



UNIVERSIDADE FEDERAL DO MARANHÃO
Curso de Ciência da Computação

Hannah Moreira Costa

**Uma Solução de Chatbot para Auxiliar os
Alunos do Curso de Ciência da Computação da
UFMA**

São Luís
2024

Hannah Moreira Costa

Uma Solução de Chatbot para Auxiliar os Alunos do Curso de Ciência da Computação da UFMA

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof^ª. Dr^ª. Simara Vieira da Rocha

São Luís

2024

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Moreira Costa, Hannah.

Uma Solução de Chatbot para Auxiliar os Alunos do Curso de Ciência da Computação da UFMA / Hannah Moreira Costa. - 2024.

51 f.

Orientador(a): Simara Vieira da Rocha.

Curso de Ciência da Computação, Universidade Federal do Maranhão, UFMA, 2024.

1. Chatbot. 2. Dialogflow. 3. Processamento de Linguagem Natural. 4. Telegram. I. Vieira da Rocha, Simara. II. Título.

Hannah Moreira Costa

Uma Solução de Chatbot para Auxiliar os Alunos do Curso de Ciência da Computação da UFMA

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho _____ em São Luís, 5 de janeiro de 2024:

Prof^a. Dr^a. Simara Vieira da Rocha
Orientador

**Prof. Msc. Carlos Eduardo Portela
Serra de Castro**
Examinador

Profa. Dr. Ivo José da Cunha Serra
Examinador

São Luís
2024

Agradecimentos

A minha família, especialmente aos meus pais e avós, por me apoiarem nos meus sonhos e não me deixarem desanimar.

Ao meu namorado Thalles, que esteve me incentivando e ajudando desde o início do curso até o fim, ouvindo meus desabafos e comemorando comigo cada vitória.

A Prof^ª. Dr^ª. Simara, por ser uma excelente orientadora e uma grande inspiração pra mim.

A Leidiane por ter me auxiliado com os processos da Coordenação durante o curso e durante a realização deste trabalho.

Aos meus amigos Alana, Antonio, Durand, Giovanna, Igor e Wesley pela parceria e companheirismo durante a graduação.

A todos os professores que tive na UFMA e na vida, por todo o conhecimento e educação que me proporcionaram.

Ao Sid pela sugestão de material para essa monografia, foi de grande ajuda.

*"A ciência atua na fronteira entre
o conhecimento e a ignorância
sem medo de admitir que não sabemos.
Não há nenhuma vergonha nisso.*

A única vergonha é fingir que temos todas as respostas."

Neil deGrasse Tyson, em *"Cosmos: Uma Odisseia do Espaço-Tempo"*

Resumo

Com o contínuo avanço da tecnologia, a inteligência artificial deu origem a diversas inovações notáveis, entre as quais se destacam os chatbots. Esses agentes virtuais inteligentes fazem uso do Processamento de Linguagem Natural (PLN), uma subárea da Inteligência Artificial dedicada à compreensão e interação linguística. A tecnologia dos chatbots, ao incorporar o PLN, abre caminho para a criação de soluções inovadoras, proporcionando aos usuários uma maneira conveniente de buscar informações. Atualmente existe uma multiplicidade de plataformas especializadas na criação de chatbots, assim como uma variedade de contextos nos quais esses podem ser implantados. Destacando-se no atendimento ao cliente, os chatbots encontram aplicação em diversas áreas, abrangendo desde setores de vendas até suporte e saúde. Esta versatilidade torna esses agentes virtuais valiosos aliados em questões práticas e estratégicas. Nesse cenário, esse trabalho tem como objetivo propor uma solução de chatbot para auxiliar os alunos do curso de ciência da computação da UFMA. Para atingir este objetivo, foi desenvolvido o AdaBot, um chatbot criado na plataforma Dialogflow, para proporcionar comodidade aos estudantes, centralizando informações importantes em um único ponto de acesso, o aplicativo de mensagens Telegram. Para compreender a aceitação dos alunos ao AdaBot foi realizada uma pesquisa com pessoas que têm ou já tiveram relação com o curso de computação, onde o chatbot foi bem avaliado e demonstrou ter potencial no meio acadêmico de computação.

Palavras-chave: Chatbot. Dialogflow. Processamento de Linguagem Natural. Telegram.

Abstract

With the continuous advancement of technology, artificial intelligence has given rise to several notable innovations, among which chatbots stand out. These intelligent virtual agents make use of Natural Language Processing (NLP), a subarea of Artificial Intelligence dedicated to linguistic understanding and interaction. Chatbot technology, by incorporating PLN, paves the way for the creation of innovative solutions, providing users with a convenient way to search for information. There are currently a multitude of platforms specialized in creating chatbots, as well as a variety of contexts in which they can be deployed. Standing out in customer service, chatbots find application in several areas, ranging from sales to support and healthcare sectors. This versatility makes these virtual agents valuable allies in practical and strategic matters. In this scenario, this work aims to propose a chatbot solution to assist students on the computer science course at UFMA. To achieve this objective, AdaBot was developed, a chatbot created on the Dialogflow platform, to provide convenience to students, centralizing important information in a single access point, the Telegram messaging application. To understand students' acceptance of AdaBot, a survey was carried out with people who have or have already had a connection with the computer science course, where the chatbot was well evaluated and demonstrated its potential in the computer science academic environment.

Keywords: Chatbot. Dialogflow. Natural Language Processing. Telegram.

Lista de ilustrações

Figura 1 – Exemplo de correspondência entre frases do usuário e <i>intent</i> do agente virtual e extração de parâmetros.	20
Figura 2 – Etapas da metodologia para produção do chatbot.	25
Figura 3 – <i>Token</i> de acesso à API HTTP do Telegram criado pelo BotFather.	27
Figura 4 – Menu lateral e menu de integrações no Dialogflow.	28
Figura 5 – <i>Pop-up</i> de integração do Dialogflow com o Telegram para inserção do <i>token</i>	28
Figura 6 – <i>Intents</i> padrões de um agente do Dialogflow.	29
Figura 7 – Menu principal do AdaBot.	30
Figura 8 – Resposta do tipo <i>custom payload</i> para <i>intent</i> Menu.	31
Figura 9 – Seção para configuração de eventos na <i>intent</i> Menu.	32
Figura 10 – Aba de históricos de conversas com o chatbot no Dialogflow.	33
Figura 11 – <i>Raw interaction log</i> de uma interação em formato JSON.	34
Figura 12 – Contexto configurado para a <i>intent</i> TCC.	34
Figura 13 – Menu com assuntos relacionados ao tópico "TCC".	35
Figura 14 – O evento configurado para <i>fulfillment</i> na <i>intent</i> TCC.	35
Figura 15 – Resumo das normas do TCC apresentado pelo chatbot.	36
Figura 16 – Ativação de <i>fulfillment</i> dentro da <i>intent</i>	36
Figura 17 – Menu lateral e menu de <i>fulfillment</i> no Dialogflow.	37
Figura 18 – Agente instanciado como <i>fulfillment</i> do Dialogflow.	37
Figura 19 – Chamada do evento <i>anterior-tcc</i> por meio do método <i>setFollowupEvent</i>	38
Figura 20 – Botões para continuar a conversa ou finalizá-la.	39
Figura 21 – Resposta do tipo <i>custom payload</i> para <i>intent</i> de modelo de avaliação de estágio.	40
Figura 22 – Gráfico de pizza para ilustrar respostas à pergunta 1 do formulário de <i>feedback</i>	41
Figura 23 – Gráfico de pizza para ilustrar respostas à pergunta 2 do formulário de <i>feedback</i>	42
Figura 24 – Gráfico de pizza para ilustrar respostas à pergunta 3 do formulário de <i>feedback</i>	42
Figura 25 – Gráfico de pizza para ilustrar respostas à pergunta 4 do formulário de <i>feedback</i>	43
Figura 26 – Gráfico de pizza para ilustrar respostas à pergunta 5 do formulário de <i>feedback</i>	43
Figura 27 – Gráfico de pizza para ilustrar respostas à pergunta 6 do formulário de <i>feedback</i>	44

Figura 28 – Gráfico de pizza para ilustrar respostas à pergunta 7 do formulário de *feedback*. 44

Lista de tabelas

Tabela 1 – Exemplo de interação com contexto entre agente e usuário.	21
Tabela 2 – Relação entre botões, <i>callback data</i> e nomes de <i>intents</i> do menu principal.	32
Tabela 3 – Tabela relacionando botões, <i>callback data</i> e nomes de <i>intents</i> para o menu do tópico TCC.	35

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
CC	<i>Ciência da Computação</i>
CVA	<i>Cognitive Virtual Assistant</i>
IOT	<i>Internet Of Things</i>
NLP	<i>Natural Language Processing</i>
NLG	<i>Natural Language Generation</i>
NLU	<i>Natural Language Understanding</i>
PLN	<i>Processamento de Linguagem Natural</i>
UFMA	<i>Universidade Federal do Maranhão</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	Objetivos Específicos	13
1.2	Trabalhos Relacionados	13
1.3	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Processamento de Linguagem Natural	15
2.2	Chatbots	16
2.2.1	Classificações de Chatbots	16
2.2.1.1	Domínio	16
2.2.1.2	Serviço providenciado	17
2.2.1.3	Objetivo	17
2.2.1.4	Modelo de processamento de <i>inputs</i> e método de geração de resposta	17
2.3	Tecnologias analisadas para o desenvolvimento de chatbots	18
2.3.1	Interface de conversação	19
2.3.1.1	Dialogflow	19
2.3.1.2	IBM Watson	22
2.3.1.3	RASA	22
2.3.2	Ferramentas para comunicação do chatbot com o usuário	23
2.3.2.1	WhatsApp	23
2.3.2.2	Telegram	24
3	METODOLOGIA	25
3.1	Levantamento de Requisitos	25
3.2	Construção do Chatbot	26
3.3	Testes e Avaliação	41
4	CONCLUSÃO	47
	REFERÊNCIAS	48

1 Introdução

No atual cenário de rápidos avanços tecnológicos, a busca pelo aproveitamento da tecnologia para aprimorar a qualidade de vida e conforto das pessoas é cada vez mais evidente. Nesse contexto, os chatbots se destacam como uma solução tecnológica voltada para desempenhar tarefas diversas, oferecendo informações aos clientes e usuários de forma instantânea e acessível em qualquer local.

Os chatbots utilizam inteligência artificial, mais precisamente processamento de linguagem natural. Essa tecnologia permite que o computador receba a entrada do usuário, a processe e compreenda, habilitando-o a responder ou executar tarefas de maneira similar à interação humana.

Os chatbots mais notáveis na atualidade incluem o ChatGPT ([OPENAI, 2023](#)), Alexa ([AMAZON, 2023](#)) e Siri ([APPLE, 2023](#)), os quais se destacam por seu uso produtivo tanto no ambiente de trabalho quanto no entretenimento. Além disso, são muito comuns os chatbots para negócios, usualmente utilizados em plataformas como WhatsApp, auxiliando no atendimento ao cliente, vendas, suporte, entre outros.

Essas soluções automatizadas desempenham um papel crucial no dia a dia, facilitando a busca por informações, oferecendo auxílio em diversas tarefas e estando disponíveis a qualquer momento e em qualquer lugar. Essa versatilidade contribui para uma experiência mais eficiente e conveniente para os usuários. Além disso, os sistemas de chatbot podem ser empregados para otimizar a execução de tarefas repetitivas e pouco complexas, melhorando o atendimento ao usuário ou cliente e proporcionando maior agilidade na resolução de suas questões.

Dito isso, um exemplo de cenário onde seria vantajoso a adoção de um sistema de chatbot é encontrado no curso de Ciência da Computação da Universidade Federal do Maranhão (UFMA), pois, com frequência, os estudantes matriculados necessitam acessar informações ou obter documentos da Coordenação do Curso. Para atender a tais demandas, os discentes costumam se deslocar até a sala da Coordenação ou recorrer ao envio de e-mails para solicitações específicas.

Essas práticas, no entanto, muitas vezes acarretam lentidão no processo de obtenção das informações desejadas. Além disso, é importante destacar que essa abordagem requer a disponibilidade de funcionários e por isso, pode resultar em tempos de espera significativos.

Ao procurar informações junto à Coordenação, os estudantes podem enfrentar um período de espera devido à restrição de horários e à carga de trabalho dos funcionários, que por vezes estão ocupados com outras solicitações. Adicionalmente, no caso de atendimento

presencial, é necessário considerar o deslocamento do estudante até o local, o que pode representar um desafio, especialmente para aqueles que residem em áreas mais afastadas.

Nesta monografia será abordada a criação de um chatbot informativo para alunos de computação da UFMA, chamado AdaBot, utilizando o framework Dialogflow e o aplicativo de mensagens Telegram. Também serão abordados os conceitos fundamentais para entendimento da solução implementada, bem como uma pesquisa realizada com os usuários que testaram o AdaBot para entender a aceitação do chatbot.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver uma solução de chatbot para auxiliar os alunos do Curso de Ciência da Computação da UFMA no atendimento de suas demandas acadêmicas.

1.1.1 Objetivos Específicos

Destaca-se como objetivos específicos deste trabalho:

- Definir e estruturar as informações mais relevantes relacionadas às demandas acadêmicas.
- Modelar e desenvolver o sistema de chatbot utilizando ferramentas majoritariamente gratuitas.
- Validar o uso da ferramenta desenvolvida com os usuários.

1.2 Trabalhos Relacionados

O trabalho de [Reyes et al. \(2019\)](#) propõe uma metodologia para desenvolver e implementar um chatbot como tutor inteligente em cursos universitários. O artigo propõe uma metodologia centrada na utilização de cálculos de predicados de primeira ordem para representar unidades de conhecimento, dividida em modelagem de conhecimento e fluxo de conversação. A modelagem sugere agrupar unidades de conhecimento por similaridade de entradas do usuário, enquanto o fluxo de conversação envolve a criação de um glossário com IDs únicos para cada unidade de conhecimento. Os autores aplicaram a metodologia desenvolvendo chatbots para cursos de física e matemática utilizando o IBM Watson.

No trabalho de [Ranavare e Kamath \(2020\)](#), é abordada a modelagem e construção de um chatbot para o ambiente acadêmico, focado em fornecer informações sobre a participação dos alunos em atividades institucionais. Os autores destacam a conveniência de disponibilizar um agente virtual que pode ajudar os alunos de forma acessível e flexível,

a qualquer momento e em qualquer lugar. Além disso, ele ressalta a facilidade de utilizar o *framework* Dialogflow para desenvolver o chatbot e integrá-lo ao site da instituição.

Em seu trabalho, [Muhammad et al. \(2020\)](#) aborda a relevância do aprendizado da língua inglesa, destacando os desafios enfrentados pela maioria dos estudantes, com ênfase na habilidade da fala. Diante desse cenário, o autor propõe a utilização do *framework* Dialogflow para desenvolver um chatbot com reconhecimento de fala. O propósito desse chatbot é auxiliar os alunos a aprimorarem suas habilidades de fala em inglês, proporcionando diálogos interativos disponíveis no aplicativo desenvolvido como parte de seu trabalho.

[Barus e Surijadi \(2022\)](#) trazem em seu trabalho um cenário em que a criação e implementação de um chatbot seriam oportunas. Eles abordam a situação específica da biblioteca universitária de Matana, localizada na Indonésia, durante a pandemia de coronavírus, a qual provocou significativas transformações no funcionamento da instituição. Tais mudanças incluíram a necessidade de adotar protocolos de saúde que impactaram o conforto no local, a escassez de funcionários e a urgência de aprimorar os serviços online, visando torná-los acessíveis a qualquer momento. Com todas as aulas e atividades da universidade sendo conduzidas de maneira remota durante esse período, surgiu a demanda por um serviço online para a biblioteca. O objetivo era responder às perguntas frequentes dos estudantes de forma eficiente. Para atender a essa necessidade, os autores propõem a implementação de um chatbot construído na plataforma Dialogflow.

1.3 Organização do Trabalho

Este trabalho está organizado em quatro capítulos, além do atual, de forma a apresentar o conteúdo mais claramente, conforme os parágrafos a seguir.

No capítulo 2 será contemplada a fundamentação teórica, a fim de esclarecer os conceitos principais para entendimento da solução adotada, abordando o processamento de linguagem natural, o conceito de chatbots e tipos de classificação, bem como as plataformas sondadas para implementação do chatbot apresentado neste trabalho.

O capítulo 3 aborda detalhadamente o caminho percorrido para realização da solução aqui apresentada, desde a sua concepção e desenvolvimento até seu teste e avaliação pelos usuários.

Por fim, no capítulo 4 serão apresentadas as considerações finais sobre o trabalho realizado, bem como sugestões de trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo será apresentada a fundamentação teórica usada no desenvolvimento desta monografia. Para tanto, abordaremos conceitos sobre processamento de linguagem natural (PLN), chatbot e as tecnologias analisadas para seu desenvolvimento e implantação.

2.1 Processamento de Linguagem Natural

Antes de explorar o tema dos chatbots, é essencial ter um entendimento prévio de um conceito fundamental que permeia a maioria desses sistemas atualmente: o PLN.

O Processamento de Linguagem Natural (PLN), também conhecido em inglês como *Natural Language Processing*, é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana. O adjetivo “Natural”, na sigla, se refere às línguas faladas pelos humanos, distinguindo-as das demais linguagens (matemáticas, visuais, gestuais, de programação etc.) (CASELI; NUNES, 2023).

Dessa forma, no PLN buscam-se soluções para situações que requerem o tratamento computacional de uma língua. Os chatbots são um exemplo clássico de problema computacional que é tratado com PLN, pois o processamento e compreensão da expressão do usuário são os primeiros passos para que o chatbot entenda como deve responder ou que ação deverá tomar.

O PLN se divide em duas grandes subáreas: Interpretação (ou Compreensão) de Linguagem Natural – NLU (do inglês, *Natural Language Understanding*), e Geração de Linguagem Natural – NLG (do inglês, *Natural Language Generation*).

No âmbito do NLU, engloba-se tudo relacionado ao processamento voltado para a análise e interpretação da linguagem, onde a análise trata da segmentação e classificação dos componentes linguísticos e a interpretação compreende a construção de significados a partir do que foi dito. Para um chatbot, a interpretação determina o que deve ser feito em resposta ao usuário.

Já a NLG foca na geração de linguagem natural. Um exemplo prático de aplicação dessa área ocorre na geração de respostas para usuários em chatbots construídos em modelo generativo, como o ChatGPT (OPENAI, 2023). Algumas das classificações mais utilizadas para chatbots serão abordadas na seção 2.2.1, incluindo a classificação por tipo de modelo.

2.2 Chatbots

Um chatbot, também conhecido como bot inteligente, agente interativo, assistente digital ou entidade de conversação artificial, é um programa de computador projetado para responder como uma entidade inteligente que compreende uma ou mais linguagens humanas, por texto ou por voz, empregando o PLN (KHANNA et al., 2015). Para Sabharwal et al. (2020), um chatbot também pode ser definido como um assistente cognitivo virtual (*Cognitive Virtual Assistant*, CVA), ou seja, um agente de software que executa tarefas para humanos ou máquinas baseado em texto, voz e informações visuais. Ele possui uma habilidade de entender os dados oferecidos, interpretá-los, executar operações passo a passo e sondar o usuário para obtenção de informações faltantes. Como resultado, pode apresentar uma informação ou executar a tarefa pretendida. CVAs possuem a capacidade de entender conversações em linguagem natural e manter o contexto da conversação para alcançar o objetivo final.

Esses bots inteligentes têm uma aplicação diversificada, podendo ser ferramentas altamente produtivas ou destinadas ao entretenimento, entre outras possibilidades. No contexto produtivo, os chatbots são utilizados para automatizar tarefas, responder a dúvidas frequentes e facilitar a interação em diversos setores. Sua capacidade de compreender e processar a linguagem humana permite que desempenhem funções complexas, melhorando a eficiência operacional. Um exemplo de chatbot muito utilizado nos dias de hoje, com foco produtivo, é o ChatGPT (OPENAI, 2023) ou o Bard (GOOGLE, 2023a).

Além disso, no âmbito do entretenimento, os chatbots proporcionam interações envolventes, simulando diálogos naturais com os usuários. Essa versatilidade na aplicação reflete a adaptabilidade dos chatbots em diversas situações, desde aprimorar a experiência do cliente até criar experiências de entretenimento personalizadas. A plataforma Character.AI (CHARACTER.AI, 2023), por exemplo, oferece diversos chatbots com personalidades diferentes para conversar com o usuário, variando desde assistentes de estudo/trabalho a personagens históricos ou fictícios.

2.2.1 Classificações de Chatbots

Segundo Nimavat e Champaneria (2017), chatbots podem ser classificados de diversas formas:

2.2.1.1 Domínio

Essa classificação se baseia no conhecimento que o chatbot tem ou acessa.

- Aberto: possui uma área de conhecimento ampla e aberta, podem ser utilizados para executar diversas tarefas ou responder diversas questões, ou seja, lidam com tópicos gerais. Exemplo: Amazon Alexa ([AMAZON, 2023](#)) e Apple Siri ([APPLE, 2023](#)).
- Fechado: possui foco em executar tarefas e oferecer informações dentro um domínio bem definido e pode falhar em outras situações. Exemplo: um chatbot para marcar consultas.

2.2.1.2 Serviço providenciado

Esta classificação refere-se à proximidade emocional do chatbot com o usuário.

- Interpessoal: Chatbots interpessoais têm como único foco acessar suas informações e fornecê-las ao usuário e oferecer serviços, não sendo esperado que aja como seu companheiro. Exemplo: um chatbot de atendimento ao cliente.
- Intrapessoais: Chatbots intrapessoais providenciam serviços e informações ao usuário de forma pessoal, como por exemplo, marcar compromissos no calendário. Exemplo: Amazon Alexa, Apple Siri.
- Interagentes: Chatbots interagentes são bots que interagem entre si, de forma integrada e que podem acessar serviços um do outro. Este tipo de chatbot é característico de cenários de Internet das Coisas (*Internet Of Things*, IOT). Exemplo: integração Alexa-Cortana, esteve ativa até 2021 ([VOICE, 2021](#)).

2.2.1.3 Objetivo

Esta classificação diz respeito à finalidade e objetivo que pretende-se alcançar com o chatbot.

- Informativos: visam oferecer informações como respostas ao usuário, informações essas que são obtidas de uma fonte fixa como uma página de perguntas frequentes de um site ou um banco de dados. Exemplo: bots para responder a perguntas frequentes.
- Conversacionais: objetivam responder corretamente à conversação com o usuário da mesma forma que faria outro ser humano. Exemplo: Amazon Alexa, Apple Siri.
- Baseado em tarefas: têm como foco executar ações para o usuário. Exemplo: bots para fazer pedidos em um restaurante ou reservas.

2.2.1.4 Modelo de processamento de *inputs* e método de geração de resposta

Essa classificação se refere a como o modelo utilizado processa inputs e gera as devidas respostas. Os modelos podem ou não ser baseados em IA ([IRANI; OJAMERUAYE; KAPOOR, 2020](#)).

- Modelo baseado em regras: chatbots desenvolvidos no modelo baseado em regras escolhem suas respostas baseado em um conjunto predefinido de regras, em que reconhecem as entradas do usuário sem criar novas respostas em texto, sendo o chatbot totalmente programado por um ser humano e estruturado na forma de árvore de diálogos. Este modelo não é baseado em IA. Exemplo: um bot para fazer pedidos num restaurante.
- Modelo baseado em recuperação: chatbots de modelo baseado em recuperação oferecem mais flexibilidade em relação ao modelo anterior, uma vez que recuperam recursos e analisam a entrada do usuário utilizando APIs para que possam classificar a intenção da conversa e selecionar uma resposta satisfatória para o contexto, dentre uma coleção de respostas predefinidas. Este modelo é baseado em IA. Exemplo: chatbots construídos com ferramentas como Dialogflow, IBM Watson, etc.
- Modelo generativo: chatbots com modelo generativo geram respostas mais próximas às de um humano e se baseiam tanto na mensagem atual do usuário quanto nas mensagens anteriores. Esse modelo utiliza algoritmos de *machine learning* e *deep learning*. Este modelo é baseado em IA. Exemplo: ChatGPT e Bard.

É importante frisar que chatbots podem ser classificados de várias formas ao mesmo tempo e por vezes podem pertencer a mais de um tipo dentro da mesma classificação.

Tendo em vista as classificações aqui apresentadas, podemos classificar o chatbot desenvolvido neste trabalho dentro das categorias a seguir.

- Domínio: fechado.
- Serviço providenciado: interpessoal.
- Objetivo: informativo.
- Modelo de processamento de *inputs* e método de geração de resposta: modelo baseado em regras.

Para seguir o entendimento da solução implementada neste trabalho, serão apresentadas as tecnologias analisadas para o desenvolvimento do chatbot e as plataformas analisadas para sua implantação, com foco nas tecnologias escolhidas para a sua execução propriamente dita.

2.3 Tecnologias analisadas para o desenvolvimento de chatbots

Essa seção apresenta algumas das tecnologias que podem ser utilizadas para o desenvolvimento de sistemas de chatbot.

2.3.1 Interface de conversação

Foram analisadas três tecnologias principais para construção de chatbots: Dialogflow, IBM Watson e RASA, que serão abordadas a seguir.

2.3.1.1 Dialogflow

O Dialogflow ([GOOGLE, 2023c](#)) é uma plataforma especializada na criação de chatbots e assistentes virtuais desenvolvida pela Google, existindo em duas edições: a edição paga CX, com recursos mais complexos e avançados, voltados para o uso em larga escala; e a edição gratuita ES, para uso em pequena ou média escala, com recursos padrões moderadamente complexos ([GOOGLE, 2023d](#)). A plataforma oferece recursos de Inteligência Artificial voltados para a compreensão da linguagem natural e é capaz de fornecer respostas adequadas com base nos fluxos de conversação definidos pelo desenvolvedor. Além disso, a plataforma permite integração com vários sistemas externos, entre eles, Twilio, Facebook Messenger, LINE, Slack, Kik e Telegram ([GOOGLE, 2023e](#)).

Um chatbot construído na plataforma é chamado de agente virtual. O agente é um módulo de processamento de linguagem natural que traduz as entradas do usuário final, sejam elas em forma de texto ou áudio, em dados estruturados compreensíveis por seus aplicativos e serviços.

O agente do Dialogflow responde ao usuário final com base nas intenções - *intents* - previamente definidas para ele. As *intents* categorizam as intenções, ou seja, objetivos ou propósitos que um usuário tem ao interagir com o chatbot. Em resposta à expressão do usuário, o agente procura uma correspondência nas *intents* que foram configuradas, a fim de entender o que o usuário deseja.

Uma *intent* básica é composta por frases de treinamento, ação, parâmetros e respostas associadas, permitindo que o agente estabeleça uma conexão entre a entrada do usuário e a resposta previamente definida na *intent* correspondente.

As frases de treinamento são exemplos de expressões que o usuário final poderá utilizar para acionar uma *intent*. O *machine learning* integrado ao Dialogflow permite que não seja necessário uma correspondência exata entre as frases de treinamento e as frases do usuário.

Ações podem ser definidas dentro das *intents*, de forma que, quando ativadas, o próprio Dialogflow fornece a ação que será executada ao sistema.

Os parâmetros são valores identificados e extraídos da expressão do usuário que podem ser utilizados em código para lógicas de processamento do agente ou para gerar respostas.

As respostas determinam o que a *intent* irá retornar à expressão do usuário a que foi associada. Podem ser respostas diretas ao usuário, solicitações de mais informações ou encerramento da conversa.

Na Figura 1 é mostrado o exemplo de um agente virtual que fornece previsões do tempo. Nesse caso, há três frases de usuário, todas correspondentes à mesma *intent* de "Previsão do tempo" devido a expressões correlatas que fazem parte das frases.

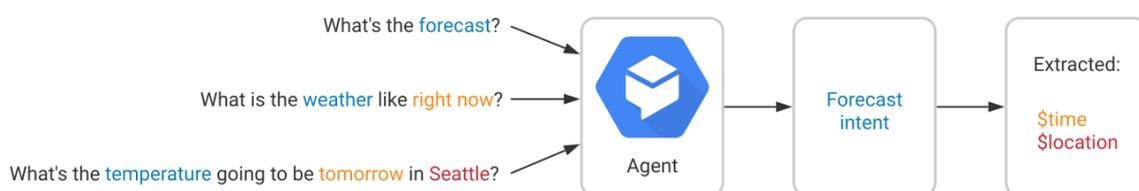
1. "Qual é a previsão do tempo?"
2. "Como está o tempo agora?"
3. "Qual será a temperatura amanhã em Seattle?"

O agente identifica a intenção do usuário por meio de expressões de treino, como "previsão", "tempo" e "temperatura", representados na cor azul na Figura 1. Assim é determinada a *intent* correspondente.

No exemplo da segunda frase, o agente identifica um parâmetro de tempo ($\$time$) na frase do usuário por meio da expressão "agora", destacada na cor laranja na Figura 1. Esse parâmetro é extraído da frase para ser incluído no processamento, permitindo que o agente retorne uma resposta personalizada ao usuário, levando em consideração o parâmetro fornecido.

Na terceira frase, são identificados dois parâmetros; um de tempo ($\$time$), "amanhã"; e outro de lugar ($\$location$), "Seattle", representados respectivamente pelas cores laranja e vermelho na Figura 1. O agente responderá fornecendo a previsão do tempo conforme o período e localidade especificados pelo usuário.

Figura 1 – Exemplo de correspondência entre frases do usuário e *intent* do agente virtual e extração de parâmetros.



Fonte: (GOOGLE, 2023g).

Dentro dos conceitos fundamentais de um chatbot no Dialogflow, é importante abordar também dois conceitos que serão utilizados na seção 3.2 do capítulo 3, os contextos e os *lifespan count*.

Para chatbots, contextos funcionam de forma semelhante à vida real. Em frases como o exemplo, "eles são azuis", não há sentido sem contexto. É necessário especificar a

quem "eles" se refere para obter-se um completo entendimento. Da mesma forma, o agente virtual precisa de um contexto para que possa processar a frase do usuário e correspondê-la a uma *intent*. Dessa forma, a uma *intent* do agente podem ser atribuídos contextos, que podem ser de entrada ou saída.

Quando atribuído a uma *intent*, o contexto de entrada determina que aquela *intent* só será correspondida (ou seja, chamada) quando a conversa se encontrar dentro do contexto específico configurado para ela.

Quando configurados em uma *intent*, os contextos de saída determinam o sentido da conversa naquele momento, de forma que *intent* mais específicas relacionadas àquele contexto possam ser chamadas.

No entanto, é necessário especificar um período de tempo em que um contexto faz sentido, afinal, uma conversa não irá permanecer no mesmo contexto permanentemente. Por isso existe o conceito de *lifespan count*, que determina o tempo de duração de um contexto. A duração do *lifespan* é medida em rodadas de conversa.

A Tabela 1 exemplifica uma interação onde o contexto em questão possui *lifespan count* de 2.

Tabela 1 – Exemplo de interação com contexto entre agente e usuário.

Rodada	Diálogo	Explicação
1	Agente: de que tipo de animal de estimação você gosta?	A intent "pet-init" é correspondida.
2	Usuário: Gosto de gatos. Agente: O que você quer saber sobre gatos?	A intent pet-select é correspondida. O contexto de saída "pet-chosen" se torna o contexto ativo.
3	Usuário: qual é a aparência deles? Agente: aqui está uma foto de um gato. O que mais você quer saber?	A intent pet-show tem um contexto de entrada "pet-chosen", por isso é correspondida. O contexto "pet-chosen" esteve ativo por uma rodada.
4	Usuário: qual é o som deles? Agente: ouça o som de um gato. O que mais você quer saber?	A intent "pet-audio" tem um contexto de entrada "pet-chosen", por isso é correspondida. O contexto "pet-chosen" esteve ativo por duas rodadas.
5	Usuário: qual é o tamanho deles? Agente: de que tipo de animal de estimação você gosta?	A intent "pet-init" é correspondida novamente, porque o contexto "pet-chosen" não está mais ativo.

Fonte: adaptado de (GOOGLE, 2023b).

Conforme é possível verificar na Tabela 1, na rodada 1, a *intent* "pet-init" é correspondida.

Na rodada 2, após o usuário selecionar o animal, o contexto de saída "pet-chosen" se torna ativo e possui duração de 2 rodadas.

A rodada 3 tem "pet-chosen" como contexto de entrada e resta uma rodada na duração do mesmo.

Na rodada 4, o contexto de entrada é "pet-chosen", que continua ativo, por isso a *intent* é correspondida.

Na rodada 5, o contexto "pet-chosen" não está mais ativo e o agente não corresponde uma *intent* à entrada do usuário.

2.3.1.2 IBM Watson

O IBM Watson, também chamado de Watson Assistant, é uma plataforma de inteligência artificial conversacional que permite a construção e manutenção de chatbots a partir de uma interface intuitiva com o objetivo de evitar que o desenvolvedor precise escrever linhas de código (IBM, 2023a). Para o *framework* são oferecidas uma versão de teste gratuito com limite de chamadas de API mensais e versões pagas que variam de acordo com o plano escolhido.

No Watson, a estrutura principal e equivalente à *intent* do Dialogflow é a ação. A ação representa um fluxo de conversa, onde em primeiro lugar são definidas as expressões de treinamento que darão início àquele fluxo. Nele podem ser definidas as etapas, onde são determinadas as respostas do assistente e do usuário e lógicas e condições daquele fluxo.

Assim como no Dialogflow, uma vez que cada ação é definida, o assistente Watson passa por um treinamento com base nas configurações de frases de treinamento. Isso possibilita que, a partir da próxima interação, o chatbot responda o usuário conforme a ação reconhecida.

Da mesma forma que o Dialogflow, o IBM Watson oferece várias alternativas de integração, como Slack, Twilio, Facebook Messenger, entre outras.

Uma das principais diferenças notadas entre o Dialogflow e o IBM Watson é a maior disponibilidade de configurações das ações por meio da interface do IBM Watson. Atualmente, no Dialogflow, algumas lógicas específicas só podem ser implementadas por meio de código.

Um exemplo disso seria a ação de voltar ao menu anterior. No Dialogflow é necessário implementar um código para a executar essa ação. Mesmo que o código implementado seja curto, o IBM Watson proporciona mais comodidade e intuitividade na criação dessa ação, pois permite que essa lógica seja criada dentro das ações, sem necessidade de código, por meio da interface.

2.3.1.3 RASA

O RASA é um *framework* de IA conversacional que permite a criação de assistentes virtuais. O RASA possui uma distribuição *open source* e de código aberto, gratuita para uso pessoal e comercial que, no entanto, não oferece uma interface gráfica para o desenvolvimento de assistentes virtuais. Por outro lado, também existem versões pagas do *framework* que oferecem a comodidade da interface gráfica e ferramentas de análise para entendimento do desempenho do assistente virtual.

A criação de um assistente no *framework* RASA está estruturada em algumas partes principais (NORONHA, 2020):

- NLU: ou *Natural Language Understanding* - para que o RASA entenda as mensagens do usuário em forma de dados estruturados, é necessário oferecer exemplos para o treinamento do modelo. Na parte do NLU são definidas as *intents*, que são compostas por exemplos de frases de treinamento.
- Domínio: no domínio são especificadas estruturas como as *intents*, entidades, respostas, ações, entre outras.
- Histórias: ou *stories*, onde são definidos os fluxos. Cada fluxo possui um passo-a-passo, formado por *intents* e ações.
- Regras: descrevem pequenas partes de conversas que devem seguir sempre o mesmo caminho.
- Ações: são tarefas que o bot pode executar, como por exemplo, responder ao usuário, fazer chamadas a uma API, consultar um banco de dados, etc.

2.3.2 Ferramentas para comunicação do chatbot com o usuário

Para realizar a implantação do chatbot e disponibilizá-lo aos discentes do curso de computação, foram analisados dois aplicativos de mensageria: Whatsapp e Telegram.

2.3.2.1 WhatsApp

O WhatsApp ([WHATSAPP, 2023](#)) é um aplicativo de mensagens e chamadas privadas amplamente utilizado no Brasil, ocupando o primeiro lugar na lista de aplicativos instalados nos celulares dos brasileiros, segundo pesquisa Panorama, realizada por [MobileTime e OpinionBox \(2023\)](#).

O aplicativo oferece ainda uma opção de contas para empresas e outros negócios, conhecido como WhatsApp *Business*. Isso permite que várias contas dessa modalidade adotem o uso de chatbots para auxiliá-los com a demanda e atendimento.

Ao contrário da maioria das outras plataformas de mensagens, o WhatsApp foi projetado como uma rede de comunicação *peer-to-peer* criptografada. Essa criptografia de ponta a ponta exige que toda a comunicação flua através de um "dispositivo" com um número de telefone exclusivo, com cada lado usando chaves para criptografar e descriptografar a carga útil. Os servidores do WhatsApp estão lá apenas para ajudar a rotear as mensagens, permanecendo incapazes de descriptografá-las ou analisá-las de qualquer forma ([MEYER, 2018](#)).

Dessa forma, a integração de um chatbot com o aplicativo requer uma plataforma que faça o meio de campo entre eles. Essa plataforma é chamada de orquestradora e se encarrega de gerenciar todas as interações do chatbot com o WhatsApp, de forma a

garantir que sejam seguidos todos os protocolos do aplicativo e abstrair a complexidade de integrar o chatbot diretamente com a rede do WhatsApp.

Um dos orquestradores de chatbot mais conhecidos para esse fim é o Twilio. Vários *frameworks* de chatbots já oferecem integração com o Twilio, como exemplo, todos os que foram mencionados anteriormente na seção 2.3.1. No entanto, a integração com este *middleware* oferece um obstáculo a mais, já que a obtenção de um número celular e as chamadas de API são um serviço pago.

2.3.2.2 Telegram

O Telegram é um aplicativo de mensagens instantâneas gratuito, disponível para dispositivos móveis e computadores, baseado em nuvem. Atualmente, conta com mais de 700 milhões de usuários ativos mensais e está entre os 10 aplicativos mais baixados globalmente (TELEGRAM, 2023b). Segundo a pesquisa Panorama, realizada por MobileTime e OpinionBox (2023), o Telegram ocupa a quarta posição dentre os *apps* de mensageria mais utilizados no Brasil.

Uma característica distintiva do Telegram é a possibilidade de utilizar nomes de usuário, além de números de telefone, o que facilita a criação de chatbots, eliminando a necessidade de possuir um número de celular. Dentro do contexto de chatbots, a plataforma de bots do Telegram hospeda mais de 10 milhões de bots de forma gratuita (TELEGRAM, 2023a).

A integração de chatbots com o Telegram é feita por meio do BotFather, o bot oficial da plataforma destinado à criação e gerenciamento de outros bots.

O BotFather oferece diversos comandos para essa finalidade, abrangendo desde a criação e edição até a configuração de bots, bem como a integração de aplicativos web e jogos que podem ser incorporados ao bot no Telegram. Além disso, o Telegram disponibiliza uma API para o desenvolvimento de bots na plataforma (TELEGRAM, 2023c).

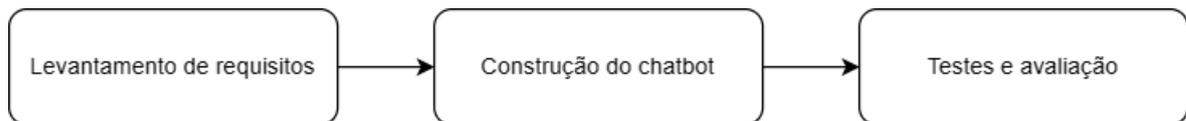
Diferentemente do WhatsApp, o Telegram dispensa a necessidade de um orquestrador para integrar-se a um chatbot, uma vez que a própria plataforma Telegram oferece uma API direta para o desenvolvimento de bots. Devido a essa facilidade e a gratuidade para implantar o chatbot desenvolvido neste trabalho, a plataforma escolhida foi o Telegram.

Este capítulo abordou conceitos essenciais para a fundamentação teórica deste trabalho. O conhecimento apresentado aqui será empregado para auxiliar a compreensão da solução proposta no Capítulo 3, referente à metodologia.

3 Metodologia

Neste capítulo serão descritos os procedimentos realizados para construção do chatbot. A Figura 2 apresenta as etapas da metodologia, que são: levantamento dos requisitos, construção do chatbot e testes e avaliação dos resultados.

Figura 2 – Etapas da metodologia para produção do chatbot.



Fonte: acervo do autor.

3.1 Levantamento de Requisitos

Em primeiro lugar, foi feita uma avaliação entre as alternativas de plataformas para o desenvolvimento de chatbots disponíveis no mercado, em que foram analisadas três principais opções: IBM Watson ([IBM, 2023b](#)), reconhecida por sua interface gráfica intuitiva e fácil usabilidade, embora seu plano gratuito demande uma rigorosa avaliação de cadastro do desenvolvedor; RASA ([RASA, 2023](#)), que dispõe de uma versão gratuita, mas carece de uma interface gráfica desenvolvida para o programador; e Dialogflow([GOOGLE, 2023c](#)), que apresenta uma interface gráfica intuitiva e oferece uma versão de teste gratuita, passível de cobrança apenas após exceder um limite preestabelecido de consumo. O Dialogflow foi eleito como a escolha preferencial por ser um produto da Google, permitindo fácil integração com outras ferramentas da empresa. Além disso, destaca-se pela presença de uma comunidade ativa e pelo suporte regular oferecido.

A comunicação entre o AdaBot e o usuário foi estabelecida por meio da plataforma de mensagens Telegram. A escolha pelo Telegram se baseou na sua integração direta com o Dialogflow, em oposição ao WhatsApp, que requer um orquestrador para mediar a comunicação com o chatbot, como por exemplo, o Twilio ([TWILIO, 2023](#)), sendo este um serviço sujeito a custos.

Em sequência, buscou-se os tópicos a serem abordados pelo chatbot, os quais foram definidos com base em informações selecionadas em fontes como os *websites* da Coordenação de Ciência da Computação (COCOM), o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), bem como os documentos oficiais do curso, incluindo o Projeto Político Pedagógico vigente do Curso, instituído pela RESOLUÇÃO Nº 1932-CONSEPE de 27 de setembro de 2019 e os funcionários da própria coordenação do curso.

Para facilitar a interação com o usuário, buscou-se categorizar as informações coletadas em tópicos, que serão abordados pelo AdaBot.

A seleção desses tópicos foi fundamentada na frequência com que os estudantes do curso de Ciência da Computação buscam informações relacionadas a eles, dada a essencialidade dessas informações para o avanço do discente na graduação.

Após consulta à coordenação, constatou-se que as demandas mais frequentes estão relacionadas a normas, instruções e, no caso de componentes curriculares, à necessidade de formulários. Importante salientar que temas como TCC e estágio integram a estrutura curricular, demandando frequentemente o preenchimento de formulários específicos para sua concretização. Assim, o chatbot foi estruturado em 7 tópicos, a saber: TCC, estágio, aproveitamento de estudos, atividades complementares, plano de estudos, colação de grau e trancamento/cancelamento.

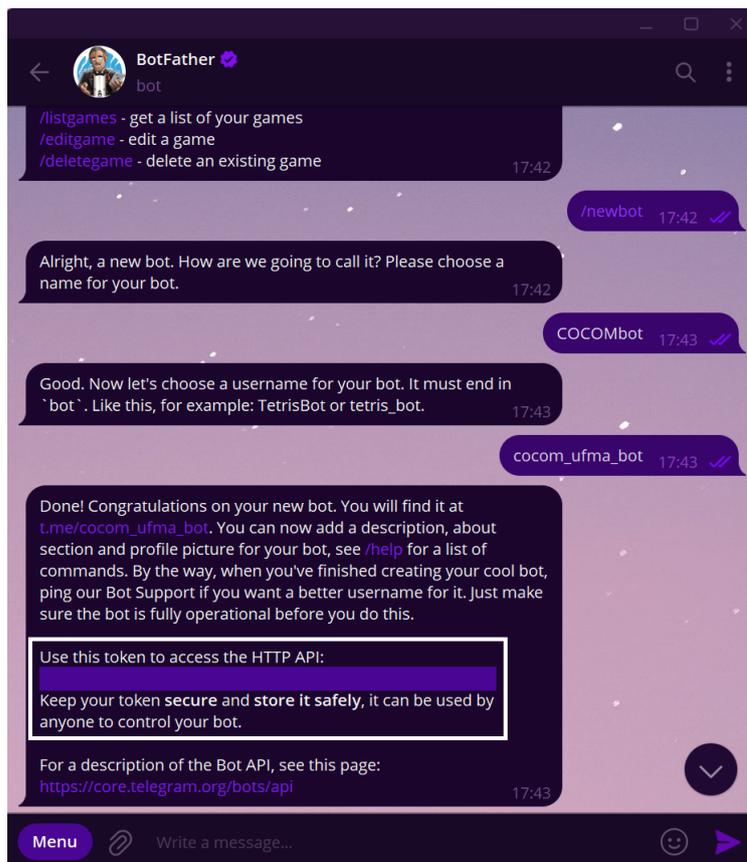
Com toda a estrutura do projeto do chatbot definida, foi possível dar início à etapa de implementação do chatbot propriamente dito.

3.2 Construção do Chatbot

Inicialmente, é preciso criar um bot no Telegram e obter seu *token*, que funciona como uma chave de acesso para o bot na plataforma. Posteriormente, esse *token* é inserido na aba de integração do agente virtual no Dialogflow, estabelecendo a conexão do bot.

Para criar um bot no Telegram, é necessário acessar a plataforma de mensagens por meio do aplicativo ou site e localizar o BotFather, seja por meio de uma pesquisa ou utilizando o link direto. Ao iniciar o bot com o comando `/start`, são apresentadas opções para criação, edição e configuração de bots. O comando `/newbot` inicia o processo de criação, solicitando o nome e o usuário desejados para o novo chatbot, sendo este último obrigatoriamente finalizado em `"bot"`.

Após a conclusão desse procedimento, o BotFather fornece um link direto para acessar o bot recém-criado no Telegram, acompanhado de um *token* de acesso para a API HTTP da plataforma. Esse *token* é o elemento fundamental que vincula o bot criado no Telegram ao agente virtual estabelecido no Dialogflow, como mostra a Figura 3.

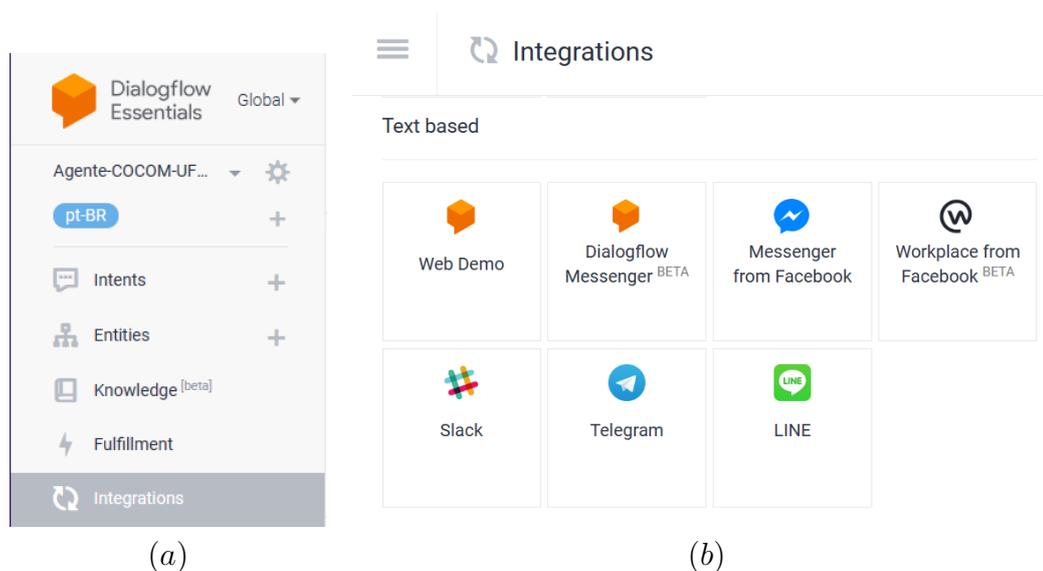
Figura 3 – *Token* de acesso à API HTTP do Telegram criado pelo BotFather.

Fonte: acervo do autor.

Ainda por meio do BotFather é possível editar o nome do bot criado, a descrição e as informações sobre o chatbot, a foto do perfil e os comandos associados ao bot.

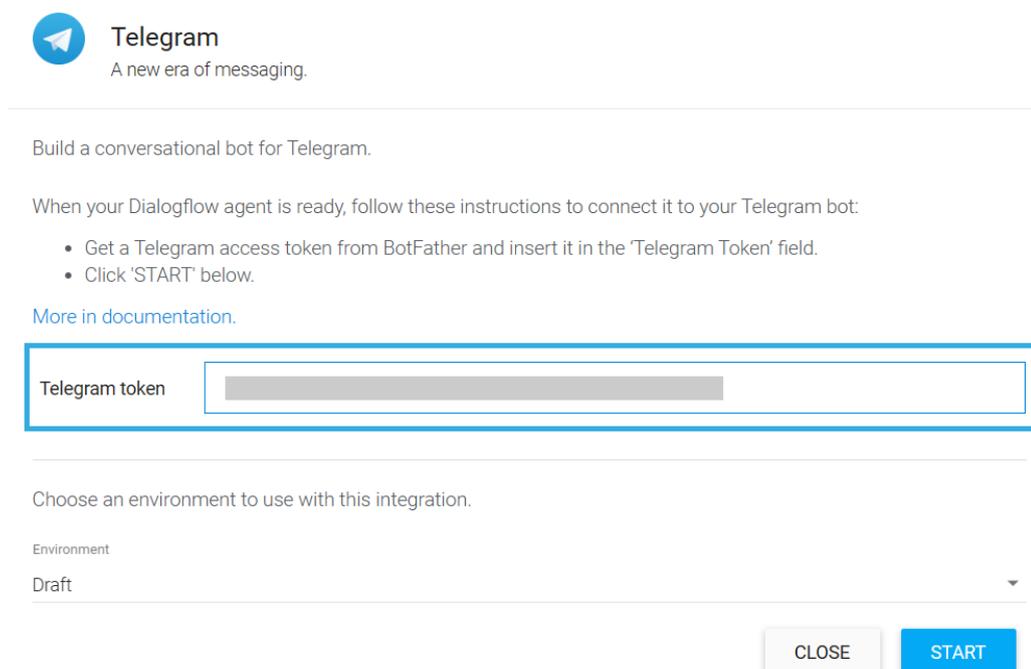
Em seguida, para criar um agente no Dialogflow, é necessário acessar o *website* do Dialogflow estando logado em uma conta Google e fornecer as informações da conta, como o tipo de conta (pessoa física ou jurídica) e os dados fiscais. Após completar esta etapa, o agente deve ser nomeado, sua linguagem padrão deve ser definida como português-Brasil e ao finalizar, será criado um Google Project. Uma vez criado o agente, é possível proceder com a integração do chatbot com o Telegram. Para realizar a integração entre o agente do Dialogflow e o Telegram, é necessário acessar a aba “*Integrations*” no menu lateral, como mostra a Figura 4.

Figura 4 – Menu lateral e menu de integrações no Dialogflow.



Fonte: acervo do autor.

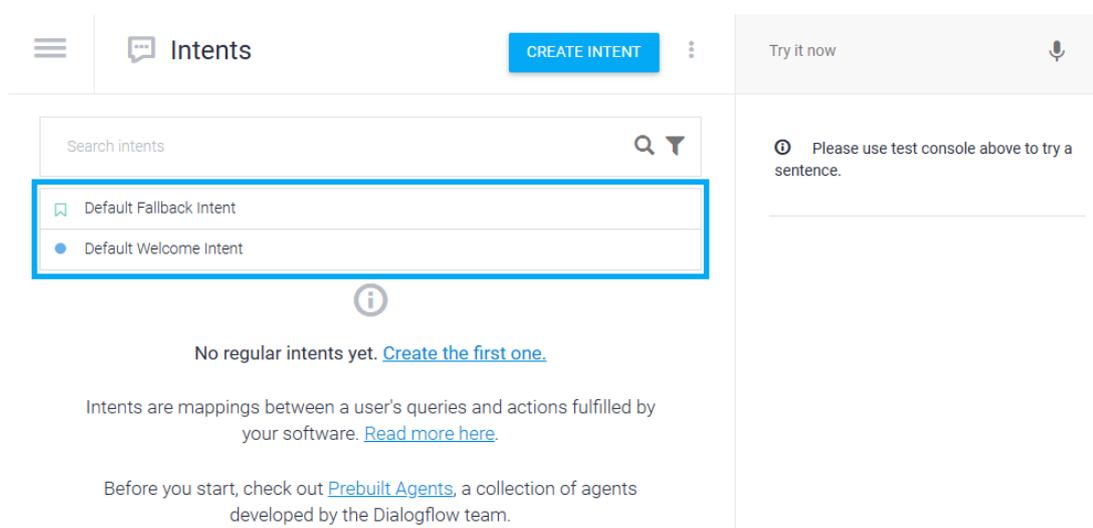
Como mencionado anteriormente, são oferecidas várias opções de integrações e, nesse caso, foi selecionado o Telegram. O pop-up solicita a inserção de um *token* do Telegram, o qual foi obtido anteriormente por meio do BotFather. Ao inserir o *token* e clicar no botão “Start”, o Dialogflow irá se conectar ao bot criado no Telegram (Figura 5).

Figura 5 – Pop-up de integração do Dialogflow com o Telegram para inserção do *token*.

Fonte: acervo do autor.

Por padrão, o agente criado possui duas *intents* iniciais, representadas na Figura 6.

- "*Default Fallback Intent*" é "*intent* padrão de recuo". Esta *intent* é acionada quando o agente não encontra correspondência entre a expressão do usuário final e as demais *intents*, por isso, essa *intent* não recebe frases de treinamento e está configurada com a ação "input.unknown".
- "*Default Welcome Intent*" é "*intent* padrão de boas vindas". Esta *intent* por padrão vem com frases de treinamento de cumprimento ao usuário e deve ser usada como *intent* inicial.

Figura 6 – *Intents* padrões de um agente do Dialogflow.

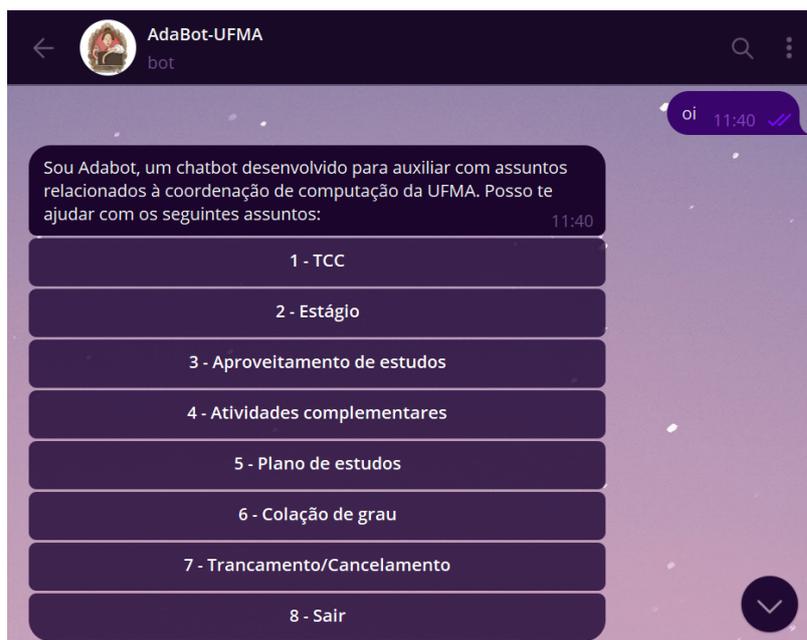
Fonte: acervo do autor.

Para a elaboração do chatbot neste projeto, optou-se por adotar uma abordagem baseada em estrutura de árvore de decisão para a definição dos fluxos de conversação. Essa escolha foi motivada pela facilidade em direcionar os usuários às informações desejadas e pela sua intuitividade.

Cada um desses fluxos representa os tópicos anteriormente definidos na fase de levantamento dos requisitos, que por sua vez, devem apresentar normas, instruções e, com exceção de “colação de grau” e “trancamento/cancelamento”, também devem apresentar o formulário para cumprimento ou inscrição no componente curricular.

Dessa forma, ao primeiro contato com o chatbot, ele deverá apresentar os tópicos que foram separados anteriormente pela sua relevância para os alunos. Os tópicos foram separados na forma como são exibidos no menu principal, representado na Figura 7.

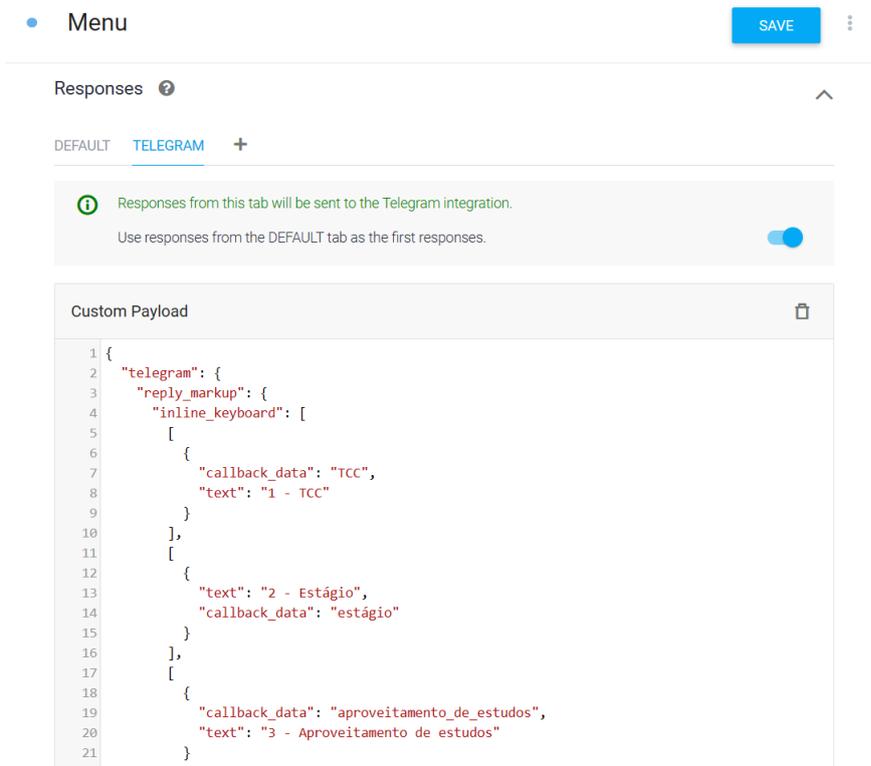
Figura 7 – Menu principal do AdaBot.



Fonte: acervo do autor.

Para isso, primeiro foi preciso modificar a *intent* padrão de boas-vindas para exibição do menu na Figura 7. Neste caso foi criada uma nova *intent* chamada “Menu”, que será chamada em sequência à *intent* de boas-vindas. A *intent* “Menu” possui como resposta um *custom payload*¹ específico para o Telegram, onde foi incluído um código JSON (JSON, 2023) que determina como será a saída do chatbot (Figura 8). Neste caso será um texto de cumprimento ao usuário e um botão para cada um dos tópicos definidos. Cada botão, por sua vez, possui uma *callback data*, que é o valor que será retornado internamente ao ter o botão apertado.

¹ O *custom payload* é um tipo de resposta avançada não-padrão no Dialogflow, sendo sua personalização realizada em formato JSON. Esse tipo de resposta é processado pela plataforma específica na qual está sendo implementado.

Figura 8 – Resposta do tipo *custom payload* para *intent* Menu.

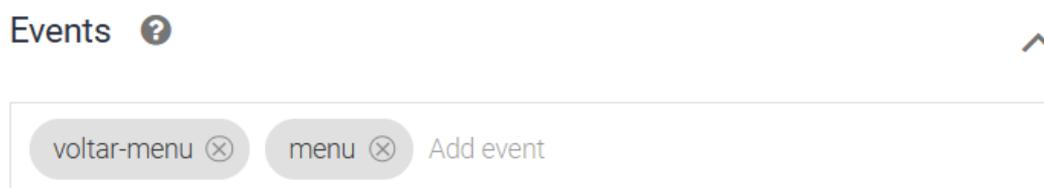
Fonte: acervo do autor.

A partir dessas definições, foram criadas *intents* para cada tópico. Para que o agente identifique qual opção do menu foi selecionada, as *callback datas* definidas anteriormente foram declaradas como frase de treinamento em suas respectivas *intents*. A Tabela 2 representa os respectivos tópicos selecionados com seus textos de botão, *callback data* e *intents*.

Tabela 2 – Relação entre botões, *callback data* e nomes de *intents* do menu principal.

Texto de botão	Callback data	Intent
1 - TCC	TCC	TCC
2 - Estágio	estagio	Estagio
3 - Aproveitamento de estudos	aproveitamento_de_estudos	AproveitamentoEstudos
4 - Atividades complementares	atividades_complementares	AproveitamentoComplementares
5 - Plano de estudos	plano_de_estudos	PlanoEstudos
6 - Colação de grau	colação_de_grau	ColacaoDeGrau
7 - Trancamento/Cancelamento	trancamento	Cancelamento
8 - Sair	finalizar_conversa	-

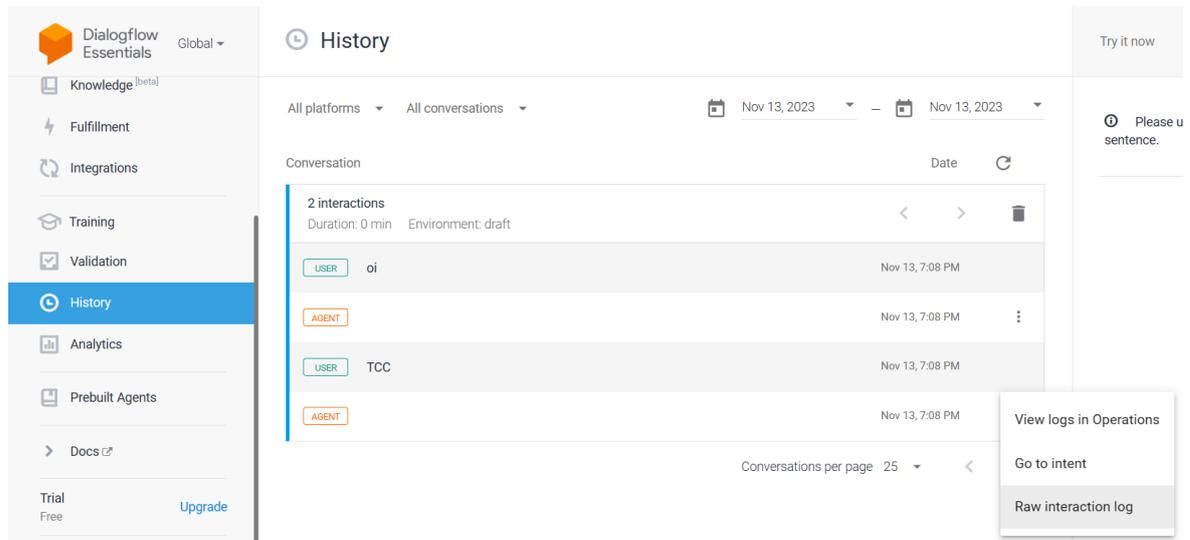
A *intent* “Menu” também precisa ser chamada novamente para alguns subfluxos, a fim de voltar ao menu principal e exibi-lo. Por isso, essa *intent* foi declarada como evento (Figura 9).

Figura 9 – Seção para configuração de eventos na *intent* Menu.

Fonte: acervo do autor.

Utilizando como referência a Figura 7, será demonstrado aqui o funcionamento do fluxo "TCC" para exemplificar o funcionamento da navegação entre as estruturas do chatbot. No menu principal, ao selecionar o botão referente a este fluxo, o Dialogflow retorna em seu log internamente que a *intent* identificada é “TCC”. Este log pode ser observado na aba lateral *History* e o seu log de interações pode ser consultado para cada interação, conforme mostrado na Figura 10.

Figura 10 – Aba de históricos de conversas com o chatbot no Dialogflow.



Fonte: acervo do autor.

Como observado no log de interações da Figura 11, foi configurado um contexto de saída para a *intent* TCC, que foi nomeado *tcc-followup*. Isto permite que os subfluxos que estiverem configurados com *tcc-followup* como contexto de entrada apenas possam ser ativados quando este pré-requisito for cumprido, além das frases de treinamento. As *intents* subsequentes, ou seja, referentes aos subfluxos, também possuem este contexto como entrada e saída. Por fim, ao *lifespanCount* foi dado o valor 1, para que este contexto tenha duração curta e não interfira em outros fluxos. A configuração de contexto é mostrada na Figura 12.

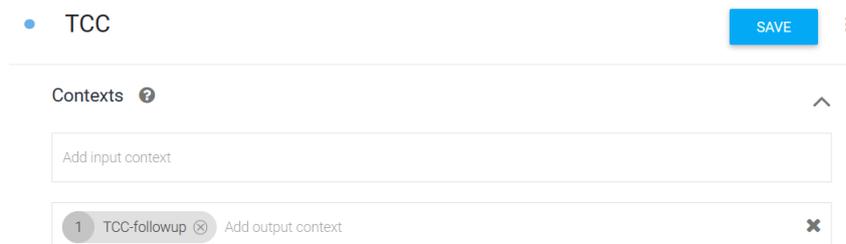
Figura 11 – *Raw interaction log* de uma interação em formato JSON.

Raw interaction log

```
48     "platform": "TELEGRAM"
49   },
50   {
51     "text": {
52       "text": [
53         ""
54       ]
55     }
56   }
57 ],
58 "webhookStatus": {
59   "webhookUsed": false
60 },
61 "outputContexts": [
62   {
63     "lifespanCount": 1,
64     "name": "tcc-followup",
65     "parameters": {
66       "number.original": "",
67       "number": ""
68     }
69   }
70 ],
71 "intent": {
72   "isFallback": false,
73   "displayName": "TCC",
74   "id": "e6bcc891-946f-4dc2-899f-9ce87833db89"
75 }
76 }
```

CLOSE COPY

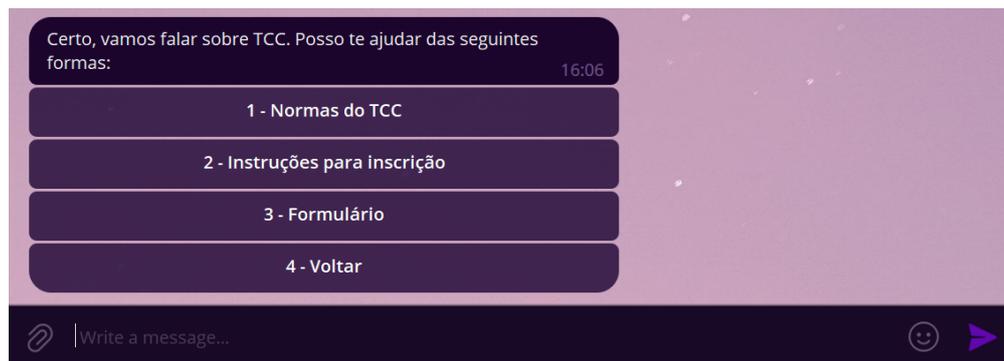
Fonte: acervo do autor.

Figura 12 – Contexto configurado para a *intent* TCC.

Fonte: acervo do autor.

Seguindo o mesmo fluxo de TCC e os requisitos de informações que devem ser oferecidas pelo chatbot, esse tópico deve apresentar normas, instruções e o formulário para inscrição neste componente curricular, além de um botão para permitir que o usuário volte ao menu anterior. Assim foi criada a tela representada na Figura 13, que foi configurada como resposta para a *intent* TCC.

Figura 13 – Menu com assuntos relacionados ao tópico "TCC".



Fonte: acervo do autor.

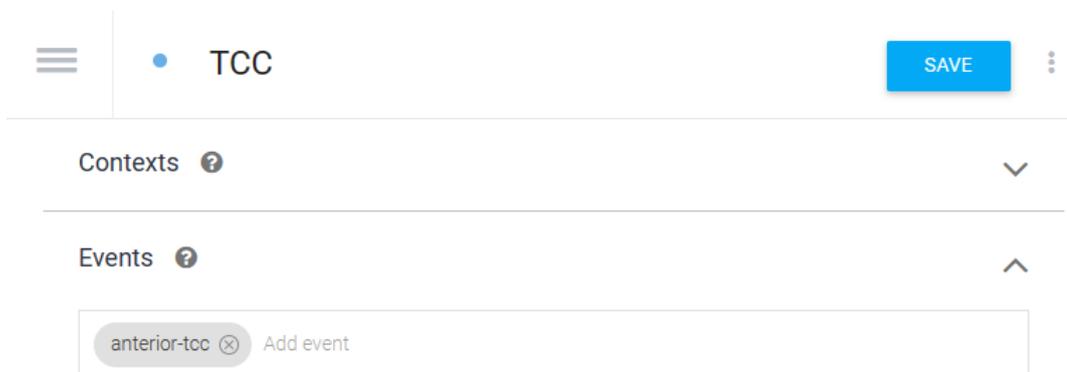
Para cada botão foi criada uma nova *intent*, onde foi seguida uma nomenclatura padrão para identificar cada fluxo principal, como mostra a tabela 3.

Tabela 3 – Tabela relacionando botões, *callback data* e nomes de *intents* para o menu do tópico TCC.

Texto de botão	Callback data	Intent
1 - Normas do TCC	tcc_normas	TCC.normas
2 - Instruções para inscrição	tcc_instrucoes	TCC.instrucoes
3 - Formulário	tcc_formulario	TCC.formulario
4 - Voltar	tcc_voltar_menu	TCC.voltar

Devido à capacidade dos subfluxos de retrocederem ao fluxo anterior, a *intent* TCC foi designada como um evento (Figura 14), permitindo, dessa forma, a sua invocação por meio do código de *fulfillment*.

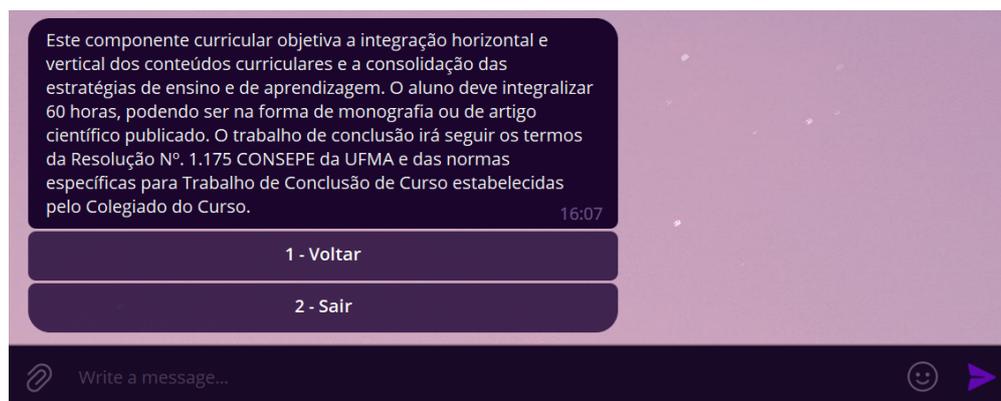
Figura 14 – O evento configurado para *fulfillment* na *intent* TCC.



Fonte: acervo do autor.

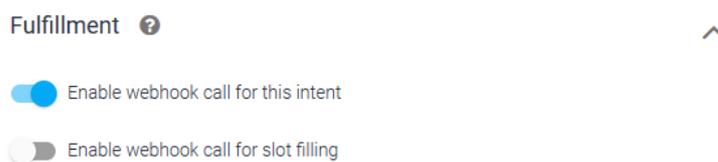
Seguindo como exemplo o botão “Normas do TCC”, este fluxo chega ao fim na Figura 15, em que, ao exibir as normas, oferece a informação principal e um botão para voltar ao menu anterior e outro para sair.

Figura 15 – Resumo das normas do TCC apresentado pelo chatbot.



Fonte: acervo do autor.

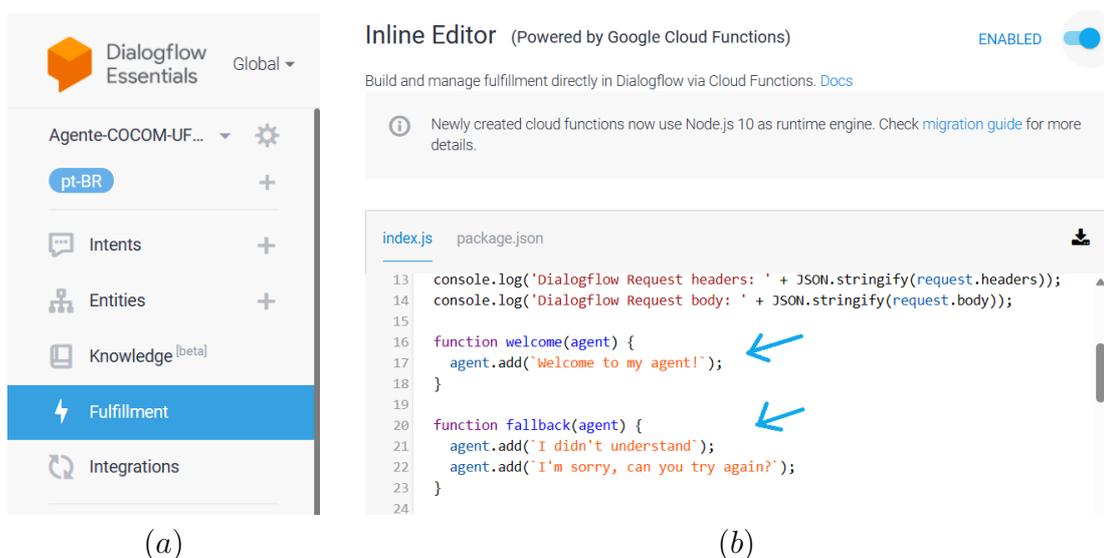
Os botões “voltar” foram desenvolvidos com a utilização de *fulfillment* no Dialogflow, pois requerem a implementação de uma lógica para acionar os eventos correspondentes. A fim de permitir a execução da lógica de *fulfillment*, é necessário ativar esta opção na *intent* relevante, como mostra a Figura 16.

Figura 16 – Ativação de *fulfillment* dentro da intent.

Fonte: acervo do autor.

A habilitação e a configuração da lógica de *fulfillment* no Dialogflow foi feita através da guia *Fulfillment* e ativação do editor *inline*, conforme Figura 17. Inicialmente, esse editor inclui um código de exemplo que incorpora uma lógica para as *intents* padrão mencionadas anteriormente, ou seja, *Default Fallback Intent* e *Default Welcome Intent*.

As lógicas foram implementadas por meio da criação de funções correspondentes para cada *intent*. Cada função recebe o agente como parâmetro, o qual foi instanciado como *dialogflow-fulfillment* (Figura 18). Dessa maneira, é possível empregar os métodos disponíveis nesta biblioteca. Um exemplo é o método *setFollowupEvent*, utilizado no contexto atual para chamada de eventos configurados em *intents*, que requer como parâmetro o evento a ser invocado.

Figura 17 – Menu lateral e menu de *fulfillment* no Dialogflow.

Fonte: acervo do autor.

Figura 18 – Agente instanciado como *fulfillment* do Dialogflow.

Fonte: acervo do autor.

No exemplo mencionado anteriormente, no qual o final do fluxo TCC inclui um botão de retorno, o evento *anterior-tcc* foi configurado para a *intent* TCC na Figura 14 e também como mostrado no código da Figura 19. Com essa configuração, ao chamar este método, não há necessidade de correspondência às frases de treinamento associadas a essa *intent*.

O método *setFollowupEvent* também foi utilizado para implementar a funcionalidade de encerrar o chat e sair. Nesse caso, ele aciona a *intent* "Ajuda", previamente configurada com o evento "ajuda". Quando o usuário clica no botão "Sair", o chatbot pergunta ao usuário se ainda precisa de ajuda, oferecendo dois botões: um para a resposta afirmativa e outro para a negativa (Figura 20). Se o botão afirmativo for pressionado, o

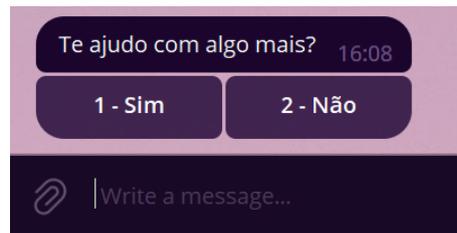
Figura 19 – Chamada do evento *anterior-tcc* por meio do método *setFollowupEvent*.

```
index.js  package.json
55     agent.setFollowupEvent("voltar-menu");
56   }
57
58   function tcc_anterior(agent) {
59     agent.add(`);
60     agent.setFollowupEvent("anterior-tcc");|
61   }
62
```

Fonte: acervo do autor.

usuário é redirecionado para o menu principal, previamente configurado como evento. No entanto, se a resposta for negativa, será invocada a *intent* "Ajuda.nao", configurada como encerramento da conversa.

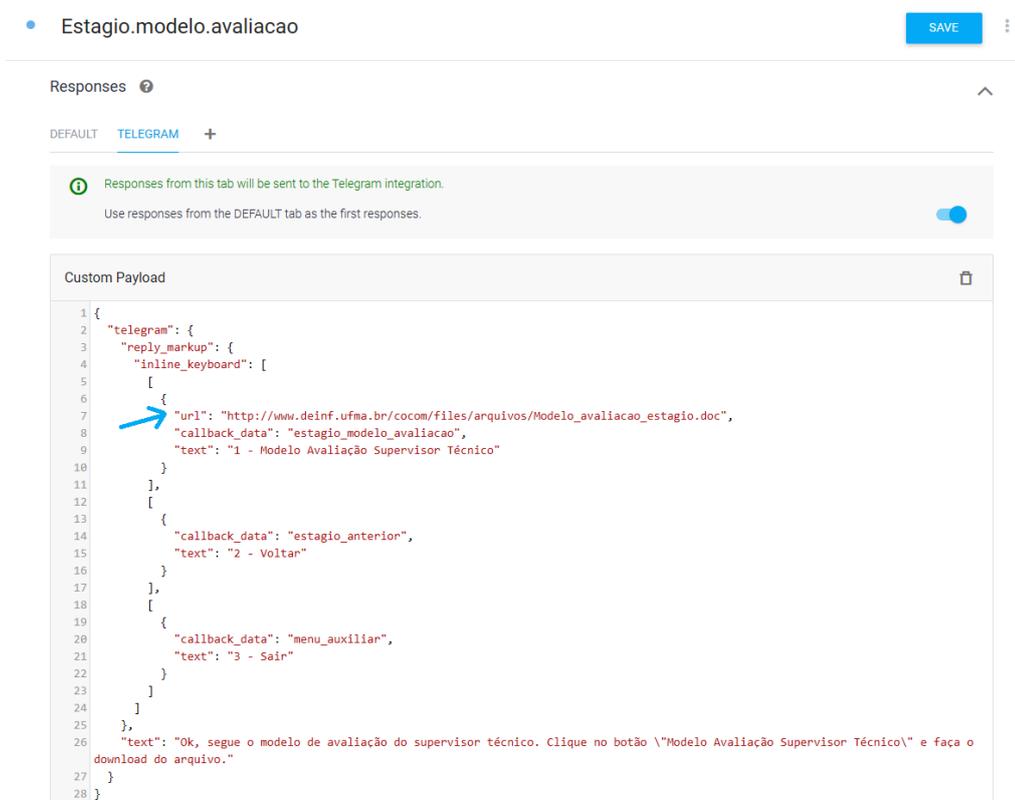
Figura 20 – Botões para continuar a conversa ou finalizá-la.



Fonte: acervo do autor.

Todos os demais tópicos que formam a Adabot seguem um fluxo semelhante ao descrito acima, com a exceção de fluxos específicos que incluem links para download de documentos. Um exemplo é o fluxo "Estágio", que contém o subfluxo "Modelos de documentos". Este subfluxo disponibiliza um modelo de documento para avaliação do supervisor de estágio e um modelo de relatório para aproveitamento do estágio. Ao selecionar qualquer uma das alternativas, o chatbot apresenta 3 botões: um para o download do arquivo modelo, outro para voltar (conforme o padrão anteriormente especificado) e um botão para sair, cujo funcionamento foi explicado anteriormente.

Para que o chatbot possa oferecer um link em um botão específico, foi adicionado um novo parâmetro chamado "url" no *custom payload* da *intent* correspondente, apresentado na Figura 21. Esse parâmetro recebe o link que o usuário poderá acessar ao clicar no botão.

Figura 21 – Resposta do tipo *custom payload* para *intent* de modelo de avaliação de estágio.

Fonte: acervo do autor.

Após a conclusão do desenvolvimento da solução, podemos realizar uma análise do que foi realizado. Ficou evidente como a ferramenta Dialogflow facilitou o desenvolvimento do chatbot, proporcionando uma interface intuitiva e uma curva de aprendizado razoável para sua utilização. No contexto da implementação das funcionalidades descritas neste trabalho, o Dialogflow atendeu às expectativas, apresentando poucas limitações. A principal restrição identificada neste projeto foi a impossibilidade de chamar um evento diretamente após o outro, conforme ilustrado na Figura 21. Para contornar essa limitação, foi necessário implementar um botão adicional, permitindo que o chatbot questionasse o usuário sobre a necessidade de assistência. A chamada de um evento ainda está condicionada à interação do usuário, o que significa que o chatbot continua dependendo da iniciativa do usuário.

Adicionalmente, a integração da solução com a plataforma de mensagens Telegram mostrou-se altamente eficiente, facilitando a implantação do agente do Dialogflow no bot. Além disso, o Telegram oferece diversas opções de customização de respostas, contribuindo para a flexibilidade do chatbot.

3.3 Testes e Avaliação

A fase final da metodologia foi dedicada aos testes e a avaliação do chatbot por parte dos usuários. Assim, foi elaborado um formulário destinado à coleta de *feedback* sobre a experiência de utilização do AdaBot. A ferramenta escolhida para a criação da pesquisa foi o aplicativo Google Forms (GOOGLE, 2023f).

O objetivo desse formulário é compreender os pontos positivos e negativos do chatbot implementado, além de captar sugestões para possíveis melhorias. O formulário foi disponibilizado para indivíduos com alguma relação com o curso de Ciência da Computação. Ao todo, foram coletadas 14 respostas.

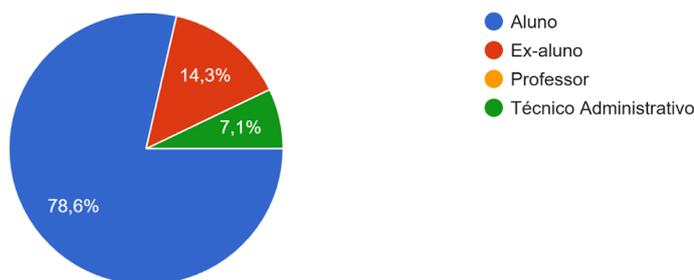
Para identificar o perfil dos respondentes, a primeira pergunta do formulário buscou esclarecer a natureza de sua relação atual com o Curso de Ciência da Computação da UFMA. As opções de resposta incluem aluno, ex-aluno, professor ou técnico administrativo.

Conforme visto na Figura 22, das 14 respostas colhidas, 78,6% responderam que são alunos, ou seja, 11 alunos testaram o AdaBot e responderam a pesquisa. 14,3% das respostas foram de ex-alunos, ou seja, 2 ex-alunos. Finalmente, 7,1% das respostas é de um técnico administrativo, ou seja, 1 pessoa, e não houveram respostas de professores.

Figura 22 – Gráfico de pizza para ilustrar respostas à pergunta 1 do formulário de *feedback*.

Qual a sua relação atual com o Curso de Ciência da Computação da UFMA?

14 respostas



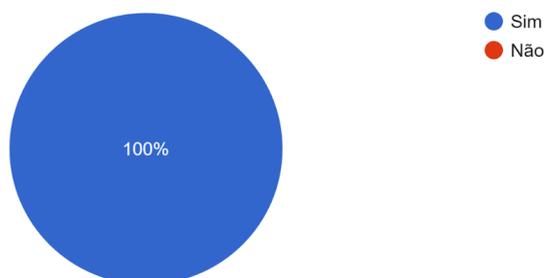
Fonte: acervo do autor.

Em seguida, com o intuito de avaliar a utilidade das informações fornecidas pelo chatbot, os usuários foram questionados: “Com relação à relevância dos tópicos (assuntos abordados), você já precisou, em algum momento, de alguma das informações disponibilizadas pelo AdaBot?” Como visto no gráfico da Figura 23, todos os respondentes afirmaram que sim, isto é, já precisaram em algum momento de alguma das informações que estão disponíveis no AdaBot.

Figura 23 – Gráfico de pizza para ilustrar respostas à pergunta 2 do formulário de *feedback*.

Com relação à relevância dos tópicos (assuntos abordados), você já precisou, em algum momento, de alguma das informações disponibilizadas no AdaBot?

14 respostas



Fonte: acervo do autor.

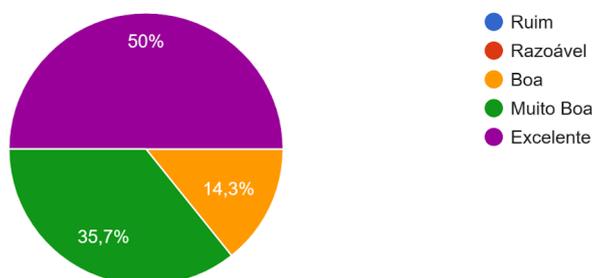
Com o objetivo de aferir se o chatbot foi eficaz ao apresentar suas informações de maneira intuitiva, as perguntas formuladas foram: “Como você avalia a facilidade para encontrar as informações buscadas no AdaBot?” e “Como você analisa o nível de facilidade apresentado para a interação com o AdaBot?”

Para esta primeira pergunta, mostrada no gráfico da Figura 24, metade dos respondentes (7 pessoas) avaliaram a facilidade para encontrar as informações no chatbot como excelente. 35,7% (5 pessoas) avaliaram como muito boa e 14,3% (2 pessoas) avaliaram como boa.

Figura 24 – Gráfico de pizza para ilustrar respostas à pergunta 3 do formulário de *feedback*.

Como você avalia a facilidade para encontrar as informações buscadas no AdaBot?

14 respostas



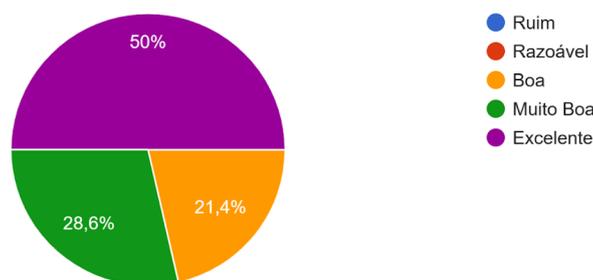
Fonte: acervo do autor.

Para a pergunta seguinte, apresentada no gráfico da Figura 25, metade dos que responderam também avaliaram a facilidade de interação com o chatbot como excelente,

28,6% responderam que é muito boa (4 pessoas) e 21,4% (3 pessoas) responderam que é boa.

Figura 25 – Gráfico de pizza para ilustrar respostas à pergunta 4 do formulário de *feedback*.

Como você analisa o nível de facilidade apresentada para a interação com o AdaBot?
14 respostas

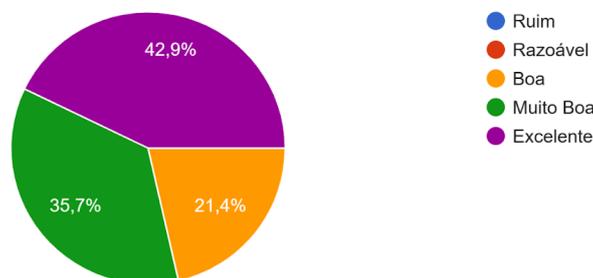


Fonte: acervo do autor.

Em seguida, a capacidade de apresentar as informações oferecidas de forma clara foi avaliada por meio da pergunta, “Como você classifica a clareza da linguagem apresentada para descrever as informações?”, em que os resultados são exibidos no gráfico da Figura 26. 42,9% avaliaram a clareza da linguagem como excelente, ou seja, 6 pessoas. 35,7% (5 pessoas) avaliaram como muito boa e 21,4% dos usuários (3 pessoas) responderam que a linguagem apresentada é boa.

Figura 26 – Gráfico de pizza para ilustrar respostas à pergunta 5 do formulário de *feedback*.

Como você classifica a clareza da linguagem apresentada para descrever as informações?
14 respostas



Fonte: acervo do autor.

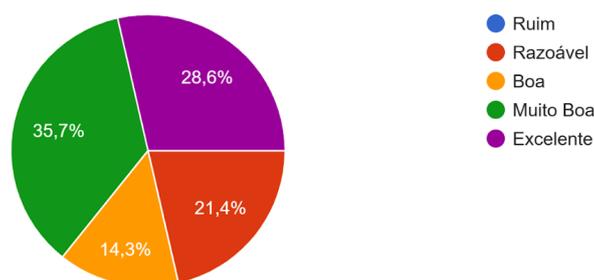
A fim de verificar a relevância das informações apresentadas no chatbot e se as mesmas satisfazem aqueles que o testaram, foi perguntado aos usuários, “Como você avalia a sua satisfação com relação às informações apresentadas em cada assunto para esclarecer suas dúvidas?” O gráfico da Figura 27 mostra que desta vez, 28,6% responderam que as

informações eram excelentes, ou seja, 4 pessoas. Para esta pergunta, a maioria (5 pessoas, 35,7%) responderam que as informações eram muito boas e 14,3% (2 pessoas) classificaram as informações como boas e 3 pessoas (21,4%) classificaram como razoável.

Figura 27 – Gráfico de pizza para ilustrar respostas à pergunta 6 do formulário de *feedback*.

Como você avalia a sua satisfação com relação às informações apresentadas em cada assunto para esclarecer suas dúvidas?

14 respostas



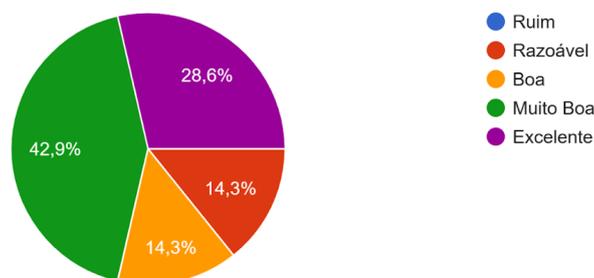
Fonte: acervo do autor.

A última pergunta de múltipla escolha do questionário foi feita para entender a avaliação dos usuários ao chatbot como um todo. “Que avaliação geral você dá ao AdaBot?” foi a pergunta feita, em que 28,6% (4 pessoas) responderam que acharam o AdaBot excelente, enquanto a maioria (6 pessoas, 42,9%) classificou-o como muito bom. Uma quantidade igual de usuários achou o chatbot bom e razoável, ou seja, 2 pessoas (14,3%) para cada classificação. O gráfico da Figura 28 mostra o resultado obtido nessa questão.

Figura 28 – Gráfico de pizza para ilustrar respostas à pergunta 7 do formulário de *feedback*.

Que avaliação geral você dá ao AdaBot?

14 respostas



Fonte: acervo do autor.

A partir das respostas para as perguntas de múltipla escolha, foi possível perceber que em geral o AdaBot foi bem recebido, mas que ainda há espaço para melhoria no conteúdo oferecido pelo chatbot, de forma que uma variedade maior de informações sejam disponibilizadas e que as informações já apresentadas atualmente podem ser refinadas para apresentar maior clareza e facilitar ainda mais o entendimento.

Essa conclusão é ainda mais aparente ao verificarmos as respostas às questões abertas do formulário, mostradas na sequência.

“Destaque os pontos positivos do Bot.” Esta questão visou entender a satisfação do usuário com o chatbot como um todo e identificar os aspectos positivos que devem ser mantidos e se os objetivos estabelecidos na fase de levantamento de requisitos foram cumpridos. Assim sendo, algumas das respostas para essa questão incluem:

- “Muito fácil de selecionar o tópico que eu quero saber sobre, cabe na palma da minha mão, não é como ter que entrar no site da UFMA para conseguir informações simples como essas.”
- “Ele compila as informações necessárias que muitas vezes os alunos têm dificuldades de obter, especialmente sobre aproveitamento e colação de grau, que exigem um pouco mais de atenção e documentação. Isso, aliado a um menu interativo, torna bem mais fácil para o aluno resolver essas questões.”
- “Centraliza as informações relevantes e direciona pros canais necessários.”

Para a questão seguinte, “Destaque aspectos que poderiam ser aperfeiçoados”, foram feitas algumas observações do que os usuários que testaram o chatbot esperam como melhorias:

- “Eu acredito que falta algumas informações que poderiam ser disponibilizadas como, por exemplo o calendário acadêmico, que eu mesma demoro um pouco pra encontrar no site da UFMA e um mapa da UFMA pra ajudar os calouros nessa tarefa de descobrir onde fica cada coisa de cada prédio.”
- “Mais informações sobre estágio como os procedimentos mais detalhados, quantidade de horas mínimas para aproveitamento, etc.”
- “Nas informações sobre TCC deveria também constar sobre os depósitos, registros na biblioteca e todos os trâmites de conclusão.”
- “Poderia disponibilizar informações mais completas.”

A partir das respostas acima colocadas, percebe-se que existe a demanda de inclusão de mais informações ao chatbot, não apenas sobre assuntos relacionados à Coordenação

de Computação em si, mas à Universidade como um todo. Existe também a necessidade de oferecer mais detalhes para algumas informações, como exemplificado por alguns dos usuários, que gostariam de obter mais explicações sobre o estágio e sobre os trâmites para conclusão do curso.

- “Poderia ter uma opção de sugestões no menu para que os usuários pudessem oferecer possíveis solicitações de melhoria ou até mesmo compartilhar conhecimento, *tips and tricks*. Com isso, o responsável pelo sistema teria a possibilidade de obter um *feedback* mais acurado da experiência do usuário e também teria contribuições em informação para adicionar ao Bot.”

Este comentário sugere que o chatbot ofereça um canal de comunicação entre os usuários e os responsáveis pelo sistema, para que a coleta de *feedback* seja contínua, assim como a implementação de melhorias baseada em sugestões dos usuários.

O próximo comentário sugere uma melhoria na estrutura e interação com o chatbot:

- “Poderia ter uma opção de voltar direto para o menu principal, evitando solicitar tantos "voltar".”

A última questão aplicada no questionário destacou pontos que haviam sido levantados nas questões anteriores, como exemplo a necessidade de mais detalhamento do conteúdo apresentado e de melhorias estruturais. “Apresente, se quiser, outros comentários, críticas e sugestões.” Os comentários feitos foram:

- “Talvez colocar mais informações não só sobre o curso mas também da UFMA no geral.”
- “A interação, as opções de voltar e sair não foram implementadas devidamente, o que prejudicou a navegação de volta ao menu inicial.”
- “O chatbot é ótimo, irei usar bastante.”
- “O bot tem potencial.”

As respostas a todas as perguntas do formulário evidenciaram a importância de uma ferramenta como o AdaBot para auxílio a rotina acadêmica do curso, principalmente, para melhorar qualidade da interação da Coordenação do Curso com os seus discentes. Por outro lado, apesar de se mostrar promissora, percebeu-se que alguns pontos ainda merecerem melhorias antes que o bot seja disponibilizado para uso.

Esse capítulo descreveu a metodologia empregada na construção do AdaBot, que é o objeto desta monografia. No próximo capítulo serão feitas algumas conclusões, bem como sugestões de trabalhos futuros.

4 Conclusão

Este trabalho teve como objetivo a construção de um chatbot chamado AdaBot para auxiliar os alunos de ciência da computação da UFMA.

O propósito fundamental do bot desenvolvido consistiu em centralizar e disponibilizar de maneira intuitiva as informações essenciais relacionadas à coordenação do curso para, assim, simplificar a busca de informações pelos discentes, possibilitando que encontrem de maneira mais eficiente as informações desejadas. Dessa forma, a coordenação pode direcionar seus esforços para lidar com situações e solicitações mais específicas.

Para tanto, foram exploradas as plataformas utilizadas para a implementação e implantação do chatbot, destacando o Dialogflow e o Telegram, respectivamente.

Idealmente, o processo de desenvolvimento desse chatbot seria conduzido visando a implantação no WhatsApp, uma vez que essa plataforma de mensagens é prioritariamente utilizada no Brasil atualmente ([MOBILETIME; OPINIONBOX, 2023](#)). Ter o AdaBot disponível no WhatsApp proporcionaria maior facilidade de acesso para os usuários. No entanto, essa abordagem implicaria em desafios adicionais e custos mais elevados para a realização do projeto atual, devido à necessidade da utilização de um *framework* orquestrador (como Twilio) que realize a comunicação entre o Whatsapp e o Dialogflow.

Dessa forma, a alternativa viável foi a escolha do *framework* Dialogflow, integrado ao software de mensagens Telegram, possibilitando uma implementação praticamente sem custos e direcionando o foco do trabalho, principalmente, para a metodologia e a definição do escopo do projeto.

Por fim, aplicou-se um questionário aos potenciais usuários do AdaBot, buscando-se fazer uma avaliação geral de suas funcionalidades e, a partir desta, realizar ajustes no chatbot, visando atender de maneira mais eficaz às expectativas dos discentes.

Embora, a avaliação do AdaBot tenha sido promissora, observa-se que este trabalho possui algumas limitações, entre elas: a utilização exclusiva do Telegram como aplicativo de mensagens e o chatbot ser de domínio fechado.

Dessa maneira, sugere-se como trabalhos futuros:

- Adaptação do AdaBot para o aplicativo de mensagens WhatsApp.
- A possibilidade de o AdaBot aceitar perguntas abertas, não previstas dentro do domínio inicialmente definido.

Referências

- AMAZON. *Amazon Alexa Official Site: What is Alexa?* 2023. Acesso em: 4 dez. 2023. Disponível em: <<https://developer.amazon.com/pt-BR/alexa>>. Citado 2 vezes nas páginas 12 e 17.
- APPLE. *Siri - Apple (BR)*. 2023. Acesso em: 4 dez. 2023. Disponível em: <<https://www.apple.com/br/siri/>>. Citado 2 vezes nas páginas 12 e 17.
- BARUS, S. P.; SURIJATI, E. Chatbot with dialogflow for faq services in matana university library. *International Journal of Informatics and Computation*, v. 3, n. 2, p. 62–70, 2022. Citado na página 14.
- CASELI, H. d. M.; NUNES, M. d. G. V. Processamento de linguagem natural: conceitos, técnicas e aplicações em português. 2023. Citado na página 15.
- CHARACTER.AI. *Character.AI*. 2023. Acesso em: 9 dez. 2023. Disponível em: <<https://beta.character.ai/help?>> Citado na página 16.
- GOOGLE. *Bard*. 2023. Acesso em: 10 dez. 2023. Disponível em: <<https://bard.google.com/chat>>. Citado na página 16.
- GOOGLE. *Contextos | Dialogflow ES | Google Cloud*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://cloud.google.com/dialogflow/es/docs/contexts-overview?hl=pt-br>>. Citado na página 21.
- GOOGLE. *Dialogflow | Google Cloud*. 2023. Acesso em: 2 dez. 2023. Disponível em: <<https://cloud.google.com/dialogflow?hl=pt-BR>>. Citado 2 vezes nas páginas 19 e 25.
- GOOGLE. *Dialogflow | Google Cloud*. 2023. Acesso em: 5 dez. 2023. Disponível em: <https://cloud.google.com/dialogflow?hl=pt_br>. Citado na página 19.
- GOOGLE. *Dialogflow | Google Cloud*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://cloud.google.com/dialogflow/es/docs/integrations?hl=pt-br>>. Citado na página 19.
- GOOGLE. *Formulários Google: criador de formulários on-line | Google Workspace*. 2023. Acesso em: 2 dez. 2023. Disponível em: <<https://www.google.com/intl/pt-BR/forms/about/>>. Citado na página 41.
- GOOGLE. *Intents | Dialogflow ES | Google Cloud*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://cloud.google.com/dialogflow/es/docs/intents-overview?hl=pt-br>>. Citado na página 20.
- IBM. *IBM watsonx Assistant*. 2023. Acesso em: 10 dez. 2023. Disponível em: <<https://www.ibm.com/br-pt/products/watsonx-assistant>>. Citado na página 22.
- IBM. *IBM® Watson*. 2023. Acesso em: 29 nov. 2023. Disponível em: <<https://www.ibm.com/br-pt/watson>>. Citado na página 25.

- IRANI, P.; OJAMERUAYE, O.; KAPOOR, A. *Decoding Dialogflow by coding your own chatbot | by Ovo Ojameruaye | SFU Professional Computer Science | Medium*. 2020. Acesso em: 10 dez. 2023. Disponível em: <<https://medium.com/sfu-csmpm/decoding-dialogflow-by-coding-your-own-chatbot-e49253ac39a8>>. Citado na página 17.
- JSON. *JSON*. 2023. Acesso em: 2 dez. 2023. Disponível em: <<https://www.json.org/json-pt.html>>. Citado na página 30.
- KHANNA, A.; PANDEY, B.; VASHISHTA, K.; KALIA, K.; PRADEEPKUMAR, B.; DAS, T. A study of today's ai through chatbots and rediscovery of machine intelligence. *International Journal of u-and e-Service, Science and Technology*, v. 8, n. 7, p. 277–284, 2015. Citado na página 16.
- MEYER, B. *Abstraction & Orchestration - Integrating the WhatsApp Business API*. 2018. Acesso em: 17 dez. 2023. Disponível em: <<https://www.twilio.com/blog/building-the-twilio-api-for-whatsapp>>. Citado na página 23.
- MOBILETIME; OPINIONBOX. *Mensageria no Brasil - Agosto de 2023 - Panorama Mobile Time/Opinion Box*. 2023. Acesso em: 15 dez. 2023. Disponível em: <<https://www.mobiletime.com.br/pesquisas/mensageria-no-brasil-agosto-de-2023/>>. Citado 3 vezes nas páginas 23, 24 e 47.
- MUHAMMAD, A. F.; SUSANTO, D.; ALIMUDIN, A.; ADILA, F.; ASSIDIQI, M. H.; NABHAN, S. Developing english conversation chatbot using dialogflow. In: IEEE. *2020 International Electronics Symposium (IES)*. [S.l.], 2020. p. 468–475. Citado na página 14.
- NIMAVAT, K.; CHAMPANERIA, T. Chatbots: An overview types, architecture, tools and future possibilities. *Int. J. Sci. Res. Dev*, v. 5, n. 7, p. 1019–1024, 2017. Citado na página 16.
- NORONHA, O. *Rasa 2.0: Criando um chatbot com uma poderosa ferramenta Open Source | by Odilio Noronha | RapaduraTech | Medium*. 2020. Acesso em: 17 dez. 2023. Disponível em: <<https://medium.com/rapaduratech/rasa-2-0-criando-um-chatbot-com-uma-poderosa-ferramenta-open-source-289430afcfb5>>. Citado na página 22.
- OPENAI. *ChatGPT*. 2023. Acesso em: 9 dez. 2023. Disponível em: <<https://openai.com/chatgpt>>. Citado 3 vezes nas páginas 12, 15 e 16.
- RANAVARE, S. S.; KAMATH, R. Artificial intelligence based chatbot for placement activity at college using dialogflow. *Our Heritage*, v. 68, n. 30, p. 4806–4814, 2020. Citado na página 13.
- RASA. *Conversational AI Platform | Superior Customer Experiences Start Here*. 2023. Acesso em: 2 dez. 2023. Disponível em: <<https://rasa.com/>>. Citado na página 25.
- REYES, R.; GARZA, D.; GARRIDO, L.; CUEVA, V. De la; RAMIREZ, J. Methodology for the implementation of virtual assistants for education using google dialogflow. In: SPRINGER. *Advances in Soft Computing: 18th Mexican International Conference on Artificial Intelligence, MICAI 2019, Xalapa, Mexico, October 27–November 2, 2019, Proceedings 18*. [S.l.], 2019. p. 440–451. Citado na página 13.

- SABHARWAL, N.; BARUA, S.; ANAND, N.; AGGARWAL, P.; SABHARWAL, N.; BARUA, S.; ANAND, N.; AGGARWAL, P. Introduction to cognitive virtual bots. *Developing Cognitive Bots Using the IBM Watson Engine: Practical, Hands-on Guide to Developing Complex Cognitive Bots Using the IBM Watson Platform*, Springer, p. 1–13, 2020. Citado na página 16.
- TELEGRAM. *Bots: An introduction for developers*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://core.telegram.org/bots>>. Citado na página 24.
- TELEGRAM. *Perguntas Frequentes*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://telegram.org/faq#p-o-que-e-telegram-o-que-faco-aqui>>. Citado na página 24.
- TELEGRAM. *Telegram APIs*. 2023. Acesso em: 5 dez. 2023. Disponível em: <<https://core.telegram.org/api#bot-api>>. Citado na página 24.
- TWILIO. *Twilio – APIs de comunicação para SMS, voz, vídeo e autenticação | Twilio*. 2023. Acesso em: 2 dez. 2023. Disponível em: <<https://www.twilio.com/pt-br>>. Citado na página 25.
- VOICE, A. N. *Cortana deixa de ter integração com a Alexa - NewVoice*. 2021. Acesso em: 10 dez. 2023. Disponível em: <<https://newvoice.ai/2021/11/17/cortana-deixa-de-ter-integracao-com-a-alexa/>>. Citado na página 17.
- WHATSAPP. *WhatsApp | Mensagens e ligações privadas, gratuitas, seguras e confiáveis*. 2023. Acesso em: 17 dez. 2023. Disponível em: <https://www.whatsapp.com/?lang=pt_BR>. Citado na página 23.