótipo construído atingiu as expectativas esperadas, já que as funcionalidades necessárias para conseguir simular uma experiência de diálogo foram implementadas de forma satisfatória. Além disso, foi possível realizar consultas e inserções no banco de dados, enviar imagens, utilizar o contexto de cada conversa e construir um fluxo de diálogo com navegação numérica. A partir dessa base de construção é possível implementar os conhecimentos adquiridos em diversos outros tipos de soluções do mesmo gênero. É plausível concluir que o estudo realizado obteve sucesso em demonstrar as ferramentas de construção de um *chatbot* e desenvolver uma solução de atendimento automatizado para uma lanchonete.

# Referências

- ADAMOPOULOU, E.; MOUSSIADES, L. An overview of chatbot technology. In: SPRINGER. *IFIP International Conference on Artificial Intelligence Applications and Innovations*. [S.l.], 2020. p. 373–383. [30](#)
- BARBOSA, M. *91% dos internautas realizaram compras pela internet nos últimos 12 meses, aponta pesquisa CNDL/SPC Brasil*. 2021. Disponível em: <https://site.cndl.org.br/91-dos-internautas-realizaram-compras-pela-internet-nos-ultimos-12-meses-aponta-pesquisa-cndlspc-brasil/> Acesso em 24 mar. 2022. [14](#)
- BARNEY, P. pelo M. *CHATBOTS para WhatsApp : A Nova Melhor Maneira de Criar o Seu*. YouTube, 2021. 1 vídeo (14:27). 2021. Disponível em: <https://www.youtube.com/watch?v=Ifh5FVoQg8>. Acesso em : 23set.2022.[41](#)
- BOHBOT, K. *Preços do Google Firebase Para Leigos*. Disponível em: <https://blog.back4app.com/pt/precos-do-google-firebase-para-leigos/> Acesso em 03 out. 2022. [55](#)
- BURBECK, S. Applications programming in smalltalk-80 (tm): How to use model-view-controller (mvc). *Smalltalk-80 v2*, v. 5, p. 1–11, 1992. [40](#)
- CALADO, A. *Entenda o que é um webhook e como ele funciona*. 2022. Disponível em: <https://rockcontent.com/br/blog/o-que-e-um-webhook/> Acesso em 21 set. 2022. [26](#)
- CETIC.BR. *Proporção de empresas brasileiras que venderam pela Internet cresce durante a pandemia e chega a 73%, revela pesquisa do CGI.b*. 2022. Disponível em: <https://cetic.br/pt/noticia/proporcao-de-empresas-brasileiras-que-venderam-pela-internet-cresce-durante-a-pandemia-e-chega-a-73-revela-pesquisa-do-cgi-br/> Acesso em 30 set. 2022. [19](#)
- ESCOVEDO, T.; KOSHIYAMA, A. *Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise*. [S.l.]: Casa do Código, 2020. [29](#), [31](#)
- GONZÁLES, M. *O que é machine learning e quais são as 7 etapas do aprendizado das máquinas*. 2018. Disponível em: <https://blog.idwall.co/o-que-e-machine-learning/> Acesso em 28 mar. 2022. [28](#)
- GOOGLE, C. *Preços do Dialogflow*. Disponível em: <https://cloud.google.com/dialogflow/pricing#agent> Acesso em 03 out. 2022. [54](#)
- JANARTHANAM, S. *Hands-on chatbots and conversational UI development: build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills*. [S.l.: s.n.], 2017. Packt Publishing Ltd. [24](#)
- MASCHIETTO LUIS G.; MORAES, D. M. P. D. A. N. S. R. e. a. *Desenvolvimento de Software com Metodologias Ágeis*. [S.l.: s.n.], 2021. Disponível em: <https://integrada.minhabiblioteca.com.br//books/9786556901824/>. Acesso em: 06 nov. 2022. [30](#)

- MAULDIN, M. L. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In: *AAAI*. [S.l.: s.n.], 1994. v. 94, p. 16–21. [17](#)
- MUICAHY, S. *State of the connected customer 4th edition*. 2020. Disponível em: [https://www.salesforce.com/content/dam/web/en\\_us/www/documents/research/salesforce-state-of-the-connected-customer-4th-ed.pdf](https://www.salesforce.com/content/dam/web/en_us/www/documents/research/salesforce-state-of-the-connected-customer-4th-ed.pdf) Acesso em 27 abr. 2022. [18](#)
- NIC.BR. *Pesquisa sobre o uso das tecnologias de informação e comunicação nas empresas brasileiras: Pesquisa TIC Empresas, ano 2021*. 2022. Disponível em: <https://cetic.br/pt/arquivos/pesquisa/2021/empresas/> Acesso em 30 set. 2022. [18](#)
- NICHOLS, J. A.; CHAN, H. W. H.; BAKER, M. A. Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical reviews*, Springer, v. 11, n. 1, p. 111–118, 2019. [28](#)
- PAIVA, F. *Bot da Domino's vende pizza pelo WhatsApp no Rio de Janeiro*. 2018. Disponível em: <https://www.mobiletime.com.br/noticias/13/11/2019/bot-da-dominos-vende-pizza-pelo-whatsapp-no-rio-de-janeiro/> Acesso em 30 mar. 2022. [20](#)
- RAUPP, F. M.; BEUREN, I. M. Metodologia da pesquisa aplicável às ciências. *Como elaborar trabalhos monográficos em contabilidade: teoria e prática*. São Paulo: Atlas, p. 76–97, 2006. [30](#)
- SILVA, B. C. D. da et al. Introdução ao processamento das línguas naturais e algumas aplicações. *Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional*, v. 3, 2007. [22](#)
- SOFTIĆ, A. et al. Health chatbot: design, implementation, acceptance and usage motivation. In: IEEE. *2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH)*. [S.l.], 2021. p. 1–6. [17](#)
- SUTHERLAND, J.; SCHWABER, K. The scrum guide. *The definitive guide to scrum: The rules of the game*. Scrum. org, v. 268, p. 19, 2013. [31](#)
- TEIXEIRA, J. de F. *Inteligência artificial*. [S.l.]: Pia Sociedade de São Paulo-Editora Paulus, 2014. [28](#)
- TURING, A. M.; HAUGELAND, J. Computing machinery and intelligence. *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*, p. 29–56, 1950. [14](#)
- WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, ACM New York, NY, USA, v. 9, n. 1, p. 36–45, 1966. [14](#)