

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN

FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT

DEPARTAMENTO DE INFORMÁTICA – DI

Emídio Lopes de Souza Neto

**Desenvolvimento de um chatbot para auxílio no atendimento ao discente da
PRAE**

MOSSORÓ - RN

2022

Emídio Lopes de Souza Neto

**Desenvolvimento de um chatbot para auxílio no atendimento ao discente da
PRAE**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Isaac Lima Oliveira Filho.

MOSSORÓ - RN

2022

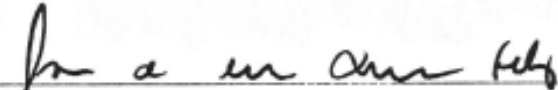
Emídio Lopes de Souza Neto

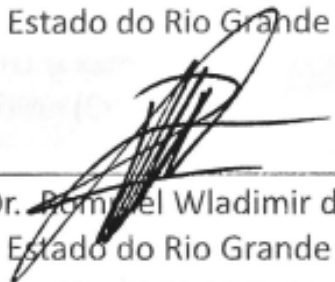
Desenvolvimento de um chatbot para auxílio no atendimento ao discente da PRAE

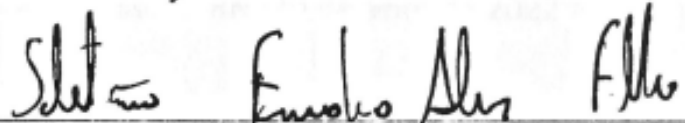
Monografia apresentada como pré-requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovado em: 20 / 04 / 2022

Banca Examinadora


Prof. Dr. Isaac de Lima Oliveira Filho
Universidade do Estado do Rio Grande do Norte - UERN


Prof. Dr. Rommel Wladimir de Lima
Universidade do Estado do Rio Grande do Norte - UERN


Prof. Dr. Sebastião Emídio Alves Filho
Universidade do Estado do Rio Grande do Norte - UERN

A minha família que sempre me apoiou.

AGRADECIMENTOS

Agradeço a toda a minha família que mesmo com todas as dificuldades enfrentadas durante essa jornada, sempre me apoiou e se fez presente. A minha tia Simone que me ajudou nesse trabalho e me motivou todos os dias.

Um agradecimento especial ao professor Isaac, um grande amigo que fez para o restante da vida, que me proporcionou várias oportunidades de aprendizado e crescimento como pessoal e profissional.

“Try not. Do. Or do not. There is no try.”

(Master Yoda)

RESUMO

Um *chatbot* é um programa capaz de simular o pensamento e fala humano através de técnicas e algoritmos de inteligência artificial, interagindo de forma completamente automatizada por mensagens de texto e até mesmo voz. É apresentado neste trabalho o desenvolvimento da primeira versão de um chatbot para auxílio no atendimento ao aluno da Universidade do Estado do Rio Grande do Norte. O protótipo atende a pró-reitoria de assuntos estudantis - PRAE, sendo capaz de responder a dúvidas referentes a um dos mais recentes editais disponibilizados pelo órgão, o auxílio creche. As respostas são dadas de forma automatizada e sem a necessidade de nenhuma interação humana, reduzindo a carga de trabalho necessária para a resposta de diversos e-mails repetitivos. O roteiro contendo os dados utilizados para o desenvolvimento do *chatbot* foi disponibilizado pela própria pró-reitoria, após um levantamento das maiores dificuldades dos discentes durante a realização da inscrição em seus editais. A plataforma utilizada para desenvolvimento da ferramenta foi o *Dialogflow*, do *Google*, e utiliza serviços em nuvem do *Google Cloud Platform* para se conectar ao usuário final, possibilitando uma alta disponibilidade e escalabilidade.

Palavras-chave: Chatbot, Processamento de linguagem natural, Inteligência artificial.

ABSTRACT

A chatbot is a program capable of simulating human thought and speech through artificial intelligence techniques and algorithms, interacting in a completely automated way through text messages and even voice. It is presented in this work the development of the first version of a chatbot to assist in the service to the students of the University of the State of Rio Grande do Norte. The prototype serves the Dean of Student Affairs - PRAE, being able to answer questions regarding one of the most recent services made available by the agency, the daycare aid. Responses are provided automatically and without the need for any human interaction, reducing the workload required to respond to multiple repetitive emails. The script containing the data used for the development of the chatbot was made available by the pro-rectory itself, after a survey of the greatest difficulties of the students during the registration in their public notices. The platform used to develop the tool was Dialogflow, from Google, and it uses cloud services from the Google Cloud Platform to connect to the end user, enabling high availability and scalability.

Keywords: Chatbot, Natural language processing, Artificial intelligence.

LISTA DE SIGLAS

PLN	Processamento de linguagem natural
IA	Inteligência Artificial
MIT	Massachusetts Institute of Technology
IBM	International Business Machines Corporation
API	Interface de Programação de Aplicações
ML	Machine Learning
HTTPS	Hyper Text Transfer Protocol Secure
JSON	JavaScript Object Notation

LISTA DE FIGURAS

Figura 1	Sub-áreas da inteligência artificial	21
Figura 2	Fluxo de correspondência de uma intenção	27
Figura 3	Fluxo de conversa utilizando contextos	29
Figura 4	Exemplo de utilização do fulfillment	30
Figura 5	Quadro de desenvolvimento - A fazer	35
Figura 6	Intenção de boas-vindas	37
Figura 7	Entidades do sistema	38
Figura 8	Intenções sobre o edital	39
Figura 9	Intenção de feedback negativo	40
Figura 10	Fallback intent - Text response	41
Figura 11	Telegram	42
Figura 12	Código do webhook	43
Figura 13	Gráfico de utilização do chatbot	44
Figura 14	Gráfico de utilização das integrações	45
Figura 15	Gráfico de requisições por intenção	46
Figura 16	Dúvidas respondidas	46
Figura 17	Satisfação dos usuários	47

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivos	14
1.1.1 Objetivos específicos	14
1.2 Trabalhos relacionados	14
1.2.1 Desenvolvimento de um chatterbot para a página web de um curso de nível superior	15
1.2.2 Helena: Um chatbot para auxílio dos discentes do decom em trâmites universitários	16
2 REFERENCIAL TEÓRICO	17
2.1 Inteligência Artificial	17
2.2 Machine Learning	19
2.3 Deep Learning	20
2.4 Processamento de linguagem natural	21
2.5 Chatbot	23
3 TECNOLOGIAS UTILIZADAS	25
3.1 Dialogflow	25
3.2 Integração com a API de armazenamento	29
3.2.1 Python	31
3.2.2 Flask	31
3.2.3 MongoDB	32
3.2.4 Heroku	33
4 DESENVOLVIMENTO	34
4.1 Scrum	34
4.2 Criação do roteiro	35
4.3 Implementação no Dialogflow	36
4.3.1 Interface humano-computador	41

4.4 Criação do webhook	42
4.5 Conexão com o banco de dados.....	43
5 RESULTADOS	44
6 CONSIDERAÇÕES FINAIS	48
6.1 Trabalhos futuros	48
REFERÊNCIAS	50
APÊNDICE A – MATERIAL COMPLEMENTAR	52

1 INTRODUÇÃO

Os diversos avanços tecnológicos dos últimos séculos fomentaram um gigantesco anseio da população por conhecimento e informação que apenas aumenta com o advento do tempo. A facilidade de acesso a conteúdos, serviços e pessoas modificou radicalmente a forma como empresas e clientes pensam, tornando necessário cada vez mais a disponibilidade rápida e fácil de informações e atendimento de forma rápida e eficiente.

Atender a um cliente é um relacionamento interpessoal, o qual se processa através da comunicação impulsionada pela motivação, visando nessa interação, detectar as necessidades ocultas do cliente e satisfazê-la com qualidade (SANTOS, 2011). Dessa forma, disponibilizar a informação necessária não é o suficiente, havendo então a necessidade de se implementar um atendimento inteligente e prestativo, para que a satisfação do cliente seja garantida durante todo o processo de atendimento. Para SANTOS (2011), “o atendimento é considerado fator determinante para a conquista e a manutenção de clientes, transformando-se numa atitude fundamental para a sobrevivência da própria empresa”.

Segundo UERN (2021), a universidade atende a mais de oito mil estudantes ativos, de seus vários núcleos espalhados pelo Rio Grande do Norte. A pró-reitoria de assuntos estudantis (PRAE), é o órgão responsável por fomentar e desenvolver mecanismos que favoreçam a permanência, o aprimoramento profissional, o intercâmbio com outras instituições e a melhoria na qualidade de vida dos discentes da UERN. Com o objetivo de aprimorar o atendimento ao discente promovido pela instituição, este trabalho propôs o desenvolvimento de um *chatbot* que será responsável por responder os questionamentos mais comuns em relação aos editais disponibilizados pela PRAE, mais especificamente o edital referente ao Auxílio Creche. A ferramenta será de grande importância para o órgão, pois além de facilitar o atendimento ao cliente e automatizar conversas, irá reduzir a carga de trabalho, retirando a demanda necessária de um tempo exclusivo para a resolução de dúvidas e questionamentos através de e-mails e afins. Além disso, é também uma forma mais eficiente de se classificar e analisar quais as maiores dificuldades apresentadas pelos discentes na hora da realização do cadastro em editais e afins,

pois o sistema não só será capaz de tirar dúvidas como também armazenar questionamentos ao qual não pode responder. Considera-se por meio da análise realizada que o sistema será de extremo benefício, aperfeiçoando o atendimento ao discente da universidade e o tornando cada vez mais abrangente e eficiente ao longo do tempo.

1.1 Objetivos

Esse trabalho tem como objetivo principal a implementação e análise da utilização de um Chatbot de atendimento ao aluno na pró-reitoria de assuntos estudantis (PRAE), através de uma das inúmeras ferramentas de desenvolvimento de chatbots existentes.

1.1.1 Objetivos específicos

- Estudar as ferramentas de *chatbot* existentes;
- Testar as funcionalidades das ferramentas verificadas;
- Apresentar os resultados adquiridos;
- Desenvolver material de treinamento.

1.2 Trabalhos Relacionados

A utilização de uma tecnologia de inteligência artificial para resolução de problemas não é nenhuma novidade nos dias de hoje, sendo utilizada mundialmente nas mais diversas áreas. Serão listados a seguir alguns trabalhos que seguiram o mesmo rumo deste, fazendo o uso de um chatbot para atender a alguma demanda do mundo real.

Foi feita uma pesquisa através da ferramenta Google Acadêmico, utilizando-se a tag *chatbot* com fontes em artigos e outros trabalhos acadêmicos. Pelo viés acadêmico apresentado nesta pesquisa, buscou-se a apresentação de trabalhos que também foram desenvolvidos para atender a alguma demanda acadêmica da universidade ou instituição na qual foram desenvolvidos e utilizados.

1.2.1 Desenvolvimento de um chatterbot para a página web de um curso de nível superior

Desenvolvido por Brito (2017), o *chatbot* tem como objetivo atender as demandas dos discentes do curso de Sistemas de Informação da Universidade Federal do Ceará, através da criação de um FAQ com perguntas e respostas de dúvidas recorrentes sobre o curso de ensino superior disponibilizadas pela inteligência artificial. A plataforma utilizada para sua criação foi o *IBM Watson Conversation*, com o *Facebook Messenger* servindo como interface humano-computador.

Após a primeira validação do *chatterbot* pode-se concluir que ele conseguiu responder a maioria das perguntas em um nível satisfatório. No entanto tiveram algumas perguntas em que o *chatterbot* não conseguiu responder por não conseguir entender a intenção na pergunta, coisa que um operador humano entenderia [BRITO, 2017].

O resultado do trabalho foi em termos gerais positivo, com trabalhos futuros como expansão da sua área de conhecimento e novos testes de validação após maior tempo de uso sendo apresentados pelo criador. Considerou-se um trabalho satisfatório pois foi capaz de atender a maioria das necessidades apresentadas pelos usuários, além de cumprir com sucesso as funções na qual foi preparado através do treinamento com a base de dados oferecida.

No contexto de aprendizagem de máquina, quanto maior o número de interações que o *chatterbot* tiver, melhor será seu desempenho em responder perguntas de seu domínio. Portanto, fica como trabalho futuro fazer uma nova validação com o *chatterbot* depois que ele esteja funcionando há algum tempo para ver seu desempenho. Além disso, expandir as áreas de conhecimento do *chatterbot* para dar suporte a outros assuntos da UFC, como infraestrutura [BRITO, 2017].

1.2.2 Helena: Um chatbot para auxílio dos discentes do decom em trâmites universitários

Idealizado e desenvolvido por Monteiro (2021), surge para que alunos de graduação da Universidade Federal de Ouro Preto possam sanar dúvidas sobre os cursos de graduação durante o período de pandemia do Covid-19, fato que dificultou as atividades das universidades do mundo inteiro.

No trabalho, foram coletados dados do Manual do Aluno, da página do Colegiado de Ciência da Computação (COCIC) e de exemplos cedidos pela própria instituição. O treinamento foi realizado pela plataforma *IBM Watson Conversation*, com a interface entre o usuário e o chatbot sendo desenvolvida no formato de um site. A base de treino utilizada no trabalho foi hospedada na nuvem e armazenada no formato de planilha do Google.

Pelos resultados obtidos no presente trabalho, podemos dizer que o chatbot Helena obteve resultados satisfatórios, tanto no quesito de experiência no uso da plataforma quanto na acurácia durante os testes realizados [MONTEIRO, 2021].

Para trabalhos futuros, foram apresentadas melhorias como a criação de um sistema independente de processamento de linguagem natural, para um maior controle do que acontece por trás da arquitetura definida do projeto e maior liberdade no desenvolvimento de melhorias, tornando-se independente da plataforma Watson, e expansão na base de treino, tornando assim os diálogos mais fluídos e possibilitando a resolução de mais problemas do que o que já se era resolvido.

Com os tópicos elencados implementados, pretende-se oferecer uma experiência ainda melhor ao conversar com Helena. Desta forma, é possível que ela se torne mais uma ponte de comunicação entre os estudantes de graduação e a Universidade, ajudando estudantes a resolverem problemas de seu cotidiano estudantil [MONTEIRO, 2021].

2 REFERENCIAL TEÓRICO

O objetivo deste capítulo é documentar e descrever todos os conceitos que serviram como base para o desenvolvimento do trabalho, apresentando descrições e informações pertinentes que são trabalhadas e utilizadas durante todo o seu desdobramento.

2.1 Inteligência artificial

Inteligência artificial (IA) faz referência a aplicações que simulam o comportamento e a inteligência humanos para realizar tarefas e são capazes de aprender a cada iteração a partir dos dados coletados [ORACLE, 2022]. A inteligência artificial já é usada em vários domínios da computação hoje e pode se manifestar de várias maneiras:

- Chatbots que usam inteligência artificial para entender melhor as necessidades dos clientes e automatizar os processos de atendimento.
- Assistentes inteligentes que usam inteligência artificial para realizar diversas tarefas exigidas pelos usuários.
- Um sistema de recomendação que fornece recomendações automáticas de séries e filmes com base no histórico de uso do usuário.

Tem mais a ver com capacidade de processamento e aprendizado do que qualquer função ou formato específico. O termo inteligência artificial tornou-se sinônimo de aplicativos que executam funções mais complexas que antes exigiam interação direta do usuário. Não é mais uma questão de se aplicar, mas de como aplicar.

A inteligência artificial está transformando nossa relação com a tecnologia, e é base da revolução digital em curso a partir da confluência de tecnologias do mundo digital (internet das coisas/IoT, *blockchain*, plataformas digitais), do mundo físico (veículos autônomos, impressão 3D, robótica avançada, novos materiais) e do mundo biológico (manipulação genética) [KAUFMAN, 2019, p. 10].

A IA como projeto efetivo só se tornou possível após o aparecimento dos computadores modernos, ou seja, após a Segunda Guerra Mundial. Até então, havia

dificuldades técnicas que precisavam ser superadas para que o projeto dessas máquinas mais modernas pudesse sair do papel [TEIXEIRA, 2019, p. 4].

Em 1950, o matemático Alan Turing desenvolveu o chamado Teste de Turing. A ideia do experimento é verificar se máquinas podem enviar mensagens como humanos sem criar desconfiança, fazendo com que o destinatário acredite que este não é um programa de computador. Para o autor, se pelo menos um terço dos participantes estava convencido de que a conversa fosse com um humano, a máquina poderia ser dita como inteligente.

Desde então, diversos avanços tecnológicos permitiram um avanço extremamente rápido nas formas e técnicas de desenvolvimento da IA. Coisas como máquinas inteligentes que conversam casualmente com seres humanos deixou de ser apenas uma teoria de ficção científica para se tornar realidade, como é o exemplo do robô Sophia, desenvolvida pela empresa *Hanson Robotics*, de Hong Kong.

Foi ativado pela primeira vez no dia 14 de fevereiro de 2016 e desde então já passou por diversas atualizações e melhorias para aprimorar ainda mais a forma com que conversa e interage com humanos. O robô é projetado para interagir, se comportar e imitar humanos, principais características que o diferem das demais inteligências artificiais desenvolvidas até o momento. Sophia pode realizar processamento de dados visuais e reconhecimento facial, além de aprender constantemente através de técnicas de processamento de linguagem natural e aprendizado de máquina. Seu software de inteligência artificial foi desenvolvido pela *SingularityNET* e é responsável por analisar conversas e abstrair os dados que considera significativos para melhorar suas interações futuras.

A maioria dos avanços observados na última década provém do *Deep Learning* (aprendizado profundo), inspirado no funcionamento do cérebro, por isso também conhecido como redes neurais. Estamos nos primórdios da IA, ainda um campo aberto, mas para quem quiser se manter relevante, econômica e socialmente, vale conhecer minimamente como funciona a IA [KAUFMAN, 2019].

2.2 Machine Learning

O termo surgiu na época de 1959, criado por um engenheiro da computação chamado Arthur Samuel. Seu objetivo era desenvolver máquinas autônomas, dando a capacidade dos computadores aprenderem sem serem programados para isso. Segundo o próprio, é “um campo de estudo que dá aos computadores a habilidade de aprender sem terem sido programados para tal”.

Hoje, o aprendizado de máquina é a base de uma variedade de aplicativos que usamos todos os dias, desde recomendações de produtos a reconhecimento de voz, bem como alguns que ainda não usamos todos os dias, incluindo carros sem motorista. É a base da nova abordagem da computação onde não escrevemos programas, mas coletamos dados; a ideia é que os algoritmos aprendam as tarefas automaticamente a partir dos dados coletados [ETHEM, 2016, p. 29].

O aprendizado de máquina utiliza diferentes técnicas de treinamento, utilizadas de acordo com o caso de uso, algumas delas sendo:

- **Aprendizado supervisionado:** Método de treinamento que utiliza tanto dados de saída como de entrada para treinamento. A partir dos padrões identificados, busca-se prever resultados futuros em novos dados de entrada.
- **Aprendizado não supervisionado:** Neste método, não há identificação nenhuma entre dados de entrada e saída, cabendo a própria inteligência artificial averiguar padrões entre os dados inseridos e tirar suas próprias conclusões.
- **Aprendizado por reforço:** Aqui aplica-se a lógica de “tentativa e erro”. A ideia é que a inteligência aprenda com os próprios erros de classificação durante o processo de aprendizagem, supervisionado pelo programador que averigua e informa se os dados apreendidos foram classificados corretamente ou não. Ou seja, ela aprende com base na experiência adquirida em cada teste.

Atualmente, com advento das grandes quantidades de dados armazenadas em *Big Data*, tornou-se ainda mais fácil e prático o treinamento e desenvolvimento de uma inteligência artificial através do processo de aprendizado de máquina, uma

vez que a imensa quantidade de dados gerados diariamente na internet é capaz de alimentar informações quase que infinitas para algoritmos de treinamento.

Normalmente, técnicas de machine learning são mais comumente utilizadas no treinamento de dados estruturados, como tabelas e bancos de dados. Para dados não estruturados, como imagens, opta-se pela utilização de algoritmos de deep learning, sub-área mais especializada do aprendizado de máquina que simula o funcionamento do cérebro humano.

2.3 Deep Learning

O aprendizado profundo é o subcampo da inteligência artificial que se concentra na criação de grandes modelos de redes neurais capazes de tomar decisões precisas baseadas em dados [KELLEHER, 2019].

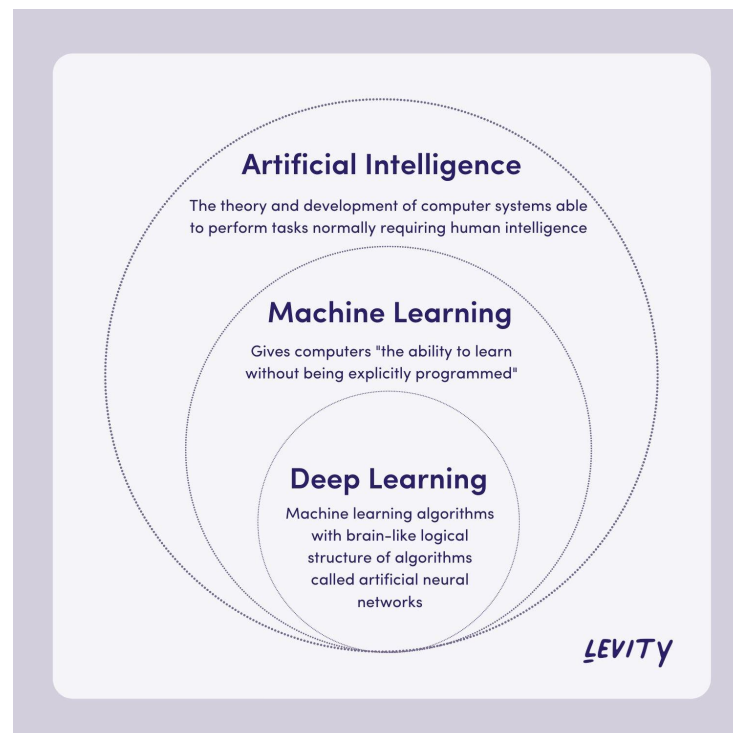
LeCun, Bengio e Hinton (2015) afirmam que:

O aprendizado profundo descobre uma estrutura complexa em grandes conjuntos de dados usando o algoritmo de retropropagação para indicar como uma máquina deve alterar seus parâmetros internos que são usados para calcular a representação em cada camada a partir da representação na camada anterior. Redes convolucionais profundas trouxeram avanços no processamento de imagens, vídeo, fala e áudio, enquanto as redes recorrentes iluminaram dados sequenciais, como texto e fala.

Ele permite que modelos computacionais aprendam representações de dados com vários níveis de abstração. O aprendizado profundo usa estruturas algorítmicas complexas modeladas no cérebro humano. Isso permite o processamento de dados não estruturados, como documentos, imagens e texto.

Talvez o exemplo mais conhecido de aprendizado profundo seja o AlphaGo. Go é um jogo de tabuleiro semelhante ao xadrez. AlphaGo foi o primeiro programa de computador a vencer um jogador profissional de Go. Em março de 2016, venceu o principal profissional coreano, Lee Sedol, em uma partida assistida por mais de duzentos milhões de pessoas [KELLEHER, 2019, p. 2].

Figura 1 - Sub-áreas da inteligência artificial



Fonte: Levity (2022)

A Figura 1 apresenta de forma breve algumas das diversas subáreas que surgem a partir da inteligência artificial.

Os algoritmos de aprendizado profundo podem ser considerados uma evolução sofisticada e matematicamente complexa dos algoritmos de aprendizado de máquina. O campo tem recebido muita atenção ultimamente e por um bom motivo: desenvolvimentos recentes levaram a resultados que antes não se pensava serem possíveis [LEVITY, 2022].

2.4 Processamento de linguagem natural

Processamento de Linguagem Natural (PLN) é um ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana. O PLN tem suas raízes em várias disciplinas, incluindo ciência da computação e linguística computacional, que visam preencher a lacuna entre a comunicação humana e a compreensão dos computadores. O termo "PLN" às vezes

é usado de forma mais restrita, muitas vezes excluindo a recuperação de informações e às vezes até excluindo a tradução automática [SINGH, 2021].

O PLN não é uma tarefa simples devido a alta ambiguidade da linguagem natural, o que torna difícil o processamento da mensagem e faz com que cada palavra ou frase tenha significados diferentes de acordo com o contexto em qual são inseridas. Isso o torna algo totalmente diferente da linguagem de programação, que trabalha e é desenvolvida tentando evitar, justamente, a ambiguidade.

O primeiro algoritmo a se destacar na história foi o *Georgetown-IBM experiment*, desenvolvido pela Universidade de Georgetown em colaboração com a IBM em 1954, traduzindo com sucesso mais de 50 frases do russo para o inglês. Embora o texto fosse inserido na forma de cartões, isso foi um grande avanço para os humanos, potencialmente indicando que tradutores automáticos poderiam ser feitos.

Contudo, grandes avanços no processamento da linguagem natural só iriam ocorrer na década de 1980, graças aos aperfeiçoamentos em hardware e software e ao uso de algoritmos de aprendizado de máquina. As regras não precisavam mais ser escritas pelo programador, sendo aprendidas através de banco de dados. Esse avanço afeta o desempenho porque, além de remover o viés humano da inserção de regras manuais, permite processar grandes quantidades de dados em um curto espaço de tempo.

O processamento de linguagem natural tem uma forte relação com o desenvolvimento de *chatbots* já que é ele a principal ferramenta utilizada para fazer com que máquinas entendam as nuances da linguagem humana, tornando sua relação com os usuários mais próxima. É também uma forma mais segura de responder às necessidades dos clientes, atraindo assim mais atenção das empresas para o desenvolvimento de soluções utilizando esta estratégia.

Mesmo com todos os avanços tecnológicos, a utilização de PLN ainda é um desafio, dada todas as dificuldades na compreensão da comunicação humana, além de existir a necessidade da máquina de compreender assuntos como erros de digitação, abreviações e erros de pontuação durante a escrita do usuário. Sendo assim, ela não passa de mais uma das formas de desenvolvimento de sistemas

baseados na linguagem natural, a fim de determinar quais trechos são realmente importantes para que o significado seja compreendido. É uma estratégia que exige o esforço de diversos profissionais capacitados, como cientistas da computação e linguistas.

2.5 Chatbot

Um *chatterbot*, ou como é mais conhecido, *chatbot*, é literalmente o que o seu nome indica: um robô que conversa. O termo surge da junção das palavras em inglês *chatter*, que pode ser traduzida como conversar, e *bot*, que significa robô. Segundo RAJ (2019) “o problema básico que esses *bots* tentam resolver é tornar-se um intermediário e ajudar os usuários a serem mais produtivos”. Eles fazem isso através da automação de processos mais simples e de fácil solução, como em sistemas de FAQ e cadastro. A tendência é de que fiquem cada vez mais inteligentes à medida que manipulam a entrada de dados dos usuários e extraem as informações mais úteis dela.

MOWBRAY (2012) diz que os *bots* são “pessoas que atuam de acordo com um programa de software em vez de serem controladas por um usuário humano”. Sendo assim, são programas criados para realizar uma interação humana com transparência e direcionadas a contexto, podendo argumentar, responder perguntas, dar informações ou até mesmo fazer sugestões.

Na década de 1960, o cientista da computação e pesquisador do *Massachusetts Institute of Technology* (MIT) Joseph Weizenbaum desenvolveu o primeiro chatbot para simular a interação humana: o software Eliza. O princípio por trás do chatbot era simples: imitar um psicoterapeuta, interagir e fazer perguntas com base nos termos que os usuários inserem durante a conversa. Infelizmente, apesar de ser uma inovação para a época, devido ao seu código bastante simples, Eliza não passou no teste de Turing. Ela era incapaz de, por exemplo, interpretar a intenção do usuário ou prolongar a conversa por muito tempo.

Um grande avanço na área e nos chatbots em geral veio em 2000, com a criação do *chatbot* Watson, pela *International Business Machines Corporation* (IBM). O intuito de sua criação era competir no programa Jeopardy, programa de televisão

que consiste em um show de perguntas e respostas. Foi só em 2011 que ele finalmente participou de uma edição e ganhou de dois especialistas respondendo perguntas de conhecimento geral. Desde então, o chatbot já passou por diversas alterações e incrementações em seu código, passando a utilizar processamento de linguagem natural (PLN) e aprendizado de máquina para processar uma ainda maior quantidade de dados e gerar respostas cada vez mais adequadas. Hoje, ele é oferecido pela IBM como uma API para desenvolvimento de bots, sendo um dos primeiros do mercado a fazer isso.

Outros exemplos de aplicações semelhantes são o *Dialogflow*, antigamente conhecido como *API.AI*, plataforma de construção de interfaces de conversação para aplicativos e dispositivos disponibilizada pelo *Google*, e o *Botsify*, plataforma de chatbot gerenciada que fornece automação de chat unificada para empresas.

3 TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas as tecnologias utilizadas para o desenvolvimento deste trabalho. Serão apresentadas as principais características, funcionalidades e descrições referentes às plataformas e *frameworks* utilizados, de acordo com as fontes de pesquisa.

3.1 Dialogflow

Uma das peças mais importantes no desenvolvimento de um chatbot é o seu treinamento, etapa responsável por fazer com que a linguagem humana seja interpretada e entendida pela máquina que, a partir da informação que lhe foi passada, deverá realizar algum tipo de ação. Em busca de um melhor nível de precisão e facilidade nesse quesito, a plataforma utilizada para realizar todo o processo de criação e treinamento do *chatbot* foi o *Dialogflow*.

O *Dialogflow* é um *framework* que utiliza Processamento de Linguagem Natural (PLN) e *Machine Learning* (ML) para extrair as intenções e as entidades de um diálogo para então elaborar a resposta mais precisa [GOOGLE, 2018]. Utilizar o *Dialogflow* traz a vantagem de que, por ser um serviço Google, é executado no *Google Cloud Platform*, permitindo o alcance de muito mais usuários, principalmente devido a possibilidade de integração ao *Google Assistant* [GOOGLE, 2018].

Atualmente, são disponibilizados dois tipos de serviços diferentes, cada um com seu agente, API, biblioteca e documentação exclusivos:

- **Dialogflow CX:** Mais indicado para projetos de larga escala que precisam de agentes mais complexos.
- **Dialogflow ES:** Modelo que foi utilizado no projeto, ele fornece o tipo de agente padrão, mais indicado para projetos mais simples e pequenos.

Uma das principais vantagens em se utilizar o *Dialogflow* é a sua estratégia de pagamento. A plataforma é de utilização gratuita e libera para seus usuários uma quota de utilização que dificilmente será ultrapassada pela maioria das aplicações,

suportando até 180 requisições por minuto. Contudo, caso isto venha a ocorrer, o *Dialogflow* é cobrado mensalmente com base na edição e nas requisições feitas durante o mês.

A plataforma consiste de cinco conceitos chaves: Agente, Intenções, Entidades, Contexto e Intenções de acompanhamento.

Agente: É um agente virtual e a peça fundamental do chatbot. Ele é responsável por conversar com diversos usuários finais e traduzir as nuances da linguagem humana para dados estruturados que possam ser entendidos por um aplicativo ou sistema.

Um agente do Dialogflow é semelhante a um agente de call center humano. Você treina os dois para lidar com os cenários de conversa esperados e seu treinamento não precisa ser excessivamente explícito [GOOGLE, 2018].

Intenções: Intenções são responsáveis por categorizar a intenção de um usuário final para cada turno da conversa. Cada agente é composto de várias intenções a fim de gerar uma conversa completa. Quando um usuário escreve alguma coisa, o *Dialogflow* tenta corresponder com a intenção que determina ser a certa. A correspondência de uma intenção também pode ser chamada de classificação de intenção.

Uma intenção contém normalmente as seguintes características:

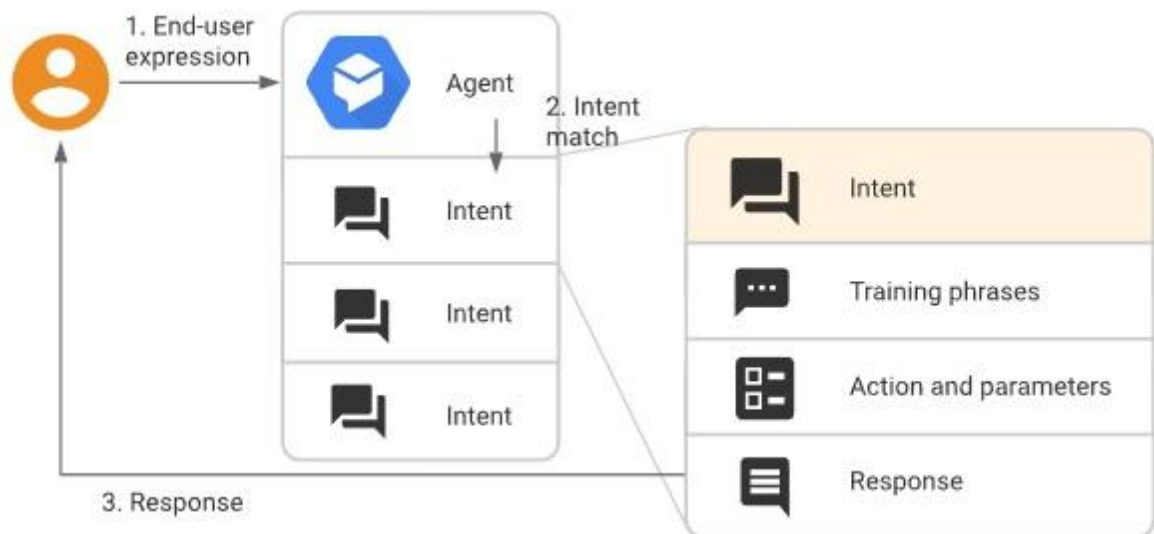
- **Frases de treinamento:** Servem como exemplos de possíveis frases que o usuário final pode dizer referindo-se a intenção. Quando uma frase semelhante é dita pelo usuário, o *Dialogflow* corresponde-a como achar adequado. Não é necessário inserir uma lista muito longa de frases, pois a inteligência artificial da plataforma expande a lista com frases semelhantes.
- **Ações:** Você pode definir certas ações para cada intenção. Quando determinada intenção for acionada, o *Dialogflow* fornece a ação ao sistema no qual esteja integrado e ele pode realizar algum tipo de atividade definida pelo programador.
- **Parâmetros:** O conceito de parâmetros é algo que se assemelha bastante a variáveis em uma linguagem de programação. Quando uma intenção é correspondida em tempo de execução, o *Dialogflow* pode extrair

determinadas informações da conversa ao armazená-las em parâmetros. Cada parâmetro tem um tipo, chamado de tipo de entidade, que determina como os dados serão armazenados.

- **Respostas:** Vários tipos de respostas podem ser inseridas na plataforma. Elas podem vir em forma de texto, áudio, imagem ou até mesmo algum tipo de ação programada no sistema.

A Figura 2 exemplifica melhor o fluxo básico para correspondência de intenção e resposta para o usuário final. Um agente é composto por diversas intenções, que são compostas de *Training phrases*, onde são inseridas as frases de treinamento para a sua correspondência, *Action and parameters*, que podem ou não ser acionados durante o reconhecimento de uma intenção e *Response*, que é como o agente irá responder caso ela seja acionada.

Figura 2 - Fluxo de correspondência de uma intenção



Fonte: Dialogflow (2018)

Entidades: Cada parâmetro de uma intenção tem um tipo, chamado de tipo de entidade, o que determina como cada dado inserido pelo usuário final é extraído e armazenado. O próprio *Dialogflow* providência tipos pré definidos chamados entidades do sistema. Eles podem ser utilizados para diferentes tipos de informação, como por exemplo cores, endereços de email, nomes, etc. Também é possível criar

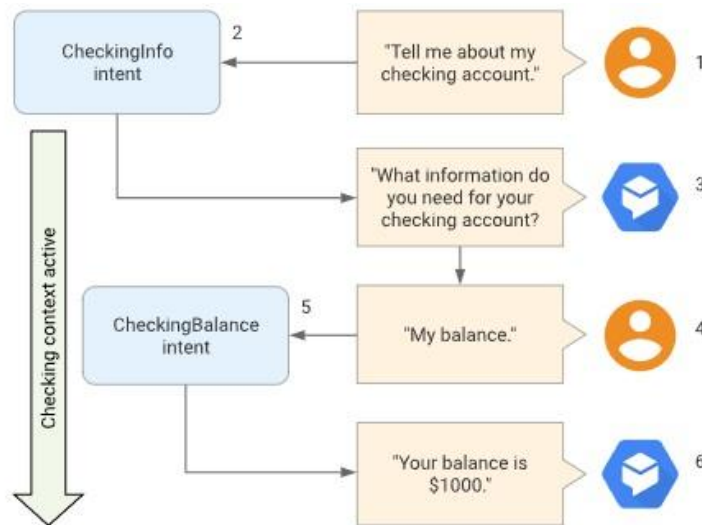
suas próprias entidades para armazenar dados customizados de acordo com a situação.

Contexto: Contextos no *Dialogflow* são semelhantes aos contextos da linguagem natural. Se uma pessoa fala “elas são redondas”, você primeiro precisará entender o contexto da conversa para só então entender ao que a pessoa está se referindo. Similarmente, para o *Dialogflow* entender corretamente a conversa, é preciso que lhe seja fornecido em qual contexto está inserida.

A utilização de contextos em uma conversa é de extrema importância para que a sensação de se estar falando com outra pessoa seja captada pelo usuário final, aumentando assim a comodidade e satisfação ao se utilizar o serviço. A configuração do contexto é feita através da definição de contextos de entrada e saída, que são identificados por nomes do tipo *string*. Quando uma intenção é correspondida, qualquer contexto de saída que esteja configurado àquela intenção se torna ativo. Enquanto ele estiver ativo, o *Dialogflow* é mais provável que corresponda a intenções que estejam configuradas com contextos de entrada semelhantes ao contexto ativo no momento.

A Figura 3 demonstra a utilização de contexto em uma conversa com o usuário final. Neste caso, o agente desenvolve um diálogo com o usuário no contexto de saber informações sobre a sua conta do banco.

Figura 3 - Fluxo de conversa utilizando contextos



Fonte: Dialogflow (2018)

Intenções de acompanhamento: É possível utilizar intenções de acompanhamento para definir contextos automaticamente para pares de intenções. Uma intenção de acompanhamento é filha da intenção a qual está associada. Na criação de uma intenção de acompanhamento, um contexto de saída é criado automaticamente na intenção pai e um de entrada na de acompanhamento. Ela só é acionada quando é seguida especificamente da intenção pai na conversa anterior.

3.2 Integração com a API de armazenamento

O *Dialogflow* oferece um espaço de armazenamento de dados limitado, podendo guardar o histórico de conversas por apenas alguns meses até serem apagados completamente da plataforma. Para que não ocorra nenhum tipo de problema em relação a perda dos dados adquiridos durante os diálogos com o usuário final, optou-se pela criação de uma API para armazenamento do *feedback* dos usuários em relação ao *chatbot*. A aplicação foi desenvolvida com a ajuda do *Dialogflow Fulfillment*, opção disponibilizada pela própria plataforma para se fazer integração com outros sistemas.

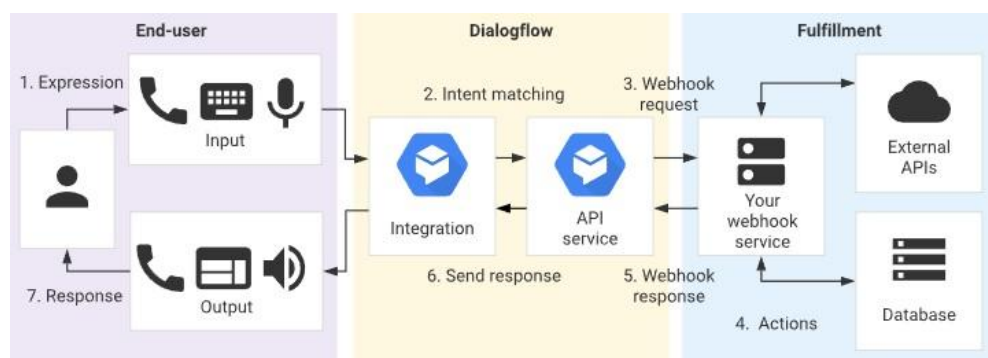
Por padrão, o *Dialogflow* responde às requisições do usuário com uma resposta estática definida na própria página de intenções. Contudo, caso haja a

necessidade, há uma forma de providenciar uma resposta mais dinâmica através do uso de *fulfillments*. Cada intenção tem uma opção para que o *fulfillment* seja ativado. Caso aquela intenção específica precise utilizar algum sistema externo à plataforma, é necessário que se ative o *fulfillment* para ela.

Se uma intenção com a opção de *fulfillment* ativada for correspondida, o *Dialogflow* envia uma requisição para o sistema de *webhook* informado pelo desenvolvedor com informações pertinentes a intenção que foi acionada. É válido lembrar que apenas requisições no formato HTTPS são aceitas, sendo necessário que a API ou sistema esteja hospedado seguindo esses requisitos para que o *webhook* seja acionado corretamente. As informações são repassadas no formato JSON e é através desse documento que o sistema poderá realizar qualquer ação necessária e informar o *Dialogflow* em como prosseguir com a conversa. Caso haja qualquer tipo de falha com a chamada ao serviço de *webhook*, a resposta estática definida na intenção será exibida no lugar.

Na Figura 4 pode-se visualizar mais graficamente o funcionamento do *fulfillment*. Pode-se notar que o usuário final se conecta por diversos meios, como voz ou texto, e é responsabilidade do *Dialogflow* realizar a conexão com os serviços externos através do sistema de *webhook*.

Figura 4 - Exemplo de utilização do fulfillment



Fonte: Dialogflow (2018)

O serviço de *webhook* para armazenamento dos dados foi desenvolvido em Python utilizando a biblioteca Flask. As informações foram armazenadas em um banco de dados MongoDB e a aplicação hospedada por intermédio do Heroku.

3.2.1 Python

Python é uma linguagem de altíssimo nível orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa (BORGES, 2014). Ela tem tomado uma grande notoriedade nos últimos anos, sendo utilizada em diversos projetos de pequeno e grande porte. A linguagem inclui diversas estruturas de alto nível como listas, dicionários, bibliotecas e diversos módulos para utilização já prontos, além de frameworks terceirizados que podem ser adicionados à livre escolha do programador que enriquecem ainda mais a linguagem de programação.

Foi criada no final de 1989 por Guido Van Rossum no Centro Holândes de Matemática e Tecnologia da Informação (CWI) como sucessora da linguagem de programação ABC, desenvolvida para lidar com exceções e interagir com o sistema operacional Amoeba.

Python é uma das melhores linguagens para inteligência artificial por causa da sua sintaxe simples, alta legibilidade e capacidade de testar rapidamente algoritmos complexos de aprendizado de máquina [COUTINHO, 2021]. Além disso, possui ótimas ferramentas de trabalho colaborativo como o *Google Colab* e o *Jupyter Notebook*. Outro aspecto positivo é que o Python é escalável: o código escrito é executado em quase todos os sistemas operacionais, como Windows, Linux, Mac OS e Unix, necessitando apenas algumas alterações mínimas no código para que seja realizada a portabilidade entre plataformas.

Ele disponibiliza várias bibliotecas específicas para utilização em projetos de inteligência artificial. *Numpy*, *Scipy*, *scikit-learn* e *matplotlib* são mundialmente utilizadas como base para diversos projetos de machine learning e afins. A disponibilidade fácil desses materiais torna o Python uma ótima opção para inteligência artificial.

3.2.2 Flask

Flask é um poderoso framework web que ajuda desenvolvedores a construir grandes projetos usando suas ferramentas favoritas. Flask usa a linguagem de

programação flexível Python e fornece um modelo simples para desenvolvimento web [THANH, 2019].

A biblioteca foi lançada em 2010 por Armin Ronacher. É o que se chama de micro-framework, que são frameworks modularizados que possuem uma estrutura bem mais simples do que frameworks mais tradicionais. A ideia inicial do Flask é que seja utilizado em projetos e aplicações com requisitos mais simples, como a criação de um site básico. Contudo, também possui um núcleo de atividade expansível, podendo-se ser adicionados novos pacotes para incrementar a funcionalidade da aplicação.

Dessa forma, o Flask torna-se uma ferramenta extremamente capaz e adaptável para diversos tipos de aplicações, de pequeno ou grande porte, apesar de ainda ser considerado um micro-framework. Ele é o principal concorrente do Django, principalmente pelo fato de ser detentor de uma arquitetura extremamente adaptável que pode ser personalizada pelo programador para atender às suas necessidades.

3.2.3 MongoDB

O MongoDB é um banco de dados orientado a documentos que possui código aberto (open source) e foi projetado para armazenar uma grande escala de dados, além de permitir que se trabalhe de forma eficiente com grandes volumes [LEANDRO, 2021]. É categorizado como um banco de dados NoSQL (not only SQL) pois o armazenamento e recuperação dos dados não é feita no formato de tabelas.

Foi desenvolvido pela empresa MongoDB Inc. e publicado sob as licenças GNU e Apache. As informações são armazenadas através de um processo mais fluido, independente e os elementos são identificados de forma exclusiva. O ambiente fornece um servidor que se pode criar e ativar vários bancos de dados diferentes. Esses projetos não estão necessariamente relacionados e sua hierarquia é completamente flexível.

É especialmente vantajoso ao se trabalhar com grandes volumes de dados, pois a sua arquitetura permite a criação de vários bancos de dados e coleções,

semelhante a tabelas em bancos de dados mais tradicionais, dentro do principal. Na coleção estão contidos os documentos que se deseja armazenar e uma única coleção pode contar vários documentos diferentes.

3.2.4 Heroku

Heroku é uma plataforma como serviço (PaaS) que permite aos desenvolvedores criar, executar e operar aplicativos inteiramente na nuvem [HEROKU, 2022]. Desenvolvido por Adam Wiggins, Orion Henry e James Lindenbaum, surgiu como uma ferramenta de suporte para projetos compatíveis com Rack, com um protótipo desenvolvido em seis meses.

É uma oferta de plataforma como serviço amplamente confiável que permite aos desenvolvedores implantar, dimensionar e gerenciar aplicativos com facilidade. Suporta várias linguagens de programação, incluindo Java, Ruby, PHP, Node.js, Python, Scala e Clojure [CLARK, 2017]. O Heroku executa aplicativos por meio de contêineres virtuais chamados Dynos.

A cobrança é feita com base no número de máquinas necessárias para a execução do projeto. Utiliza-se o *Amazon Web Services* (AWS) como infraestrutura subjacente para a execução das aplicações hospedadas. Apesar de existir a necessidade de pagamento em certos casos, o Heroku também contém um plano grátis, que apresenta funções mais limitadas mas, ao mesmo tempo, poderosas. É permitido a criação de quantas aplicações forem necessárias, mas apenas uma pode rodar 24 horas por dia. Além disso, o padrão é que ela desligue após 30 minutos sem nenhuma requisição de acesso.

4 DESENVOLVIMENTO

O objetivo deste capítulo é descrever todo o processo de desenvolvimento do chatbot, demonstrando como cada etapa do processo foi pensada e desenvolvida e também o funcionamento das ferramentas escolhidas.

4.1 Scrum

O desenvolvimento do trabalho foi realizado baseando-se nos conceitos da metodologia Scrum pois, após uma curta análise de todas as partes envolvidas, foi a que melhor se encaixou no projeto. Sua forma rápida e prática de gerenciamento de projetos, com reuniões curtas e foco na aplicação foi de extrema importância para o sucesso do chatbot.

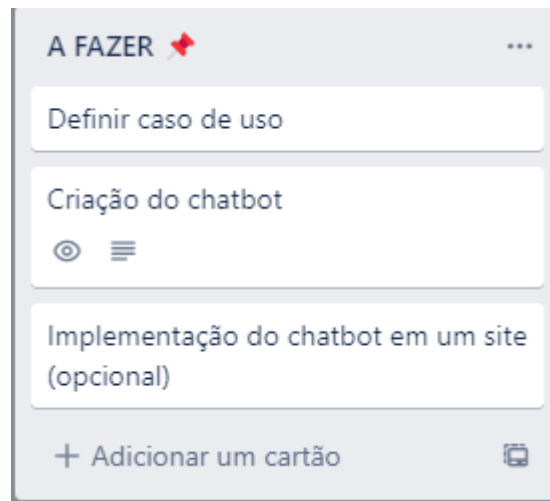
O Scrum é uma estrutura que ajuda as equipes a trabalharem juntas. Semelhante a uma equipe de rugby (de onde vem o nome) treinando para o grande jogo, o Scrum estimula as equipes a aprenderem com as experiências, a se organizarem enquanto resolvem um problema e a refletirem sobre os êxitos e fracassos para melhorarem sempre [DRUMOND, 2022].

A ideia principal da Scrum é que o desenvolvimento de softwares envolve muitas variáveis técnicas e do ambiente, como requisitos, recursos e tecnologia, que podem mudar durante o processo. Isto torna o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças. O resultado do processo deve ser um software que é realmente útil para o cliente [SOARES, 2022, p. 5].

A plataforma utilizada para a utilização da metodologia ágil foi o Trello, ferramenta criada para organizar projetos pessoais e corporativos. Funciona como um painel de gerenciamento de projetos, permitindo a personalização de fluxo de trabalho para uso individual ou em equipe.

O desenvolvimento foi dividido em três quadros principais: Em desenvolvimento, A fazer e Concluído. As tarefas eram movidas de quadro em quadro de acordo com o seu status no processo de desenvolvimento do projeto.

Figura 5 - Quadro de desenvolvimento - A fazer



Fonte: Autoria Própria (2022)

A Figura 5 demonstra um dos quadros utilizados no desenvolvimento do trabalho para ilustração.

4.2 Criação do roteiro

Foi feita uma análise junto com a equipe da pró-reitoria para identificar quais eram as maiores dificuldades dos alunos durante o processo de inscrição em seus editais. Os resultados da pesquisa foram responsáveis pela criação de duas categorias de informações que foram implementadas no chatbot: conhecimentos extras e conhecimentos técnicos. Os conhecimentos técnicos englobam todas as informações pertinentes à inscrição no edital do auxílio creche, onde o chatbot atuou, com dados como datas de inscrição, cronogramas, quem pode concorrer a vaga e afins.

Em busca de tornar o comportamento do chatbot o mais próximo de um humano possível e apresentar ao usuário mais informações úteis sobre a PRAE, foi desenvolvida a área de conhecimentos extras. Essa área contém uma série de sub-áreas externas ao edital, responsável por tornar o *chatbot* mais sociável. Se inclui aqui coisas como as boas-vindas, agradecimentos, redes sociais, entre outros.

4.3 Implementação no Dialogflow

Essa foi uma das mais importantes etapas no desenvolvimento do *chatbot*, pois é aqui que definimos passo a passo como a engine irá se comportar durante toda a interação com o usuário final. Para tornar a implementação o mais interativa possível, o Google disponibiliza uma interface totalmente gráfica e intuitiva para a definição das regras e ações.

Primeiramente foram definidas as possibilidades de iniciação de diálogo, analisando, a partir do roteiro mencionado no capítulo anterior, as possíveis intenções dos usuários ao entrar em contato com o *chatbot*. No total foram definidas quatro possíveis intenções de um usuário ao entrar em contato com o *chatbot*:

- Iniciar um diálogo;
- Saber sobre o edital;
- Saber sobre a PRAE;
- Agradecer.

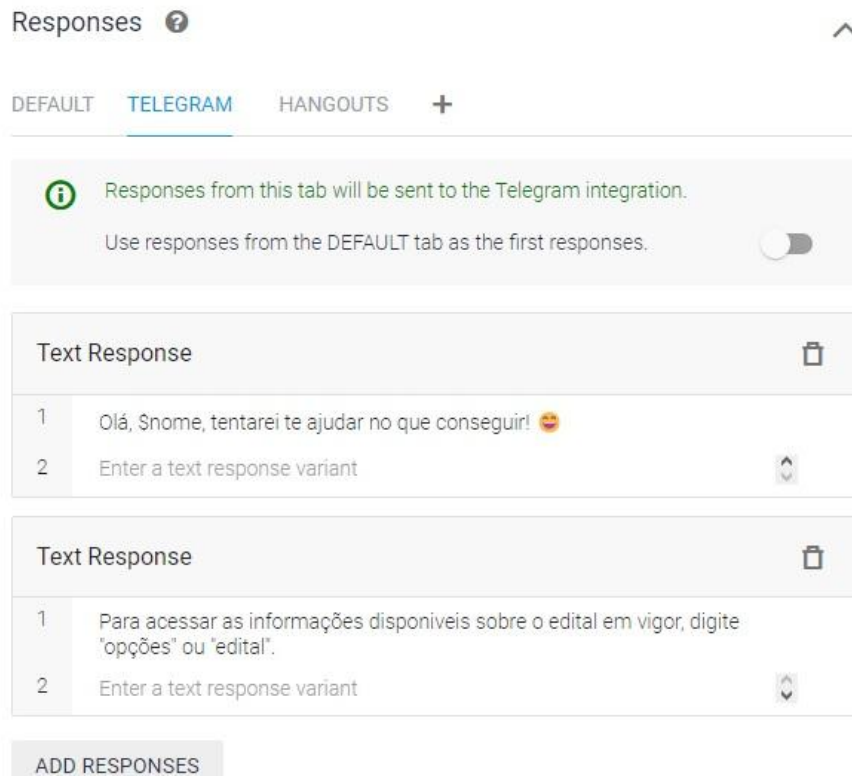
Ao se iniciar um diálogo, a frase de boas-vindas pede para que seja informado um nome, para melhor identificação dos usuários. Reconhecer a intenção do usuário é um dos passos mais importantes no desenvolvimento da conversa, pois é a partir daí que será possível definir as possíveis ramificações que aquele diálogo poderá gerar. Contudo, para que haja um melhor esclarecimento e facilidade no uso do *chatbot*, também foi disponibilizado um pequeno texto que identifica a sua funcionalidade, informando ao usuário uma maneira fácil e rápida de acessar as informações pertinentes ao edital através da escrita da palavra “edital”.

As respostas são divididas em duas categorias:

- **Text Response:** As *text responses* são literalmente o que a palavra indica, respostas de texto. É a forma mais simples de se responder ao usuário e suporta apenas respostas estáticas, podendo-se também fazer referência a algum parâmetro do diálogo através do símbolo \$.
- **Custom Payload:** É uma forma mais dinâmica e editável de se responder a requisições, fortemente customizável de acordo com a plataforma utilizada para conexão com o usuário final. O formato de edição e utilização do custom

payload depende da integração, com funcionalidades e notações diferentes de acordo com o ambiente.

Figura 6 - Intenção de boas-vindas



Fonte: Autoria Própria (2022)

A Figura 6 demonstra o funcionamento básico das intenções programadas na plataforma. Quando ela era correspondida pelo *Dialogflow*, era retornado para o usuário o texto pré-definido nas *Text Response* e *Custom Payload*, caso houvesse.

Informações como o nome do usuário, sua cidade, departamento, entre outros, são retiradas casualmente durante o seguimento do diálogo, tornando necessário a utilização e criação de entidades que tratam esses dados para que não sejam perdidos no decorrer da conversa. Foram criadas três entidades extras, além das entidades do sistema que tratam números e nomes, para armazenar dados customizados, como o nome do curso do usuário.

Para que esses dados pudessem ser retirados da maneira mais casual possível, sem que o fluxo de conversa com o usuário fosse interrompido, perguntas eram feitas no decorrer da conversa caso a inteligência artificial detectasse que

havia espaço. Exemplo disto é a intenção de boas-vindas, que pede para que o usuário informe o seu nome ao início da sessão.

Figura 7 - Entidades do sistema



Fonte: Autoria Própria (2022)

A Figura 7 mostra as entidades customizadas criadas especificamente para o desenvolvimento deste trabalho. O campo Campus faz referência aos diferentes campos da universidade do estado do rio grande do norte, Cursos sobre os diferentes tipos de cursos de graduação e Departamento sobre os diversos departamentos que constituem a instituição.

Cada interação do usuário com o chatbot gera uma saída de contexto, que é carregada para a próxima etapa da conversa para que os dados armazenados não sejam perdidos. Por exemplo, ao digitar “edital”, o usuário é então redirecionado para um menu de opções que apresenta as dúvidas que o chatbot é capaz de responder no momento, gerando assim um contexto de edital para as possíveis respostas naquela etapa da conversa. Entretanto, o contexto não limita a resposta exclusivamente a assuntos sobre o edital, apenas faz com que o chatbot entenda que, possivelmente, as próximas etapas do diálogo possam estar se tratando de alguma das opções que foram apresentadas.

A Figura 8 é responsável por demonstrar todas as intenções criadas para tratar as dúvidas referentes ao Edital, definidas no Capítulo 4.2, criação de roteiro.

Figura 8 - Intenções sobre o edital

<input type="radio"/> [Edital] Auxilio
<input type="radio"/> [Edital] Banco
<input type="radio"/> [Edital] Cronograma
<input type="radio"/> [Edital] Documentacao ▾
<input type="radio"/> [Edital] Duvida - Nao ▾
<input type="radio"/> [Edital] Duvida - Sim
<input type="radio"/> [Edital] Etapas
<input type="radio"/> [Edital] Inscricao
<input type="radio"/> [Edital] Opcoes
<input type="radio"/> [Edital] Requisitos
<input type="radio"/> [Edital] Requisitos - Nada consta
<input type="radio"/> [Edital] Valor

Fonte: Autoria Própria (2022)

Após a resolução de cada dúvida, é dada a oportunidade de feedback para o usuário, onde ele pode informar se a sua pergunta foi respondida de forma satisfatória ou não. Caso isso aconteça e a informação for passada ao chatbot, uma resposta é dada de acordo com o resultado: se a resposta foi respondida satisfatoriamente, é retornada uma mensagem de agradecimento e a disponibilização de retorno ao contexto anterior, para que, caso necessário, possa-se tirar mais dúvidas sobre o edital. Se não foi respondida satisfatoriamente, é feito um pedido de desculpas e apresentado ao usuário outras formas de comunicação com a PRAE, onde ele poderá ser atendido por um atendente humano que será capaz de sanar suas dúvidas com mais precisão. Contudo, caso nenhuma resposta seja dada, o usuário poderá continuar conversando normalmente sem nenhuma interferência no fluxo do diálogo.

A Figura 9 demonstra a estruturação do texto de resposta em caso de *feedback* negativo por parte do usuário após ser correspondido pelo agente do *chatbot*.

Figura 9 - Intenção de feedback negativo

Text Response	
1	Que pena! 😞 Espero poder te ajudar em outros assuntos.
2	Enter a text response variant

Text Response	
1	Você pode digitar edital/opções para tirar outras dúvidas ou nos contatar pelos demais contatos abaixo. Tenho certeza que eles poderão te ajudar, ok? 😊
2	Enter a text response variant

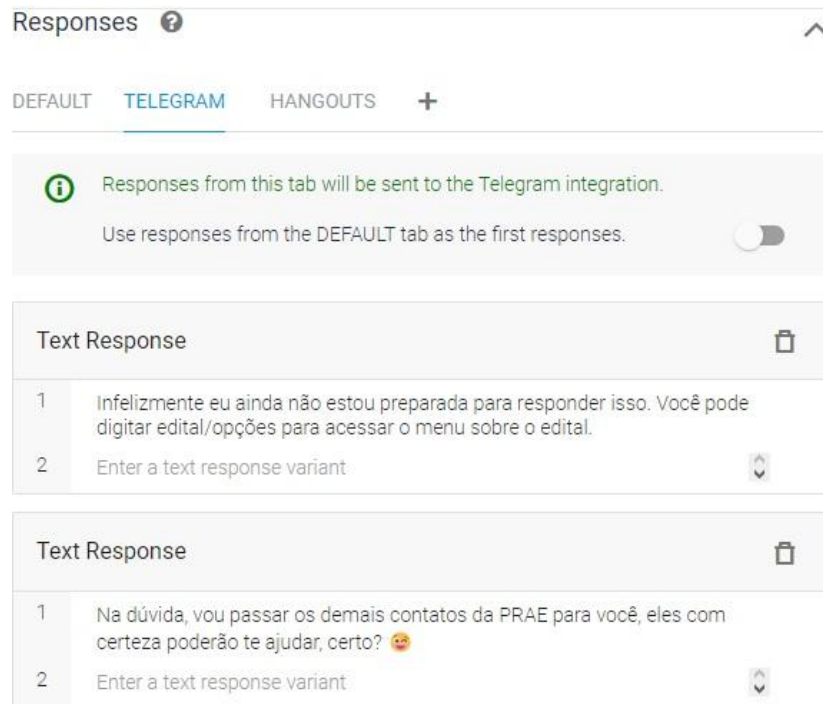
Custom Payload	
1	{
2	"telegram": {
3	"text": "✉️ *E-mail*: sae.prae@uern.br",
4	"parse_mode": "Markdown"
5	}
6	}

Fonte: Autoria Própria (2022)

Nos casos em que o chatbot não é capaz de conectar a frase do usuário a nenhuma das intenções programadas na plataforma, o *fallback intent* é acionado. Ele é responsável por definir uma ação padrão que será tomada sempre que o agente se deparar com um diálogo ao qual não esperava ou não foi desenvolvido para responder, informando ao usuário formas externas de comunicação com a pró-reitoria para que o assunto possa ser resolvido, semelhante ao que acontece caso a resposta a uma dúvida não tenha sido satisfatória.

A Figura 10 demonstra a estruturação das *Responses* do agente caso a intenção não fosse reconhecida pela plataforma. O usuário final era redirecionado para outras formas de contato com a pró-reitoria após um pequeno texto explicando a situação e o porquê da resposta não ter sido respondida corretamente.

Figura 10 - Fallback intent - Text Response



Fonte: Autoria Própria (2022)

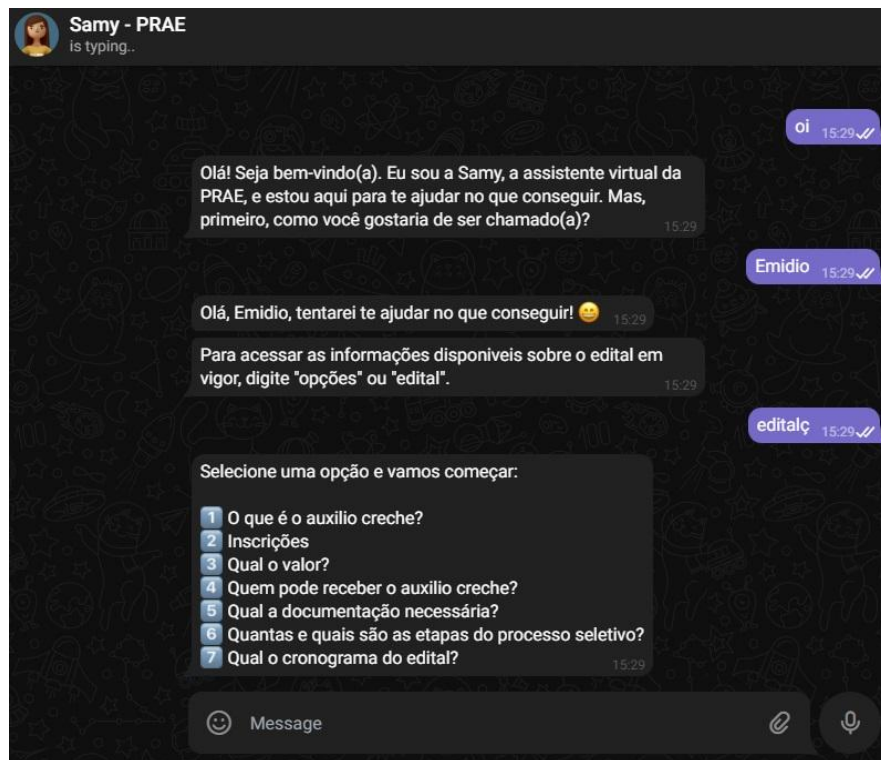
4.3.1 Interface humano-computador

A implementação de uma interface interativa e de fácil acesso é muito importante para a usabilidade e teste do *chatbot*, sendo um dos fatores mais determinantes para o seu sucesso. O *Dialogflow* oferece gratuitamente várias formas de conexão a interfaces através da opção integrations, incluindo o *Telegram* e o *Hangouts chats*, plataformas escolhidas para o desenvolvimento do trabalho.

Utilizar essas plataformas como forma de conexão com o usuário final traz a facilidade de conexão com o chatbot através do smartphone, por intermédio do *Telegram*, ou do e-mail institucional, através do *Hangouts*.

Todas as interações com o usuário são feitas através de chat de texto, sendo essa a única maneira de se comunicar corretamente com o *chatbot*, independente da plataforma utilizada para tal. Devido às limitações de personalização do *Dialogflow*, foi desenvolvida apenas uma imagem de perfil e nome, não sendo possível a implementação de um ambiente fora dos padrões das plataformas de comunicação com as ferramentas atuais.

Figura 11 - Telegram



Fonte: Autoria Própria (2022)

A Figura 11 simula a conversa entre um usuário final e o chatbot através do Telegram.

Delimitar o acesso a aplicação por e-mail ao domínio institucional da UERN faz com que o público alvo seja muito bem definido e evita o uso indevido do *chatbot* por usuários mal-intencionados, uma vez que apenas os discentes da instituição tem acesso a essa forma de interação com a plataforma de respostas.

4.4 Criação do webhook

O *webhook* é responsável por realizar a conexão entre o chatbot e o banco de dados. Seu funcionamento é simples, ele aguarda por requisições do método POST oriundas do *Dialogflow* assim que a intenção de *feedback* é reconhecida, conectando-se com o banco de dados e, para a resposta recebida, salva o seu resultado e a retorna para a plataforma.

Figura 12 - Código do Webhook

```

# Recebendo e enviando respostas para o Dialogflow
@app.route('/webhook', methods=['POST'])
@cross_origin()
def webhook():
    req = request.get_json(silent=True, force=True)
    res = processRequest(req)
    res = json.dumps(res, indent=4)
    print(res)
    r = make_response(res)
    r.headers['Content-Type'] = 'application/json'
    return r

# Processando as requests do Dialogflow
def processRequest(req):
    log = Conversations.Log()
    sessionID = req.get('responseId')
    result = req.get("queryResult")
    intent = result.get("intent").get('displayName')
    query_text = result.get("queryText")
    bot_message = result.get("fulfillmentText")
    parameters = result.get("parameters")
    db = ConfigureDatabase()

    if intent == "[Edital] Duvida - Sim" or intent == "[Edital] Duvida - Nao":
        log.saveConversations(sessionID, query_text, intent, bot_message, db)

```

Fonte: Autoria Própria (2022)

A Figura 12 mostra o trecho principal do código de desenvolvimento do *webhook*.

4.5 Conexão com o banco de dados

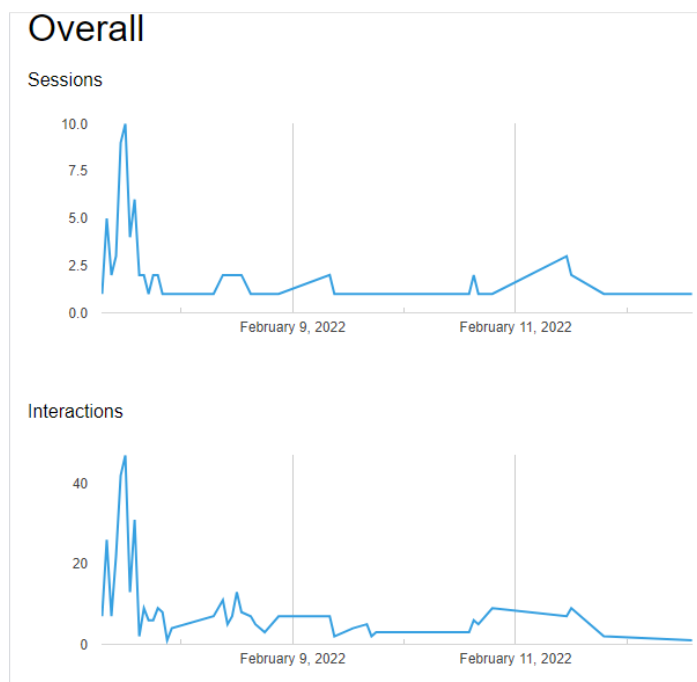
A conexão com o banco de dados é feita sempre que a intenção de *feedback* é correspondida e o *webhook* é acionado. Ela recebe as informações que devem ser salvas do *Dialogflow* através do método POST e configura a conexão do aplicativo principal com o banco de dados sempre que é acionada. Os dados ficam salvos em formato de dicionário, com cada variável correspondente a um campo, e são armazenados de acordo com o número da sessão em uma coleção no MongoDB.

5 RESULTADOS

Para a apresentação dos resultados obtidos pelo *chatbot*, foi escolhido o intervalo entre o dia 7 de fevereiro de 2022 até 13 de fevereiro de 2022, tempo em que as inscrições para o auxílio creche foram disponibilizadas.

Durante esse intervalo de tempo, o *Dialogflow* recebeu cerca de noventa e uma requisições, provindas de até setenta e sete sessões únicas. Teve um maior pico de utilização no primeiro dia de disponibilização da ferramenta, como demonstra o gráfico da Figura 13. Esse grande número de requisições em relação ao número de sessões únicas iniciadas demonstra certa recorrência de usuários, o que justifica o grande número de dúvidas respondidas pelo chatbot.

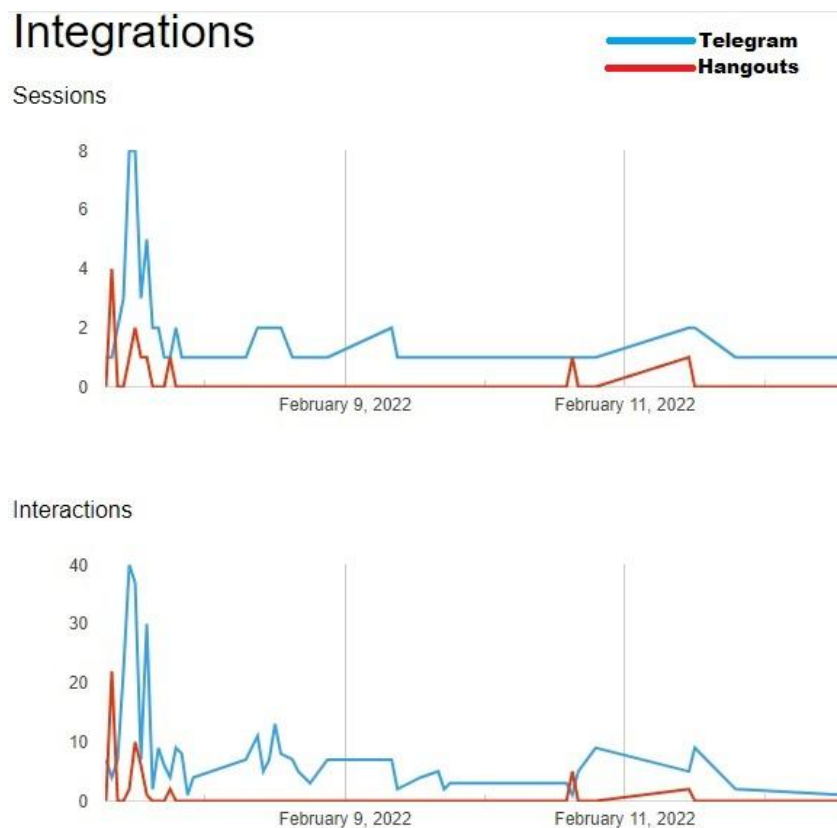
Figura 13 - Gráfico de utilização do chatbot



Fonte: Dialogflow (2022)

Como é mostrado no gráfico da Figura 14, a maior parte da interação com o usuário final foi feita através do *Telegram*, com apenas algumas poucas requisições atendidas através do e-mail institucional pela plataforma *Hangouts*. Em números, cerca de 80% das requisições foram atendidas somente pelo Telegram.

Figura 14 - Gráfico de utilização das integrações

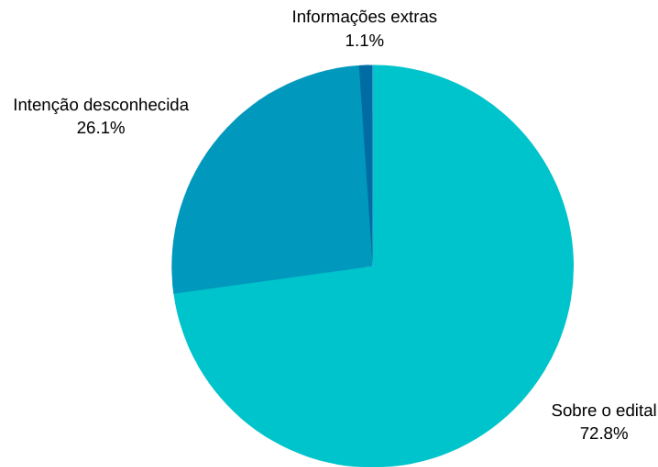


Fonte: Dialogflow (2022)

A Figura 14 demonstra melhor os meios de comunicação do usuário final com o *chatbot*. A linha azul no gráfico representa as interações feitas através do *Telegram*, enquanto a vermelha é referente ao *Google Hangouts*.

Como é visto na Figura 15, das noventa e uma requisições recebidas, apenas cerca de 73% foram correspondidas pelo Dialogflow, necessitando-se de uma maior base de treinamento para a correspondência dos 26% restantes, que foram redirecionadas para a resposta padrão de *fallback intent*.

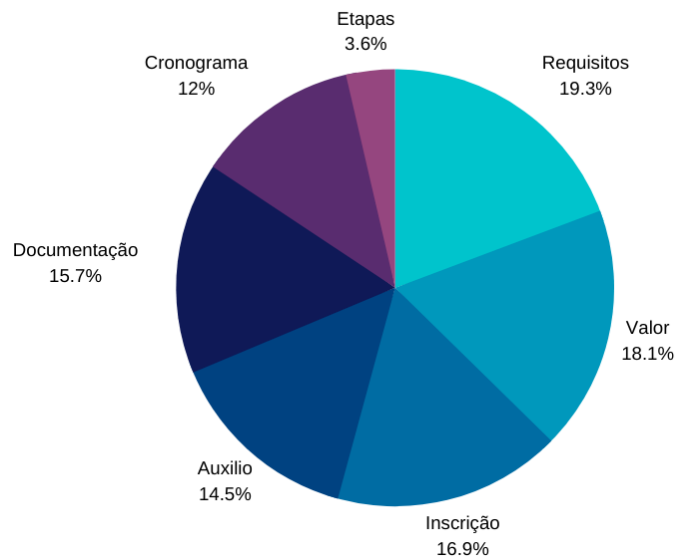
Figura 15 - Gráfico de requisições por intenção



Fonte: Aatoria Própria (2022)

Ao se analisar a Figura 16, houve uma boa distribuição na procura de dúvidas sobre o edital, sem nenhuma resposta sendo muito mais requisitada do que as demais. As duas principais dúvidas dos usuários sobre o auxílio creche foram sobre os requisitos para a inscrição e o valor da bolsa.

Figura 16 - Dúvidas respondidas



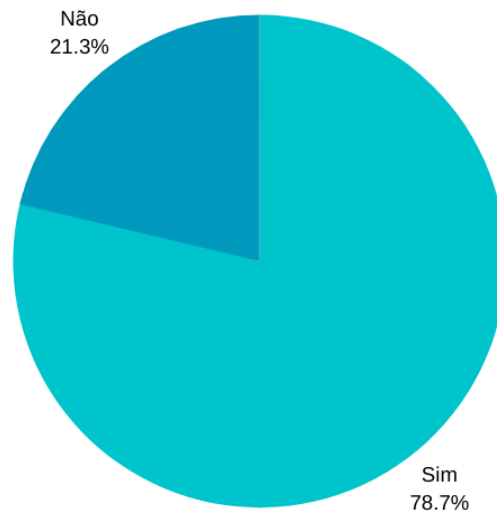
Fonte: Aatoria Própria (2022)

Foram recebidos quarenta e sete comentários dos usuários em relação a satisfação nas respostas do chatbot através do sistema de webhook. Dessas, trinta

e sete afirmaram que suas dúvidas foram sanadas. A Figura 17 mostra o resultado da avaliação.

Figura 17 - Satisfação dos usuários

A dúvida foi respondida satisfatoriamente?



Fonte: Autoria Própria (2022)

6 CONSIDERAÇÕES FINAIS

A utilização de *chatbots* para atendimento ao cliente pode trazer vários benefícios, como a redução da carga de trabalho e custos. Além disso, pode melhorar a satisfação dos usuários pela velocidade no atendimento e na resolução de problemas.

Ao se analisar os resultados obtidos, pode-se dizer que o *chatbot* cumpriu com o seu propósito, sendo capaz de atender a grande parte da demanda dos alunos por informações. Houve também uma grande satisfação da pró-reitoria na implementação do sistema em seus serviços, evidenciando a maturidade e escalabilidade do trabalho apresentado, uma vez que ele é composto por uma estrutura que pode ser facilmente modificada e projetada de acordo com as necessidades. Atualmente, se faz presente uma API, plataforma e interface de interação humano-máquina, além da disponibilização de um tutorial de desenvolvimento no *Dialogflow*, todos hospedados em nuvem.

6.1 Trabalhos Futuros

O objetivo principal deste trabalho foi criar uma arquitetura de desenvolvimento de baixo custo e simples entendimento capaz de monitorar, validar e responder a requisições, que se aplica não só ao problema proposto como também a qualquer outra dificuldade que abranja o acesso rápido a informações. Com isso finalizado, se faz proposto a melhoria das outras características da aplicação.

Em decorrência do tempo, não foi capaz realizar um teste de satisfação mais abrangente com os usuários da plataforma através de um formulário enviado por e-mail, ficando assim para futuros trabalhos a realização deste.

Como se trata de uma inteligência artificial, manter o *chatbot* em atividade com a adição de uma base de dados cada vez mais robusta é essencial para o seu crescimento, fazendo com que ele não só responda a questões específicas ao edital como também a outras dúvidas mais externas ao órgão de assuntos estudantis. Uma melhoria na API também pode ser considerada, fazendo-a armazenar não só o

feedback do usuário, como também todo o fluxo de conversa, que pode ser futuramente utilizado como alvo de treinamento do *chatbot*.

Outro diferencial seria a implementação do chatbot no site da pró-reitoria, como uma pequena caixa de diálogo que torna possível a comunicação do aluno com os serviços da mesma.

REFERÊNCIAS

GOMES, D. dos S. "Inteligência Artificial: conceitos e aplicações." *Olhar Científico*. v1 2 (2010).

TEIXEIRA, João de Fernandes. *Inteligência artificial*. São Paulo: Editora Paulus, 2014.

MOWBRAY, M. Ethics for Bots. Paper presented at the 14th International Conference on System Research, Informatics, and Cybernetics. Baden-Baden: jul.-ago. 2002. Disponível em: <http://www.hpl.hp.com/techreports/2002/HPL-2002-48R1.pdf>. Acesso em: 20 mar. 2022.

Construindo Chatbots com Python: Usando Natural Language Processing e Machine Learning. Brasil, Novatec Editora, 2019.

SINGH, Ajit. *Processamento de linguagem natural com Python: Simplesmente em profundidade*. Estados Unidos, Babelcube Incorporated, 2021.

BORGES, Luiz Eduardo. *Python para Desenvolvedores: Aborda Python 3.3*. Brasil, Novatec Editora, 2014.

THANH, Neos. *Flask Web Development: Developing Web Applications with Python*. Estados Unidos, O'Reilly Media, 2018.

CLARK, Jessica. O que é o Heroku?. Blog back4app, 2017. Disponível em: <https://blog.back4app.com/pt/o-que-e-o-heroku>. Acesso em: 20 mar. 2022.

KOVACS, Leandro. O que é e para que serve o MongoDB?. Tecnoblog, 2021. Disponível em: <https://tecnoblog.net/responde/o-que-e-e-para-que-serve-o-mongodb>. Acesso em: 20 mar. 2022.

COUTINHO, Thiago. Quais são as melhores linguagens para inteligência artificial?. Voitto, 2021. Disponível em: <https://www.voitto.com.br/blog/artigo/linguagens-para-inteligencia-artificial-ia>. Acesso em: 23 de mar. 2022.

SANTOS, Julio Cesar. Qualidade No Atendimento Ao Cliente. Brasil, Clube de Autores, 2011.

KAUFMAN, Dora. A inteligência artificial irá suplantar a inteligência humana?. Brasil, Estação das letras e cores, 2019.

ALPAYDIN, Ethem. Machine Learning: The New AI. Reino Unido, MIT Press, 2016.

BULLOCK, Bennett, e SILVEIRA, Guilherme. Machine Learning: Introdução à classificação. Brasil, Casa do Código, 2017.

DRUMOND, Claire. Scrum. Atlassian, 2022. Disponível em:
<https://www.atlassian.com/br/agile/scrum>. Acesso em: 29 mar. 2022.

LECUN, Yan; et al. Deep Learning. Nature, 2015.

BRITO, Felipe Neves. Desenvolvimento de um chatterbot para a página web de um curso de nível superior. Orientadora: Luciana Bolan Frigo. 2018. 78 f. TCC (Graduação) – Curso de Engenharia da Computação, Departamento de Engenharia de Computação, Universidade Federal de Santa Catarina, Araranguá, 2018.

Disponível em:
<repositorio.ufsc.br/bitstream/handle/123456789/187968/Trabalho-versao-final-Roger-Florzino-de-Souza.pdf>. Acesso em: 10 mar. 2022.

MONTEIRO, Guilherme Souza. Helena: Um chatbot para auxílio dos discentes do decom em trâmites universitários. Orientador: Jadson Castro Gertrudes. 2021. 58 f. TCC (Graduação) – Curso de Ciência da Computação, Departamento de Computação, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Rio Preto, Ouro Preto, 2021. Disponível em:
https://monografias.ufop.br/bitstream/35400000/3331/6/MONOGRAFIA_HelenaChatbotAuxilio.pdf. Acesso em: 12 mar. 2022.

What is ai?. ORACLE, 2022. Disponível em:
<https://www.oracle.com/br/artificial-intelligence/what-is-ai/>. Acesso em: 20 mar. 2022.

APÊNDICES

Apêndice A – Material Complementar

Aqui estão listados links para materiais complementares para apreço do trabalho.

- Repositório com o código fonte: github.com/EmidioLP/ChatbotPRAE
- Roteiro utilizado: bit.ly/3xgBUIL
- Tutorial do Dialogflow: github.com/EmidioLP/Tutorial_Dialogflow