Fabricio Carvalho Pacheco

Estudo e desenvolvimento de um ChatBot para automação de atendimento ao cliente

UBERLÂNDIA

2021

Estudo e desenvolvimento de um ChatBot para automação de atendimento ao cliente

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Câmpus Santa Mônica, como requisito para a obtenção do título de graduação em Engenharia de Controle e Automação

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Josué Silva de Morais

UBERLÂNDIA

2021

Pacheco, Fabricio Carvalho
Estudo e desenvolvimento de um ChatBot para automação de atendimento ao cliente/ **Fabricio Carvalho Pacheco** – **UBERLÂNDIA**, **2021**- 63 p.; 30 cm.

Orientador: Prof. Dr. Josué Silva de Morais

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia – UFU Faculdade de Engenharia Elétrica . **2021**.
Inclui bibliografia.

1. chatbot 1. 2. PLN 2. 2. Inteligência Artificial 3. I.Josué Silva de Morais. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

RESUMO

Diante da enorme demanda de atendimento pelos canais digitais devido ao acelerado processo tecnológico, o estudo e desenvolvimento para automação desta atividade via agente conversacionais torna-se essencial para o crescimento escalável de qualquer operação. Esse projeto visa discorrer sobre a natureza desses agentes, além do mais, trazer um comparativo prático sobre diferentes formas e tecnologias utilizadas na elaboração dos chatbots. Propõe-se, portanto, apresentar conceitos intrínsecos das aplicações e a maneira guiada de suas respectivas construções. Para validação do pleno funcionamento, esses agentes conversacionais foram exaustivamente testados a fim de obter melhores resultados e aprimoramentos.

***Palavras-chave**: chatbot, Processamento de Linguagem Natural (PLN), Inteligência Artificial (IA).*

ABSTRACT

Given the hude demand for customer service through digital channels due to the accelerated technological process, the study and development for automation of this activity via chatbots becomes essential for any operation scalability.That said, this project aims to discuss the nature of these agentes, in addition to providing a practical comparison of different forms and technologies used on chatbots development.Therefore, is proposed to presente the intrinsic concepts of the applications and the step-by-step way of their respective contructions. For validation purposes, these conversational agentes have been extensively tested for better results and improvements.

Keywords*: chatbot. Natural language processing (NLP) , Artificial intelligence (AI).*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com o desenvolvimento tecnológico o homem criou diversos métodos a fim de sanar suas necessidades e resolver seus problemas, com a criação do computador, esse se expandiu e passou a fazer parte de todos os espaços da sociedade moderna, desde empresas até residências. A partir do computador o mundo passou por uma transformação digital que impulsionou o surgimento de *softwares*, *smartphones*, aplicativos para assistência a esses e, a inteligência artificial.

No que tange o âmbito empresarial, a partir dessa revolução tecnológica, as empresas precisam modificar o atendimento a seus clientes e usuários, de acordo com Vasconcelos, Santos e Baldochi (2016) uma característica de negócios de sucesso é oferecer o que o cliente necessita no momento certo, sendo o atendimento fora do ambiente físico da organização uma das necessidades.

Para que empresas se sobressaiam no mercado, o atendimento virtual é um serviço essencial no relacionamento das empresas com seus clientes ou usuários, a grande dificuldade está em como lidar com os clientes e prestar o melhor atendimento para os mesmos. Uma pesquisa realizada no ano de 2020 pelo Grupo Sercom, provedor de serviços e plataformas de *customer experience* e feita em parceria com o instituto de pesquisas Qualibest, interrogou 1081 pessoas em todas as regiões do brasil a fim de identificar as formas de contato do cliente com a empresa. O chat pelas plataformas do suporte é o meio mais escolhido pelos consumidores, com 51% das preferências, seguido pela tendência no aplicativo mais popular de mensagens WhatsApp, com 49%. O e-mail fica na terceira colocação (47%), e o telefone perdendo cada vez mais seu poderio, com 43%, ficando na quarta colocação.

Porém, em geral, os clientes precisam esperar um tempo considerável para receberem o atendimento independente do canal empregado. Esse tempo pode aumentar quando o cliente escolhe utilizar o *chat* ao vivo, onde os atendentes conversam com vários usuários ao mesmo tempo. Implicando assim em um atendimento inadequado e respostas irrelevantes, que não atendem as necessidades ou sanam as dúvidas dos clientes. A fim de garantir o atendimento ao cliente por esses canais, as empresas passaram a substituir o atendente humano, pelo uso de *chats* automáticos.

Assim, surgiram os *chatbots* gerados dela demanda em estreitar a relação e garantir uma comunicação imediata, aplicando a capacidade de respostas de

empresas a usuários. Farias (2020, p. 11) destaca que os *chatbots* proporcionam a interação “entre homem e máquina, agilizando e/ou direcionando o serviço prestado de maneira fluida”. Complementam Khanna et al. (2015) que os *chatbots* são interfaces em linguagem natural, por meio do uso de recursos de Processamento de Linguagem Natural (PLN), que proporcionam a comunicação entre humanos e máquinas de forma natural como a linguagem humana.

A gigante do varejo Magazine Luiza vem investindo fortemente na Lu, influenciadora digital da empresa, que é a cara da interação com os clientes por seus canais de *ChatBot.* Dados das empresas apontam que somente no primeiro semestre de 2020, a Lu efetuou 1,4 milhão de atendimentos, com mais de 8 milhões de interações. O resultado disso são que 20% dos atendimentos foram resolvidos em primeira intenção e cerca de 60% sequer precisaram consultar o SAC (Serviço de Atendimento ao Cliente), demonstrando uma grande força e efetividade no atendimento ao cliente (REDAÇÃO NAMA, 2018).

Maciel (2019) destaca que a necessidade de interação usuário-sistema por meio da comunicação em linguagem natural usando interfaces computacionais vem crescendo como alvo de pesquisas empresariais e acadêmicas. Com o crescimento dessas pesquisas, facilitou-se a interação de pessoas com máquinas, fazendo com que a comunicação de humanos com sistemas computacionais se torne frequente no dia a dia.

Uma previsão feita pelo *Chatbots Journal* no ano de 2020, a tendência de tecnologia estratégica com foco em sistemas conversacionais foi superior a 85% dos meios de atendimento ao cliente, com *chatbots* capazes de realizar atendimento personalizado e reconhecimento de voz, face e comportamento de compra do cliente.

Assim, o uso dos *chatbots* se disseminou e se tornou abrangente, neste contexto se torna essencial estudar suas aplicações e eficiência se torna essencial a fim de aprimorar cada vez mais o sistema garantindo um melhor atendimento por parte das empresas aos clientes.

1.1 JUSTIFICATIVA

Conforme Araújo (2020) a aplicação de *chatbots* se estendeu para todos os segmentos e áreas como saúde, educação, empresas em geral, turismo dentre outras. Estes são capazes de responder perguntas sobre determinados domínios de

conhecimento e com isso facilitar o usuário em alguma determinada ação que ele queira realizar, ou mesmo fazer com que ele evite de ter de pesquisar em vários locais para encontrar uma dada informação.

No entanto, se faz essencial estabelecer um entendimento acerca de ferramentas e plataformas que possibilitem a construção desse tipo de aplicação de IA que está sendo utilizada em diversos meios e que sua necessidade e procura é crescente. Este estudo visa propor um entendimento do que são, como funcionam e como criar esses *chatbots*. Desde plataformas e frameworks atuais que permitam a criação de *chatbots*, até uma efetiva implementação, em que haja uma capacidade alta de personalização e manutenção de seu funcionamento. Ademais, é relevante disseminar conhecimento sobre as novas ferramentas de compartilhamento de conteúdo e aplicação da web semânica, que é a extensão da internet atual, garantindo uma interação soberana entre pessoas e computadores.

Justifica-se assim este estudo frente a importância social de desenvolver sistemas via *chatbot* que melhorem a eficiência de processos empresariais e organizacionais. Justifica-se ainda frente a importância acadêmica, de contribuir com os conhecimentos adquiridos durante realização desta pesquisa, servindo de base para novos estudos.

1.2 OBJETIVO

Este estudo tem como objetivo geral analisar o processo de criação de ferramentas de conversação, via um chatbot, com processamento de linguagem natural para Telegram e Whatsapp, a fim de agilizar o atendimento ao cliente em uma empresa do ramo de informática. Para isso, foram aplicados os seguintes objetivos específicos:

- Analisar sobre as tecnologias relacionadas a chatbot;
- Desenvolver ferramentas de conversação, via um chatbot, com processamento de linguagem natural para Telegram usando Python;
- Desenvolver ferramentas de conversação, via um chatbot, com processamento de linguagem natural para Whatsapp usando DialogFlow;
- Desenvolver ferramentas de conversação, via um chatbot, com processamento de linguagem natural para Whatsapp usando ChatRace;
- Analisar ambas as ferramentas desenvolvidas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ATENDIMENTO AO CLIENTE NO E-COMMERCE

O mercado consumidor moderno dispõe dos mais variados tipos de produtos a fim de satisfazer as demandas e necessidades dos consumidores. Assim, os clientes escolhem esses produtos, estabelecendo expectativas relacionadas ao valor e à satisfação que ele trará para si próprio. Quando o cliente se satisfaz com um determinado produto ou empresa, se torna um consumidor constante, comprando novamente e, ainda compartilham com outros sua boa experiência (KOTLER e ARMSTRONG, 2014).

Ghidini e Mattos (2018) destacam que a qualidade, em muitas situações, se torna a chave para conquistar um cliente e torná-lo fiel, assim ofertar serviços ou produtos com qualidade podem proporcionar a fidelidade do cliente. Mandelli (2014) complementa que dentro do cenário organizacional a qualidade é vital no decorrer da prestação de um serviço ou oferecimento de um produto, envolvendo todos os processos desde a produção até o pós-venda.

Ademais, se faz necessário incorporar serviços aos produtos e pelo oferecimento de um atendimento eficaz aos clientes, e que as empresas que forem capazes de manter o foco no consumidor e atenção às suas demandas atuais e futuras serão aquelas que prosperarão a longo prazo (SOUSA, 2011).

Esclarece Shiozawa (1993, p. 52):

> O atendimento ao cliente significa, portanto, tudo aquilo que, em conformidade com os requisitos, ou seja, o fornecimento dos produtos ou a prestação de serviços solicitados ajude a criar o produto ou serviço potencial. Esta definição ajuda a compreender que o cliente é um alvo móvel, ou seja, possui expectativas crescentes.

Shiozawa (1993) destaca ainda que o atendimento de qualidade proporciona uma grande vantagem competitiva para a organização. Sendo a satisfação um elemento capaz de fidelizar os clientes. Dessa forma, as organizações precisam buscá-la como meio de estreitar o relacionamento com sua clientela, tornando-os parceiros comerciais.

Sousa (2011) ressalta que o atendimento oferecido com qualidade não se restringe a cortesia ou fino trato. Mais do que isso, significa acrescentar benefícios a produtos e serviços visando superar as expectativas do consumidor. Falando em

atendimento, é comum relacioná-lo somente as questões de tratamento. Porém, Santos (2020) destaca que no mundo altamente globalizado e tecnológico é vital que as empresas busquem revolucionar o atendimento ao cliente. Assim, as empresas que apoiam seus negócios em alguma tecnologia de atendimento ao cliente, já estão colhendo resultados favoráveis resultados dessa revolução.

A aplicação de tecnologia no atendimento ao consumidor não é mais uma tendência, mas uma realidade que visa suprir as expectativas dos clientes quando ao relacionamento com a marca. Independente do setor de atuação, o consumidor espera que a tecnologia torne mais fácil os processos como tirar dúvidas, alteração de dados, consulta de preços, emissão de boletos, entre outros (SANTOS, 2020).

Conforme o Relatório de Tendências de Suporte ao Cliente 2021 (INTERCOM, 2020) 600 líderes de suporte foram entrevistados, desses, 73% afirmaram ser conscientes quanto ao aumento das expectativas dos clientes, no entanto somente 42% afirmaram acreditar que as estão atendendo. Ademais, a pesquisa identificou que nem toda tecnologia proporciona benefícios, já que 46% dos entrevistados relataram que a quantidade de tecnologias disponíveis atrasa os objetivos organizacionais e 44% destacaram que usam entre seis e dez ferramentas de atendimento.

No que tange o e-commerce, esse é um mercado que acende consideravelmente no Brasil, conforme o índice MCC-ENET (2020) o e-commerce nacional cresceu 73,88% no ano de 2020. A região Sudeste ocupa a primeira posição nacional, responsável por 63% dos pedidos online feitos no país. Já no primeiro trimestre de 2021 o e-commerce teve alta de 57,4% quando comparado ao primeiro trimestre do no de 2020 (MCC-ENET, 2020). Ademais, no ano de 2020 o setor movimentou cerca de R$87,4 bilhões, de acordo com Relatório Webshoppers 43 (EBIT/NIELSEN/BEXS BANCO, 2020).

Quanto ao atendimento ao consumidor do e-commerce, esse espera mais que uma boa compra, a experiência precisa ser plena e parte essencial dela é o atendimento. Esse possui cada vez mais valor, considerando que as exigências aumentam na era digital. Para atendê-las, boas práticas se tornam essenciais. A transformação digital trouxe mudanças significativas a sociedade, sendo assim aponta-se novos hábitos de consumo e na relação do público com as marcas. Apesar de o modelo de lojas online não ser recente, as novas exigências e expectativas dos clientes são. Nesse sentido se torna essencial que as empresas se adaptem aos que

seus consumidores buscam e o atendimento eficaz é vital nesse processo. Mesmo que uma parcela considerável dos consumidores já compre pela internet, boas experiências ainda faltam (SOUZA, 2018-1).

Assim, nesse cenário das vendas online, surgiu o conceito do consumidor 4.0, sendo esse um público diferenciado, munido de ferramentas modernas e capazes de realizam uma boa troca de informações como critério de decisão. O consumidor 4.0 marca presença nos meios digitais, vive totalmente alinhado às inovações tecnológicas e se mantém antenado em relação às novidades do mercado. As pessoas desse perfil também apresentam um nível de exigência bem alto. Por isso, os empreendedores de sucesso já perceberam que o consumidor 4.0 tem um perfil diferenciado e não é facilmente atraído apenas por produtos famosos ou marcas de sucesso. Esse tipo de cliente exige maior dinamismo do mercado, flexibilidade nas negociações, multicanais para atendimento e plataformas variadas (SILVA, 2020).

> Ou seja, o Consumidor 4.0 não espera que as organizações cheguem até a ele, ou ofereçam algum produto ou serviço, pelo contrário, na maioria dos casos não gostam de ter suas ações interrompidas para atender uma chamada de telemarketing, ou ter um vídeo pausado para olhar um anúncio. A “Geração do Agora” segundo Kotler, são os consumidores que exigem e consomem tudo de forma instantânea. O consumidor 4.0 não espera que a marca chegue até ele, logo, prefere ir até a marca e tem a expectativa de que ela esteja no mesmo lugar que ele. Essa mudança já ocorre em muitas empresas, principalmente para se manterem presentes no mundo online, onde esses consumidores estão boa parte do tempo. Eles reagem mais rápido às transformações que ocorrem, principalmente as tecnológicas. E, não é à toa que a maior parte das ações são mais voltadas para eles e envolvem anúncios mais criativos e digitais. Isso porque, com inúmeras opções disponíveis, se o consumidor não obtiver suas dores e respostas atendidas rapidamente por uma empresa, não demora muito para que ele procure outra que possa resolver essas questões prontamente (LOPES, 2020, p. 9).

O Consumidor 4.0 têm pouquíssimo tempo para as empresas, deseja ser ouvido e encontrado rapidamente e quer se sentir especial. Se não obtiver respostas rápidas, encontra facilmente com o concorrente. O papel das empresas é reconhecer a interação dos canais on-line e off-line na obtenção do engajamento e da defesa das marcas junto aos consumidores. Sendo o ponto chave engajar esses consumidores, pois eles não suportam mentiras, odeiam respostas prontas e cobram mais humanização social e respostas que favoreçam o meio ambiente (CLAUDINO, 2018).

Quem vai a um site comprar espera ser envolvido em uma experiência completa, e isso envolve: uma boa comunicação; um ambiente digital fácil e agradável;

condições favoráveis de pagamento e envio; amplitude de informações; atendimento rápido, descomplicado e por vários canais. Para quem gere um e-commerce, prestar um atendimento eficiente é uma obrigação, claro, se a empresa pretende sobreviver por bastante tempo. É sempre importante lembrar que a concorrência cresce constantemente, já que são cada vez mais lojas online abertas. Assim, ganha em competitividade a empresa que é capaz de satisfazer o seu cliente além do bom produto ou serviço (SOUZA, 2018-2).

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

Nos mais variados segmentos, tornou-se frequente o uso de atendentes virtuais que “fazem as vezes” de um atendente de “carne e osso”. Esses atendentes se tornaram parte do cotidiano de diversas formas, seja por tele chamadas, chats de atendimento, ou ainda, respondendo sobre nossa agenda de compromissos do dia. Por trás dessa comodidade e facilidade, atua uma técnica complexa de ferramentas e conhecimentos, que atuam conjuntamente, garantindo o desenvolvimento de assistentes virtuais, destacando-se, dentre aqueles, o chamado Processamento de Linguagem Natural (PLN), em inglês *Natural Language Processing* (NLP) (DIAS, 2019).

Maldonado et al. (2016) denominam a PLN como uma área de pesquisa dentro da Inteligência artificial (IA) que visa estudar a comunicação, por meio da linguagem natural, entre o ser humano e a máquina (computador). BrahmaleenKaurSidhu (2013) complementam que a PLN busca explorar como o computador pode ser aplicado na compreensão e manipulação de textos ou falas em linguagem natural a fim de aplicá-las em coisas úteis. Como objetivo, o Processamento de Linguagem Natural, deve garantir a produção de um programa capaz de entender a linguagem humana e responder com a maior naturalidade possível (COMARELLA e CAFÉ, 2008).

A PLN começou a ser estudada e aplicada a partir de meados da década de 1950, a fim de aplicar a linguagem natural como estratégia de comunicação em computadores. Assim, Alan Turing, percursos na área, propôs o teste de Turing, sendo um teste de IA simulando por meio de máquina, o comportamento humano (SETZER, 2015). Othero e Menizzi (2005) explicam que os estudos na área avançaram frente a necessidade de criação de programas de traduções automáticas durante os anos de 1950 e 1960, e seu desenvolvimento relaciona-se também ao desenvolvimento na

área de IA.

No ano de 1966 Joseph Weinzenbaum criou um dos primeiros chatterbots, denominado de ELIZA, e era capaz de simular um diálogo entre um paciente e um psicólogo, por meio de ferramentas de PLN, é valido destacar que na época as pessoas acreditam que estavam conversando com um ser humano de verdade e não com um computador (FARINON, 2015). Apesar de ter sido criado na década de 1960, o realismo do diálogo é muito próximo a um diálogo entre humanos. O sistema proporcionou um avanço na computação, e a partir de então os estudos na área cresceram consideravelmente (MITTMANN, 2015).

Conforme Oliveira e Navaux (2004), ao se considerar um sistema baseado em PLN, precisa atender a duas condições: “Uma parte do sistema deve ser codificado em linguagem natural; O processamento de entrada e/ou saída é baseado em aspectos sintáticos, semânticos e/ou pragmáticos de uma língua natural”.

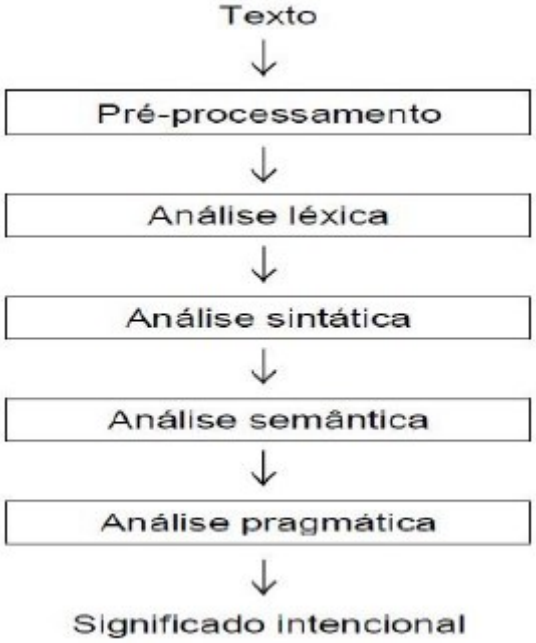
Santos (2018) destaca que o estudo da PLN é essencial no processo de relação entre a linguística e a informática, para assim conseguir aplicar na prática ferramentas como o chatbot com uma linguagem mais espontânea possível. Por meio desses avanços será possível construir sistemas que possuam a capacidade de reconhecer e reproduzir informações por meio da linguagem natural.

A análise PLN se classifica, tradicionalmente, por Dale (2010) em três categorias principais: a sintaxe, a semântica e a pragmática. Conforme Camara (2013, p. 10) elas se distinguem:

> A sintaxe trata da ordem e da estrutura. A semântica refere-se ao significado. Já a pragmática aborda o significado contextualizado. A pragmática tem foco no discurso, enquanto as demais preocupam-se com questões sentenciais. Essa classificação tem um propósito pedagógico, já que é difícil separar o processamento da linguagem em seus respectivos compartimentos.

De acordo com Dale (2010) as fases da abordagem de análise em PLN são apresentadas na Figura 1.

Figura 1 – PLN: fases de análise



Fonte: Dale, 2010

Hippisley (2010); Junglöf, Wirén (2010), Mellish e Pan (2008), esclarecem os objetivos das análises léxica, sintática e pragmática:

> Na análise léxica, o objetivo é estudar a morfologia das unidades lexicais, ou palavras, e recuperar informação que será útil em níveis mais profundos de análise (HIPPISLEY, 2010). A decomposição das palavras, assim como a detecção de regras de formação, permite a economia de espaço de armazenamento e aumenta a velocidade de processamento, considerando a hipótese simplista de armazenar cada unidade lexical encontrada em um repositório. (...) Já a análise sintática é aquela que se preocupa com a estrutura das sentenças em uma gramática formal. Um pressuposto em vários trabalhos de PLN é o de que o significado não se encontra nas palavras, mas sim na frase (LJUNGLÖF; WIRÉN, 2010) "(...) O componente pragmático, por fim, procura incluir o contexto à análise linguística, a fim de permitir a geração de um significado. Esse utiliza uma base de dados construída em um esquema de representação de conhecimento para representar o contexto externo do texto e permitir a utilização desse conhecimento para inferências automatizadas (MELLISH; PAN, 2008).

2.3 INTELIGÊNCIA ARTIFICIAL

Desde as proposições de Turing, que questionou a possibilidade de as máquinas pensarem, a IA tem recebido diversas definições. Charniak e McDermott (1985) destacam que a IA consiste no estudo das faculdades mentais por meio da aplicação de modelos computacionais. Complementam Kurzweil et al. (1990) como a arte de desenvolver máquinas que realizem atividades que demandam inteligência

quando realizadas por humanos.

A inteligência artificial (IA) cresce nas últimas décadas como meio de simular a inteligência humana por meio de códigos. Assim, a IA possui um potencial, através da programação, permite que atividades sejam realizadas sem a necessidade de intervenção humana, uma vez que o sistema se retroalimenta (FARIAS, 2020). Os sistemas inteligentes são diferentes dos tradicionais que lidam com dados numéricos, pois tem como objeto o processamento de ideias e de conhecimentos representados por simbologias (CUNHA e KOBASHI, 1991). Nesse contexto, as organizações o aplicam em sistemas de reconhecimento fácil e biométrico, ou para implementação de sistemas operacionais, visando como função um assistente pessoal (FARIAS, 2020).

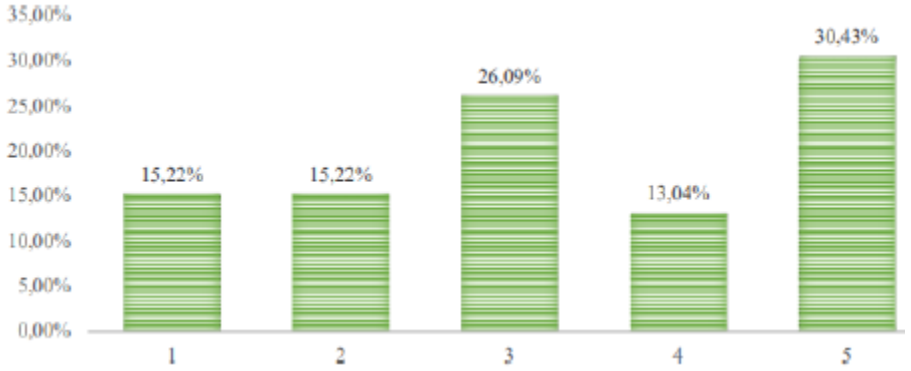
Complementa Teixeira (2014), que os comportamentos computacionais aplicados na IA incluem: raciocínio lógico-matemático, aprendizagem, linguagem, reconhecimento visual de formas e solução de problemas em geral. Dentro da IA existem diversas subáreas que visam solucionar inconvenientes específicos: representação do conhecimento, aprendizagem de máquina, processamento de linguagem natural, reconhecimento de padrões, resolução de problemas etc.

As contribuições e aplicação da IA são inúmeras, conforme Luger (2014), atuam em jogos, modelagem semântica, modelagem de desempenho humano, linguagem natural, planejamento, linguagens, robótica, algoritmos genéticos entre outros. Quanto as linhas de pesquisa, Azevedo (2005) destaca: a linha conexionista, a linha simbólica e a linha evolutiva.

> A primeira linha conexionista propõe a modelagem da inteligência humana por meio de simulações dos neurônios e suas interligações. A linha simbólica, aplica o formalismo do tipo lógico para simular o comportamento inteligente expresso por meio de linguagem. E a linha evolutiva, ou computação evolutiva, se baseia na observação de mecanismo evolutivos encontrados na natureza, tais como a auto-organização e comportamento adaptativo (FARIAS, 2020, p. 15).

Hirt (2019) pesquisou a respeito da percepção dos usuários em relação a aplicação da IA no cotidiano, conforme resultados ilustrados na Figura 2.

Figura 2 – Frequência do uso da IA



Fonte: Hirt et al., 2019

Considerando 1 como “nunca” e 5 como “sempre”, observa-se que 30,43% afirmam que sempre utilizam, enquanto 15,22% disseram nunca utilizar. Os 54,35% restantes fazem uso, porém não com alta frequência. Isso indica que mesmo o uso não sendo frequente, mais de 80% dos respondentes utilizam a inteligência artificial no cotidiano.

2.4 CHATBOT

2.4.1 Histórico

Ayres (2018) defende que, ao contrário do que se pensa, os robôs de conversação (*chatbots*) não são inovações exclusivas do século XXI. A história do *chatbot* teve início com o conceito de ficção científica, em 1940, quando Isaac Asimov publicou o livro Eu Robô, uma série de histórias sobre robôs inteligentes que pensam e conversam (INBOT, 2018).

Uma década mais tarde, em 1950, Alan Turing publicou o artigo *Computing Machinery and Intelligence*, que ganhou se destacou por propor a questão: “*seriam as máquinas capazes de pensar?*“. A fim de responder a essa questão, Turing desenvolveu um método para analisar a capacidade de uma máquina se comportar de modo semelhante a um ser humano. O método foi chamado de Teste de Turing (ou Jogo da Imitação) e se tornou pioneiro nos experimentos de *chatbots* criados no campo acadêmico a priori (AYRES, 2018).

No entanto, a tecnologia mais próxima a que conhecemos hoje, foi apresentada em 1966 por Joseph Weizenbaum, pesquisador do Instituto de Tecnologia de Massachusetts (MIT), quando esse criou do *software* denominado de Eliza. Eliza é então considerada a mãe de todos os bots, sendo desenvolvida a fim de simular uma

psicóloga (WENI, 2018).

Conforme Weni (2018) Eliza foi programada para replicar conversas humanas com base em instruções e respostas pré-definidas, esse robô de conversação tinha capacidade para identificar em média 250 frases. Porém, Eliza falhou durante o Teste de Turing, e com isso, como psicóloga automatizada. No entanto, seu desenvolvimento demonstrou um avanço significativo, impulsionando diversos experimentos até alcançarmos a tecnologia de *chatbots* de hoje.

Em 1984, no Brasil, foi criado o *chatbot* chamado de Carla, uma psicóloga virtual inspirada em Eliza, esse estudo foi publicado na Revista Microhobby n.12 (Figura 3) (INBOT, 2018).

Figura 3 – Publicação do Carla na Revista Microhobby n.12

Fonte: INBOT, 2018

Já em 1995, foi criado o Artificial Linguistic Internet Computer Entity, populamente conhecido como A.L.I.C.E. Esse sistema procurava por palavras-chave na pergunta feita pelo usuário e respondia com uma frase pré-estabelecida. Cada parte da conversa era programado de modo individual por meio de uma linguagem chamada AIML (INBOT, 2018).

No primeiro ano do século XXI, desenvolveu-se o Sete Zoom, modelo virtual com recursos de IA, atuava como garota propaganda e *hostess* no *website* da marca de pastas de dentes *Close Up*, que tinha capacidade de manter uma conversa de até 1 hora com os internautas sem que eles percebessem que a conversa era com um programa computacional. Sendo esse o primeiro chatbot a conversar por mensagens instantâneas. Interagia pelo site, pelo icq e pelo desktop no Windows..O engine usado foi a primeira versão do Inbot, desenvolvido pela Insite. Possuia conhecimento sobre cerca de 5000 assuntos diferentes (INBOT, 2018).

O primeiro bot a conversar por SMS foi o Tim Blah, criado em 2003, como se

fosse uma pessoa real. Foi criado com o Inbot, consiste em uma evolução da Sete Zoom. A Petrobras lançou em 2004 o Robô Ed, assistente virtual em um de seus sites. Era programado para responder milhares de perguntas variadas (INBOT, 2018).

A partir de 2011 diversas empresas começaram a desenvolver esse sistema, com destaque para o lançamento da Alexa Skills pela Amazon em 2015; pelo Duer lançado pela Baidu, também em 2015; Microsoft lança Cortana Intelligence Suite em 2016; Facebook lançou a integração de bots com messenger em 2016; e o Google cria o SyntaxNet, Google Now e outros serviços de IA.

2.4.2 Conceitos

Chatbot é um tipo de Agente Conversacional (AC) que interage com o usuário por meio da linguagem natural (ABUSHAWAR; ATWELL, 2015). Esse tipo de sistema simula uma conversa inteligente com o usuário, e tem o objetivo de mantê-la de forma natural e coerente (ALMEIDA, 2017). Calado (2016) complementa que se trata de *softwares* que visam interagir com os humanos de um modelo humanizado e executar tarefas em uma determinada área do conhecimento. Para Vogel (2017) Um *chatbot* pode ser entendido como um programa de computador autônomo que interage com usuários ou sistemas on-line, em tempo real, na forma de conversas, muitas vezes lúdicas e informais.

Chatbot é uma palavra que vem do inglês, chat, em português bate-papo, e bot, da palavra da língua inglesa, robot, em português robô (LIMA, 2014). Martínez-Miranda (2010) destaca os propósitos para os quais esses sistemas são criados: assistentes virtuais, agentes tutores e companheiros virtuais. E são empregados em diferentes domínios, exemplos: comércio, entretenimento, educação e saúde.

Mislēvičs (2018) explica que a depender da natureza das conversas, os *chatbots* se classificam entre domínios abertos e domínios fechados. Conforme Araújo (2020, p. 19):

> *Chatbots* de domínio aberto podem participar de uma conversa de forma livre com um usuário, não tendo nenhuma meta específica definida. Enquanto *chatbots* de domínio fechado são criados para ajudar um usuário a atingir uma meta específica.

Ademais, Pereira e Pinheiro (2018) afiram que podem ainda ser classificados

em: *searchbots*, que auxiliam a filtrar e procurar sites na internet, *mailbot*s, utilizados para classificar e responder perguntas via e-mail, *modbots*, responsáveis por moderar fóruns de discussão online e, por fim, os *chatbots*, que funcionam como simuladores de conversação.

Segundo Filho (2009) complementa que existe uma diversidade de *chatbot* e são nomeados em conformidade com as características que apresenta, como exemplo: *academic bots* (relacionado a assuntos acadêmicos), *commerce bots* (desempenhar atividades de comércio), *fun bots* (com a finalidade de divertir através de jogos), *government bot* (informações governamentais), entre outros.

O funcionamento de um *chatbot* ocorre por meio da identificação de uma palavra-chave ou de um contexto mínimo que possua sequência de palavras-chave. Assim, baseado na manipulação dessas palavras que o mecanismo se baseia para atuar de maneira inteligente (ZAMBIASI e PINHEIRO, 2016).

Franklin E Gresser (1996 apud Santos, 2018, p. 11) destacam que os *chatbots*, assim como qualquer outro agente autônomo, devem garantir certas características:

> Reatividade (reagir a estímulos do ambiente), autonomia (possuir controle de ação), proatividade (ser capaz de tomar iniciativa), continuidade temporal (estar continuamente ativo), capacidade social (comunicação com outros agentes/pessoas), capacidade de adaptação (através do aprendizado), mobilidade (poder circular dentro do ambiente), flexibilidade (ser dinâmico nas ações de resposta) e caráter (imprimir personalidade e estado emocional).

2.5 PLATAFORMAS DE CHATBOTS

Uma questão a se considerar é que a construção de uma plataforma de chatbots é uma tarefa custosa. Além disso, existem várias plataformas já existentes no mercado que podem ser utilizadas. Isso não invalida a criação de um chatbot em uma plataforma existente, pois a foco maior está na construção da base de conhecimento. Dessa forma, ao utilizar uma plataforma de chatbots neste trabalho, traz-se uma maior praticidade (CHATBOTS BRASIL, 2018c).

Conforme o portal ChatBots Brasil (2018c), as plataformas de chatbots são sistemas que facilitam a criação de chatbots e sua integração a diversos tipos de serviços e canais e, também, integram com aplicações de mensagens, como redes sociais, APIs e ferramentas de análise de métricas.

A seguir, será visto alguns exemplos de plataformas de chatbots.

2.5.1 Watson Conversation da empresa IBM

Mazon (2018) afirma: “O Watson Conversation, especificamente, trata-se de uma API para desenvolvimento de Bots, com uma interface simples para que até mesmo uma pessoa que não seja de TI consiga desenvolver e ensinar conteúdo ao bot.” Essa plataforma utiliza vários modelos de processamento de linguagem natural, já cadastrados para vários idiomas, os conceitos que a plataforma segue são: as intenções, entidades e diálogo (MAZON, 2018).

Para Mazon (2018) uma intenção trata-se da ação atrelada às perguntas realizadas pelo usuário. Isto é, o que o usuário procura ao falar algo; e sim, podemos falar a mesma coisa de diversas maneiras, sendo praticamente impossível treinar todas as opções de interação. Deste modo, no Conversation, nós damos exemplos de frases, e posteriormente, o sistema generaliza para identificação de outras intenções comuns.

As entidades são os complementos de informação, em um exemplo simples de um chatbot que faz pedidos de pizza, as entidades podem ser: sabor da pizza, o tipo de massa, CEP e data de entrega. São informações que ajudam o chatbot a definir melhor o contexto e a resposta a ser dada ao usuário (MAZON, 2018).

Conforme Mazon (2018), os diálogos são processos de conversação dentro de um contexto, onde o chatbot reconhece uma sequência de caracteres ou palavras que o faz responder conforme o diálogo está definido em sua base de conhecimento.

2.5.2 Microsoft Bot Framework da empresa Microsoft

Microsoft Bot Framework é uma ferramenta de desenvolvimento de *chatbots* desenvolvida pela empresa Microsoft, que possibilita ao usuário utilizar o robô em diversas plataformas. É possível melhorar o *chatbot* através de ferramentas de PLN incluídas na plataforma da ferramenta. Para aumentar a usabilidade são fornecidas integrações com mensageiros populares como Skype, GroupMe, Telegram ou utilizar através da Web (HADDAD, 2018).

A plataforma é muito poderosa e permite que os desenvolvedores construam e conectem *bots* inteligentes pelo mundo, possuindo integração com calendários, e-mails, ferramentas de projeto e serviços web (POLLAK, ANDERST-KOTSIS, 2017).

É fornecido um meio de aprendizagem de máquina chamado Serviço

Inteligente de Entendimento de Linguagem (*Language Undestanding Intelligent Service* - LUIS), ao ser fornecida uma entrada de texto, a ferramenta processa os dados através de modelos matemáticos e estatísticos, fornecendo uma saída mais adequada para a conversação com o usuário (HADDAD, 2018).

2.5.3 Dialogflow (Api.ai) da empresa Google

Segundo Brandes (2017), “O Dialogflow é uma plataforma para construir interfaces de conversação para bots, aplicativos e dispositivos.”

A forma de trabalho da plataforma segue a linha do processo de envio de uma mensagem pela interface do chatbot para a plataforma, onde, nela são criados diálogos e respostas que são devolvidos à interface do bot. Os diálogos são criados a partir do processo de envio de uma query que é texto em linguagem natural ou um nome de evento enviado para a plataforma como dados de entrada, que é transformado, em seguida, em uma actionable data (dados acionáveis, ou seja, as respostas configuradas na plataforma) e retorna dados de saída (BRANDES, 2017).

Para Brandes (2017), “O processo de transformar a linguagem natural em dados acionáveis, como contextos e propriedades de intenções, é chamado *Natural Language Understanding* (*NLU*)”.

Brandes (2017) afirma que o Dialogflow segue os seguintes conceitos: agente, entidades, contextos, parâmetros, intenções. Os agentes são módulos NLU (Natural Language Understanding), seu objetivo é transformar o idioma natural do usuário em dados acionáveis. As entidades são conceitos que servem como uma poderosa ferramenta para extrair valores de parâmetros de entrada de linguagem natural. Os contextos são cadeias de palavras ou caracteres que representam o contexto atual do pedido de um usuário. Os parâmetros são usados nas ações para extrair informações das entradas de usuários. As intenções representam um mapeamento entre o que o usuário diz e quais ações devem ser tomadas pelo seu software.

2.5.4 Amazon Lex da empresa Amazon

Segundo a Amazon (2018), “O Amazon Lex é um serviço para a criação de interfaces de conversa em qualquer aplicação, usando voz e texto.”. A plataforma disponibiliza funcionalidades avançadas de aprendizado profundo de ASR (Automatic

speech recognition - reconhecimento automático de fala), para a conversão de fala em texto, e NLU (*Natural language understanding* - entendimento da linguagem natural), para o reconhecimento da intenção do texto (AMAZON, 2018).

Amazon (2018) afirma que:

> O reconhecimento de fala e a compreensão de linguagem natural são dois dos problemas mais difíceis de solucionar na ciência da computação, exigindo que algoritmos sofisticados de aprendizado profundo sejam treinados, usando quantidades massivas de dados e infraestrutura. O Amazon Lex democratiza essas tecnologias de aprendizado ao disponibilizar a própria capacidade para todos os desenvolvedores. Ao usufruir dessas tecnologias, o Amazon Lex permite que você defina categorias de produtos totalmente novas, o que é viabilizado por meio de interfaces de conversa.

Conforme Amazon (2018), essa plataforma possui alguns casos de uso: bots de call center, de informações, aplicações, de produtividade corporativa e bots de internet das coisas.

Os *bots de call center* permitem que os chamadores executem tarefas simples como alteração de senha, solicitação de saldo de contas ou agendamento de compromissos sem necessidade de falar com um agente. Usando o reconhecimento automático de fala e compreensão de linguagem natural para identificar a intenção do chamador, os bots reconhecem a fala humana e identificam a intenção do chamador sem que ele precise falar frases específicas. (AMAZON, 2018).

Segundo Amazon (2018), os bots de informações são:

> [..] chatbots para as solicitações comuns dos consumidores, como acessar as mais recentes notícias, pontuação de jogos ou informações sobre o tempo. Depois de criar um bot do Amazon Lex, você poderá implantá-lo em dispositivos móveis, serviços de bate-papo e dispositivos de IoT com recursos avançados de formatação de mensagens.

Amazon (2018) afirma que os bots de aplicações são interfaces de chat por voz ou texto criados em dispositivos móveis que podem ajudar os usuários a executar várias tarefas básicas, como acessar a conta bancária, comprar alimentos, chamar um táxi ou reservar entradas. Os bots de produtividade corporativa são chatbots que otimizam atividades de trabalho comuns e aumentam eficiências organizacionais (AMAZON, 2018).

Para Amazon (2018):

> O Amazon Lex permite criar experiências de usuário altamente interativas e conversacionais para dispositivos conectados ao segmento da Internet das Coisas (IoT), em franca expansão. Isso cria oportunidades para categorias totalmente novas de produtos conversacionais em diversos mercados, de carros e dispositivos a *wearables* e eletrodomésticos.

Através da plataforma Amazon Lex é possível criar chatbots que interagem com dispositivos IOT, ou seja, eles serão capazes de controlar ações que envolvam esses dispositivos, automatizando ainda mais a vida das pessoas.

2.5.5 Python + Biblioteca Telethon

A Python é um sistema de linguagem para programação de alto nível, interpretada de *script*, orientada a objetos, imperativa, forte, funcional e de tripagem dinâmica. É formado por um modelo de desenvolvimento comunitário, aberto e gerenciado pela Python Software Foundation. Mesmo com diversas partes da linguagem apresentarem especificações e padrões formais, a linguagem, por completo, não é especificada de modo formal. O padrão de fato é a implementação CPython (VENNERS, 2003).

Frente as suas características, é aplicado, principalmente, no processamento de textos, criação de CGIs para páginas dinâmicas para a web e dados científicos. De acordo com uma pesquisa realizada pelo site Stack Overflow em 2018, foi considerada a terceira linguagem mais preferida pelos entrevistados e está entre as cinco linguagens mais utilizadas (O'GRADY, 2017; OVERVIEW, 2018).

Telethon é uma biblioteca Python 3 MTProto assíncrona para interagir com a API do Telegram como um usuário ou por meio de uma conta de bot (alternativa de API de bot) (PYPI, 2019).

2.5.6 ChatRace

O ChatRace é uma solução para criação de agentes conversacionais com multiplataforma. O software oferece um sistema com IA e facilidade na criação de diálogos através de fluxos conversacionais. É uma plataforma paga onde o valor é variável de acordo com o número de usuários que são atendidos. Apresenta um sistema simplificado de uso prático e bem consistente, possibilitando uma variedade de integrações com outros SAAS.

2.6 TELEGRAM

O Telegram é um aplicativo que garante a troca de mensagens entre usuários. Atua de modo sememlhante aos demais apliativos com o mesmo objetivo, permitindo o envio e recebimento de textos, áudios, vídeos, imagens e arquivos por meio de um pacote de dados ou de uma conexão WI-FI. Ademais, o Telegram possui recursos específicos relacionados a privacidade e segurança das mensagens. Um deles é a possibilidade de iniciar conversas secretas com qualquer outro usuário que está na sua lista de contato, nessa opção todas as mensagens trocadas são excluídas definitivamente após um determinado tempo, que pode ser definido por quem criou o chat (CIRIACO, 2015).

2.7 WHATSAPP

O WhatsApp é uma ferramenta para troca de mensagens entre usuários. Foi responsável por redefinir o modo como as pessoas se comunicam. O aplicativo possibilita a troca de mensagens via textos, aúdios, aquivos, alpém de realizar chamadas de aúdio ou vídeo por meio de acesso a intenet. O aplicativo já ultrapassa dois bilhões de usuários por todo o mundo (COSTA, 2021).

3 METODOLOGIA

3.1 CARACTERIZAÇÃO DA PESQUISA

Essa pesquisa teve como foco a análise das tecnologias empregadas ao desenvolvimento de chatbot. Assim, foram desenvolvidas três ferramentas de conversação via chatbot. A primeira fez uso de processamento de linguagem natural para Telegram com auxílio do Python. A segunda utilizou processamento de linguagem natural para Whatsapp usando DialogFlow. A terceira, utilizou o processamento de linguagem natural para Whatsapp usando ChatRace. Por fim, apresentou-se uma análise dos processos de desenvolvimento das ferramentas apresentadas, destacando as principais características vantagens, desvantagens de cada e, qual se mostra mais viável para a empresa em questão.

Quanto a abordagem a pesquisa é qualitativa. No que tange a natureza se caracteriza como aplicada, buscando gerar conhecimentos para aplicação prática. Quanto aos objetivos, a pesquisa é explicativa pois se preocupa em identificar os fatores determinantes ou contribuintes com ocorrência de fenômenos. Quanto aos procedimentos é experimental. A pesquisa foi ainda suportada por uma revisão de literatura consistente.

3.2 ETAPAS METODOLÓGICAS

- Revisão bibliográfica para desenvolvimento de fundamentação teórica;
- Desenvolvimento de chatbot para Telegram+ Python;
- Desenvolvimento de chatbot para Whatsapp+DialogFlow.
- Desenvolvimento de chatbot para Whatsapp+ChatRace.

3.3 CARACTERIZAÇÃO DA EMPRESA

O bot desenvolvido visou atender as demandas de uma microempresa do segmento de informática e PC Gamer, que atua a três anos tanto no comércio físico como digital.

3.4 IDENTIFICAÇÃO DA DEMANDA

A empresa em questão possui vários contatos telefônicos distintos, atendendo a cada setor; como suporte, vendas, serviços, administrativo entre outros; a fim de atender aos seus clientes e fornecedores. Por meio desses contatos os clientes são atendidos por colaboradores, tornando assim o atendimento mais pessoal.

No entanto, frente ao crescimento do negócio houve a necessidade de centralizar os atendimentos em um único contato telefônico e posteriormente fazer as divisões por setores de interesse.

Assim, esse contato passou a ser o número central divulgado nas mídias sociais, sites, anúncios, outdoors, dentre outros.

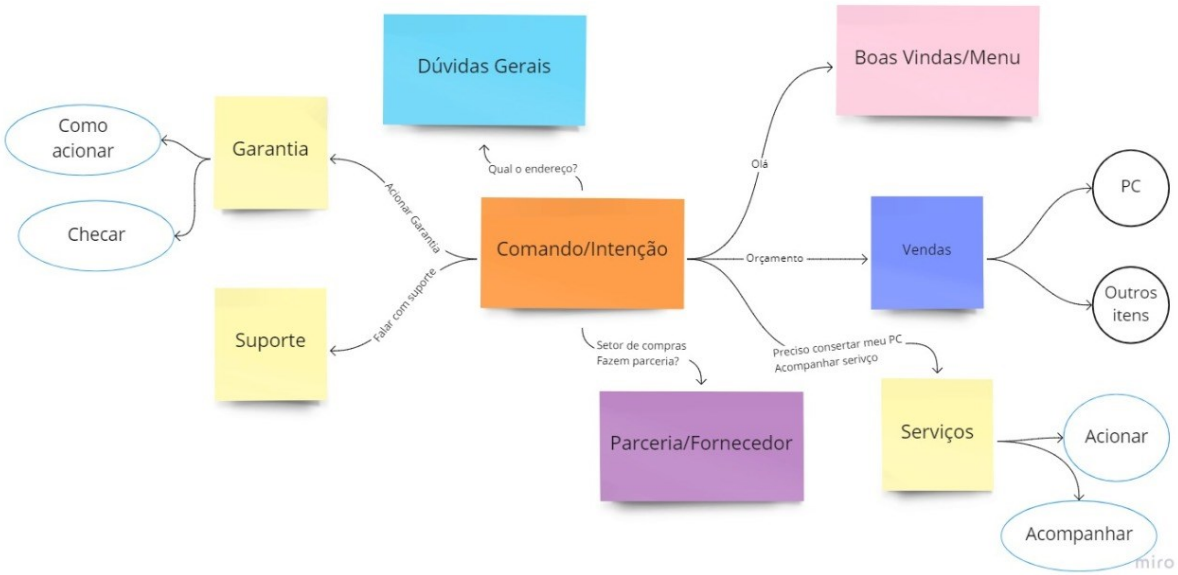
A priori pensou-se em aplicar algum *software* para unificar o atendimento em um só contato, no entanto esses *softwares* apresentam limitações em seu uso: impossibilidade de ligações, não sendo bom quanto ao nativo e, impossibilidade de uso em qualquer lugar onde o colaborador esteja.

Nesse contexto, precisou-se buscar uma nova ferramenta par unificar os atendimentos, assim a criação do bot foi a estratégia escolhida. Por meio do bot esperava-se sanar algumas questões:

- Identificar de maneira facilitada e direta a necessidade do cliente respondendo ou encaminhando diretamente para o setor responsável;
- Não será mais necessário que um atendente/recepcionista tome conta dessa responsabilidade;
- Otimizar tempos de resposta;
- Escalabilidade do processo;
- Divisão mais justa dentro da equipe de vendas (o bot alternaria entre os vendedores).

O *mindmap* ilustrado na Figura 4 apresenta as etapas para desenvolvimento dos chatbot.

Figura 4 – *Mindmap* do processo



Fonte: Autor, 2021

3.5 DESENVOLVIMENTO DOS PROJETOS

O bot desenvolvido, ao receber uma mensagem de saudação do cliente retorna com uma mensagem automática de boas-vindas/recepção visando identificar a necessidade do cliente, apresentando opções dos tópicos mais requisitados, facilitando o andamento do atendimento. Ademais, o bot é capaz de entender outras palavras-chave pré-definidas que podem ser respondidas pelo usuário, tornando o atendimento mais humanizado possível, capaz de fazer um diálogo que não esteja exatamente no *script.*

Rapidamente do chatbot pode identificar a necessidade do cliente e resolver seu problema por meio de um passo a passo eficiente, ou quando necessário entender a necessidade de já encaminhar o cliente para um atendente humano.

As opções iniciais definidas para atendimento foram: Orçamento/Venda; Acionar Garantia; Solicitar Serviço Suporte; Parcerias e Fornecedores; outras dúvidas.

3.5.1 Elaboração das APIs

Para melhor entendimento cronológico do projeto, a primeira seção abordará sobre a criação das APIS. Estas foram utilizadas para três propósitos.

1. Integração do TelegramBot com o GooglePlanilhas e outras requisições;
2. Integração do DialogFlow com a Twilio para funcionamento do WhatsApp;
3. Integração do DialogFlow com GooglePlanilhas.

As bibliotecas utilizadas para o desenvolvimento do projeto foram: gspread, oauth2client, flask, requests, twilio, datetime, google-cloud-dialoglow e python-dotenv. Foi utilizado o VisualStudio como IDE.

As rotas de API's foram criadas usando o Google Cloud Plataform conforme Figura 5.

Figura 5 - API

Fonte: Autor, 2021

Com as APIs criadas é gerado um JSON que deve ser importado ao projeto. Estão identificadas como ‘creds.sheets.json’ e ‘creds.twilio.json’.

Quanto a API com o Telegram, foram criadas rotas dentro do projeto. Uma rota para a comunicação com o GooglePlanilhas e outra rota confirma que o cliente já está em atendimento. Em outras palavras, entrou em algum fluxo conversacional para que não confunda os índices durante o atendimento.

Figura 6 – Trecho do arquivo de código para API com Telegram

```python
@APP_WEBHOOK.route('/telegram/on/<id>', methods=['POST'])
def telegramNew(id):

    if not id:
        return Jsonify({"message": "Missing arguments"}), 400

    sheet = Sheets()
    response = sheet.onTelegram(id)
    if "not found" in response:
        return Jsonify({"message": response}), 404

    return Jsonify({"message": response}), 200
```

Fonte: Autor, 2021

O processo da API integração DialogFlow com Twilio está demostrado na Figura 7.

Figura 7 - Código da API com a TWILIO

```python
# Rota especifica para receber valores da twilio service
@APP_WEBHOOK.route('/twilio', methods=['POST'])
def twilio():
    # Extraindo mensagem de texto do whatsApp
    message = Request.form.get('Body')
    # Extraindo numero da mensagem
    phone_number = Request.form.get('From')
    # instanciando a class dialogFlow para gerar as configuracoes necessarias
    dialog = DialogFlow()
    # Buscando a resposta da mensagem com a class dialogFlow
    response = dialog.FetchReply(message, phone_number)
    # instanciando a class MessagingResponse da twilio lib responsavel por responder :D
    _response = MessagingResponse()

    for fragment in response:
        _response.message(str(fragment))

    return Response(str(_response), status=200)
```

Fonte: Autor, 2021

Em dialogflow.py é também encontrado toda a parte lógica da comunicação entre Twilio e DialogFlow conforme a Figura 8.

Figura 8 – Trecho do arquivo de código dialogflow.py

```
class DialogFlow:
    def __init__(self):
        os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "src/config/creds.twilio.json"
        self.session_client = Dialogflow.SessionsClient()
        self.PROJECT_ID = dotenv_values()["PROJECT_ID"]

    def DetectIntent(self, text, session_id, language_code='pt-br'):
        session = self.session_client.session_path(self.PROJECT_ID, session_id)
        text_input = Dialogflow.types.TextInput(
            text=text, language_code=language_code)
        query_input = Dialogflow.types.QueryInput(text=text_input)
        _response = self.session_client.detect_intent(
            session=session, query_input=query_input).query_result.fulfillment_messages
        response = []
```

Fonte: Autor, 2021

No caso da API DialogFlow com GoogleSheets, essa ferramenta é necessária para que algumas funcionalidades dentro do diálogo leem e escrevem nas planilhas do Google. Parte do código da API com o Google sheets ilustrado na Figura 9.

Figura 9 - Código da API com o Google sheets

```
@APP_WEBHOOK.route('/webhook', methods=['GET', 'POST'])
def webhook():
    req = Request.get_json(silent=True, force=True)
    response = ""
    # Invocando a class Sheets que e responsavel pelos metodos de contato com as planilhas
    sheet = Sheets()

    # caso nao tenha um body request, respondera um erro
    if not req:
        return Jsonify(UseSend("Faltando Argumentos")), 400

    # Match ou Switch e um metodo que esta com a responsabilidade de localizar e extrair os valores de cada webhook pre determinado aqui.
    match req["queryResult"]["intent"]["displayName"]:
        case "Servicos - Acompanhar":
            params = req["queryResult"]["parameters"]
            response = sheet.servicoStatus(ordem=params["ordem"])
        case "Suporte":
            params = req["queryResult"]["parameters"]
            response = sheet.newSuporte(
                nome=params["nome"], celular=params["telefone"], problema=params["duvida"])
        case "Parceria":
            params = req["queryResult"]["parameters"]
            response = sheet.newParceria(
                nome=params["nome"], celular=params["telefone"], info=params["motivo"])
        case "Fornecedor":
```

Fonte: Autor, 2021

Em sheets.py é também encontrado toda a parte lógica de leitura e escrita das planilhas (Figura 10).

Figura 10 – Trecho do arquivo de código sheets.py

```python
class Sheets:
    def __init__(self):
        self.scope = ["https://spreadsheets.google.com/feeds", 'https://www.googleapis.com/auth/spreadsheets',
                      "https://www.googleapis.com/auth/drive.file", "https://www.googleapis.com/auth/drive"]
        self.creds = ServiceAccountCredentials.from_json_keyfile_name(
            "src/config/creds.sheets.json", self.scope)
        self.client = gspread.authorize(self.creds)
        self.sheet = self.client.open("Tibuff - Sistema")

    def findPromocao(self, search: str = ""):
        sheet = self.sheet.get_worksheet(0)
        extractorInformations = sheet.get_all_records(
            numericise_ignore=['all'])
        listProducts = []
        customMessage = '''Veja o(s) produto(s):'''

        if search:
            for extractorInformation in extractorInformations:
                expression_0 = search.lower(
                ) in extractorInformation["NOME DO PRODUTO"].lower()
                expression_1 = search.lower(
                ) in extractorInformation["DESCRIÇÃO"].lower()
                expression_2 = search.lower(
                ) in extractorInformation["MODELO"].lower()

                if expression_0 or expression_1 or expression_2:
                    listProducts.append(extractorInformation)
```

Fonte: Autor, 2021

Outros arquivos de código para criação da APIs foram necessários, o Utils.py, arquivo responsável por gerar o texto da response e também reportar o valor esperado no DialogFlow webook.

Figura 11 – Trecho do código utils.py

```python
# Funcao que reporta o valor de response esperado no dialogFlow webhook
def UseSend(response: list | str, lastMessage: bool = True):
    response_send = []

    # Aqui estamos validando o tipo da response, para assim adicionar varias mensagens caso preciso
    match type(response).__name__:
        case "str":
            # Caso string adicionamos na resposta :D
            response_send.append(generateTextResponse(
                response))
        case "list":
            # Utilizando o metodo for in para interar sobre o array e assim extrair as mensagens
            for part in response:
                response_send.append(generateTextResponse(
                    part))
```

Fonte: Autor, 2021

Com as APIs concluídas pode ser então apresentado os passos para o desenvolvimento de cada um dos projetos.

3.5.2 Projeto Python/Telethon (bot para Telegram)

A priori, foi realizado estudo de outros projetos que fizeram chatbots pelo telegram.

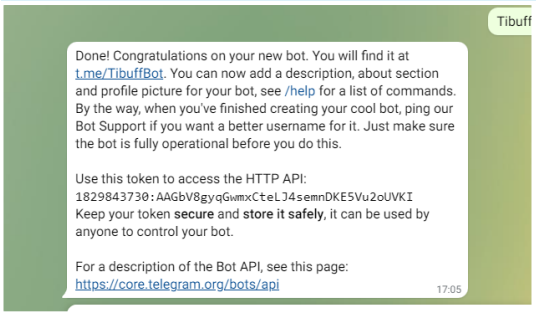
Seguiu-se com a definição de uma biblioteca que melhor atenderia as demandas do sistema. Assim optou-se pela Telethon frente aos seguintes motivos:

- A documentação é grande e completa, caso houvesse qualquer dúvida de implementação era só consultar a biblioteca;
- Sistema com bastante atualização, ou seja, quando o telegram atualizava algo, a Telethon já fazia o lançamento. Em relação ao código, consiste em uma biblioteca open-source;
- É possível visualizar o módulo de enviar mensagem dentro da biblioteca, como ele é de fato construído;
- Por ser uma biblioteca bem renomada, no código tem boas práticas, comunidade ativa e bom suporte.
- Suporte com Python 3.9 (Última versão no momento do inicio do projeto)

Foi utilizado o Modelo Arquitetônico Model View Controler (MVC), que é baseado em módulos, views e um controlador. O MVC é uma ótima opção para projetos escaláveis, como é o caso do chatbot para desenvolvimento do projeto.

Seguiu-se com a criação do bot pelo @BotFather do Telegram e ainda configurações pelo https://my.telegram.org/apps a fim de gerar um token, hash, id que fizeram a autenticação do bot dentro do código, conforme ilustrado na Figura 12.

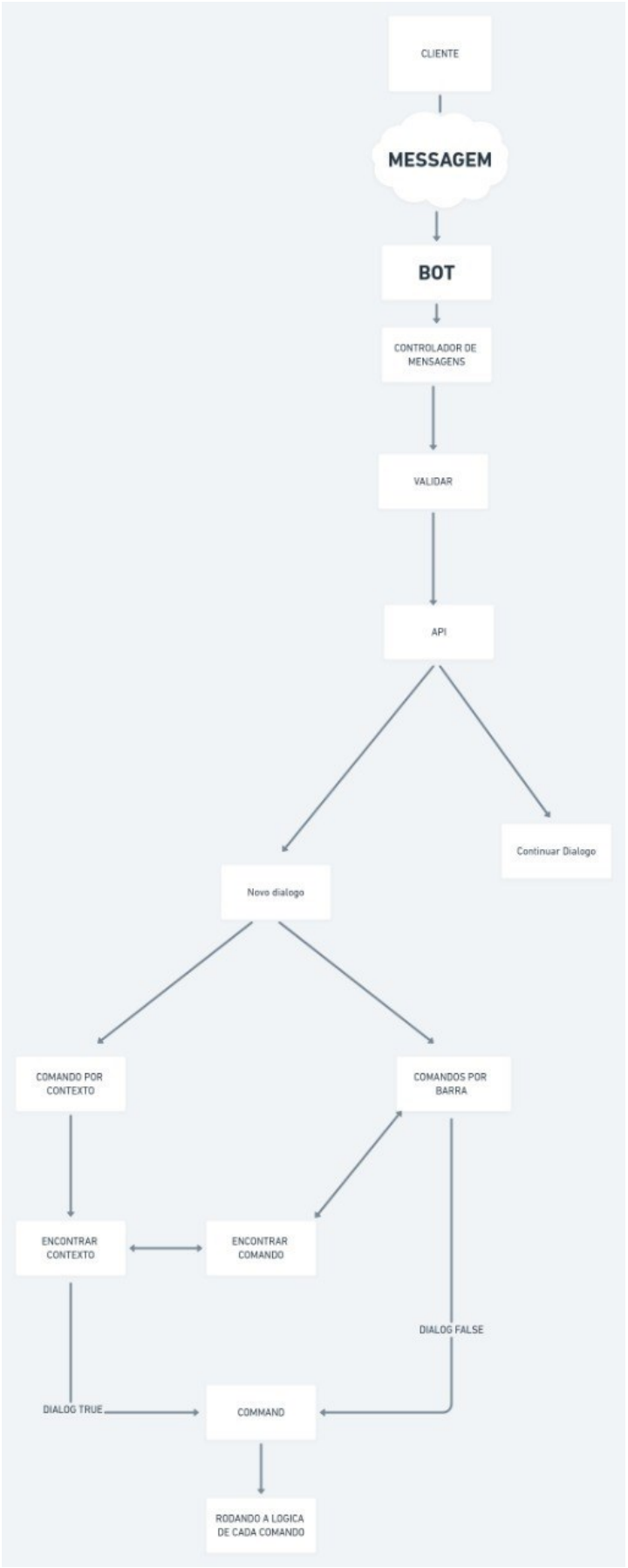
Figura 12 - Criação do BOT pelo BotFather do Telegram



Fonte: Autor, 2021

Diagrama onde se encontra o fluxo da lógica do projeto. Pode-se identificar os processos do controlador e dos módulos do . Ao receber uma nova mensagem e caso o usuário não esteja em um atendimento é então iniciado, porém se o usuário já estiver em um atendimento é executado a lógica do comando já iniciado (Figura 13).

Figura 13 – Código Telegram



Fonte: Autor, 2021

Seguiu-se com a escolha do software VisualStudio como IDE. No que se refere

ao código, as seguinte etapas foram executadas: bibliotecas necessárias; desenvolver o controlador; criação dos módulos; views; env; Init.py; outros arquivos existentes.

a) Bibliotecas necessárias

- telethon (parte de conexão com o telegram de uma maneira mais fácil);
- python-dotenv (para conseguir ler o arquivo.env);
- requests (requisição de terceiros ou serviço privado);
- unidecode (extrair caracteres especiais);
- asyncio (para fazer código assincrono).

b) Controlador (main.py)

Parte do código onde se encontra função de inicialização, execução e todo o controlador (responsável para 'ouvir' os comandos') foi utilizada uma API para os comandos funcionarem em harmonia (Figura 15).

Figura 14 - Trecho do arquivo de código main.py

```
class TibuffBot:
    def __init__(self):
        try:
            print("Configurando...")
            if BOT == None or BOT_TOKEN == None:
                return print("É preciso configurar o bot!")

            BOT.start(bot_token=BOT_TOKEN)
            print("Configurado")
        except:
            return print("Houve um erro, tente rodar novamente, se o erro persistir entre em contato com comercial@tibuff.com.br,

    def run(self):
        self.inicializeChat()
        BOT.run_until_disconnected()

    def inicializeChat(self):
        print("Inicicializado Sistema de obervação de chat")
```

Fonte: Autor, 2021

Na Figura 15 pode-se identificar o processo de inicialização do BOT com o Telegram e também o inicio da função que recebe as mensagens dos usuários via chat.

c) Modules

Há 3 arquivos de código de módulos.

- modulo de comando (responsável por incluir novos comandos/intenções ao BOT.) Na Figura 15 pode-se identificar 3 comandos, o de inicialização, o de boas vindas (start) e sobre orçamento (pc). Estes comandos são chamados dentro do módulo de checagem ao confirmar se existir um contexto, comando por barra ou índice que os representem (armazenados em um arquivo .json). Cada comando tem a sua lógica estabelecida dependendo da complexidade. Alguns comandos são mais simples como horário, localização, boas vindas, já outros requerem de mais informações do usuário para prosseguir a lógica.

Figura 15 - Trecho do arquivo de código command_module.py

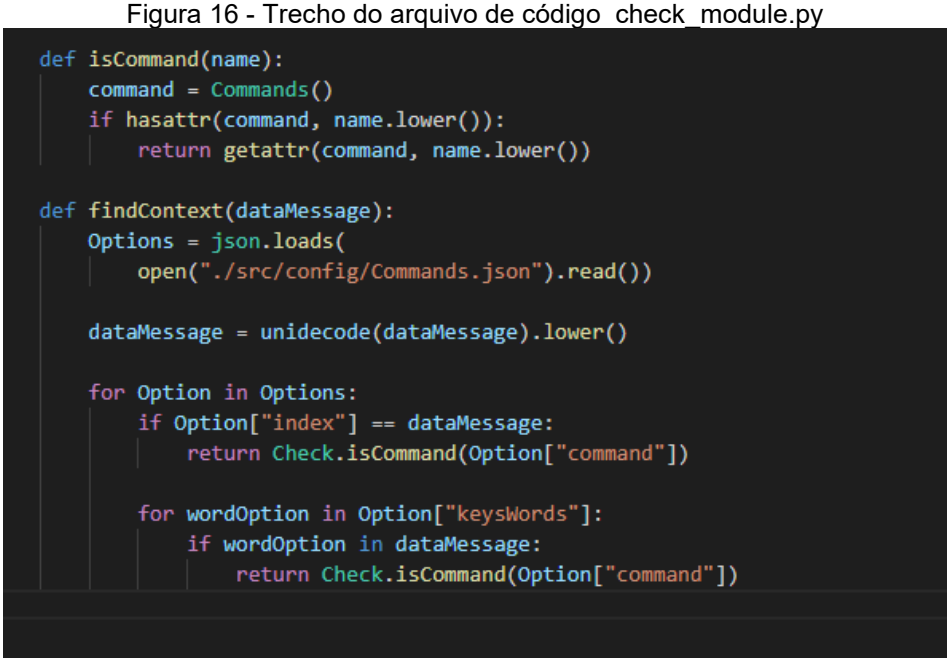
```
class Commands:
    def __init__(self):
        self.chat = chat_module.Chat
        self.findFile = findFile
        self.servicosList = json.loads(open("src/config/Servicos.json").read())

    # 0
    async def start(self, data):
        await BOT.send_message(data.id, self.findFile("start", "instruction"))
        return await BOT.send_message(data.id, self.findFile("cancelar", "instruction"))

    # 1
    async def pc(self, data):
        await BOT.send_message(data.id, self.findFile("pc", "instruction"))
        await BOT.send_message(data.id, self.findFile("cancelar", "instruction"))
        response_client = await self.chat.waitNewMessageClient(data.id)
        if not response_client:
            return await BOT.send_message(data.id, self.findFile("recomecar", "sucess"))

        if await self.chat.sendSupportMessage(data, response_client, "SUPPORT_CHANNEL_VENDAS"):
            return await BOT.send_message(data.id, self.findFile("pc", "sucess"))
```

Fonte: Autor, 2021

- módulo de checagem (responsável por identificar se as mensagens recebidas são iguais aos comandos programados.)

Responsável por identificar se a mensagem informada é um comando ou não e também por pausar a reprodução da lógica dentro de um comando, fazendo com que o usuário possa entrar em um novo diálogo (Figura 16).

Figura 16 - Trecho do arquivo de código check_module.py



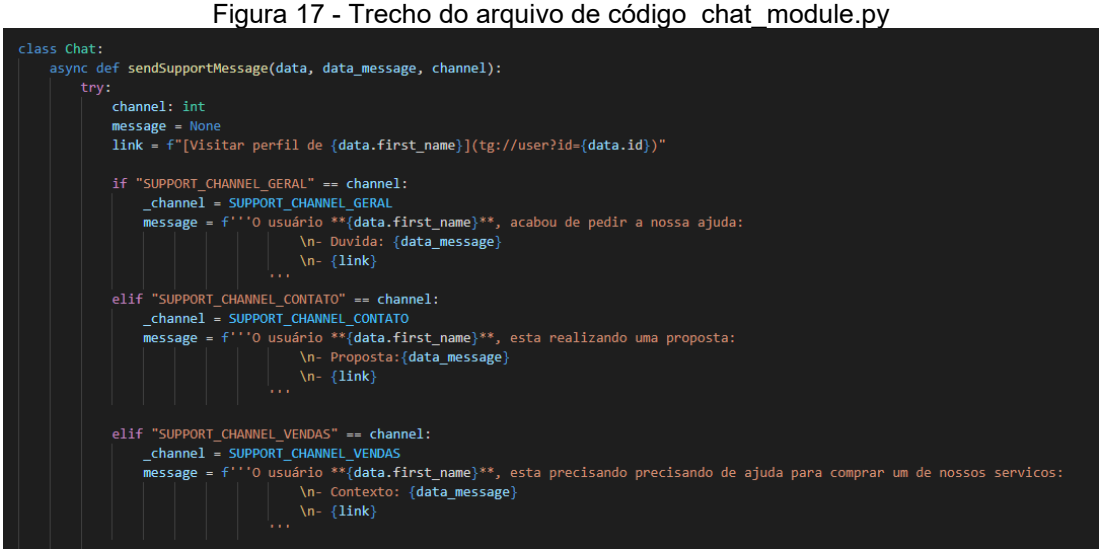
```
def isCommand(name):
    command = Commands()
    if hasattr(command, name.lower()):
        return getattr(command, name.lower())

def findContext(dataMessage):
    Options = json.loads(
        open("./src/config/Commands.json").read())

    dataMessage = unidecode(dataMessage).lower()

    for Option in Options:
        if Option["index"] == dataMessage:
            return Check.isCommand(Option["command"])

        for wordOption in Option["keysWords"]:
            if wordOption in dataMessage:
                return Check.isCommand(Option["command"])
```

Fonte: Autor, 2021

- módulo de chat (responsável por enviar mensagens para suporte e fazer requests com a API.)

Responsável por manter o diálogo ativo quando iniciado, notificar via API que o usuário está em atendimento, esperar novas mensagens dentro dos comandos, enviar mensagens de suporte para membros e por fim notificar via API que o usuário saiu do atendimento (Figura 17).

Figura 17 - Trecho do arquivo de código chat_module.py



```
class Chat:
    async def sendSupportMessage(data, data_message, channel):
        try:
            channel: int
            message = None
            link = f"[Visitar perfil de {data.first_name}](tg://user?id={data.id})"

            if "SUPPORT_CHANNEL_GERAL" == channel:
                _channel = SUPPORT_CHANNEL_GERAL
                message = f'''O usuário **{data.first_name}**, acabou de pedir a nossa ajuda:
                                \n- Duvida: {data_message}
                                \n- {link}
                            '''
            elif "SUPPORT_CHANNEL_CONTATO" == channel:
                _channel = SUPPORT_CHANNEL_CONTATO
                message = f'''O usuário **{data.first_name}**, esta realizando uma proposta:
                                \n- Proposta:{data_message}
                                \n- {link}
                            '''

            elif "SUPPORT_CHANNEL_VENDAS" == channel:
                _channel = SUPPORT_CHANNEL_VENDAS
                message = f'''O usuário **{data.first_name}**, esta precisando precisando de ajuda para comprar um de nossos servicos:
                                \n- Contexto: {data_message}
                                \n- {link}
                            '''
```

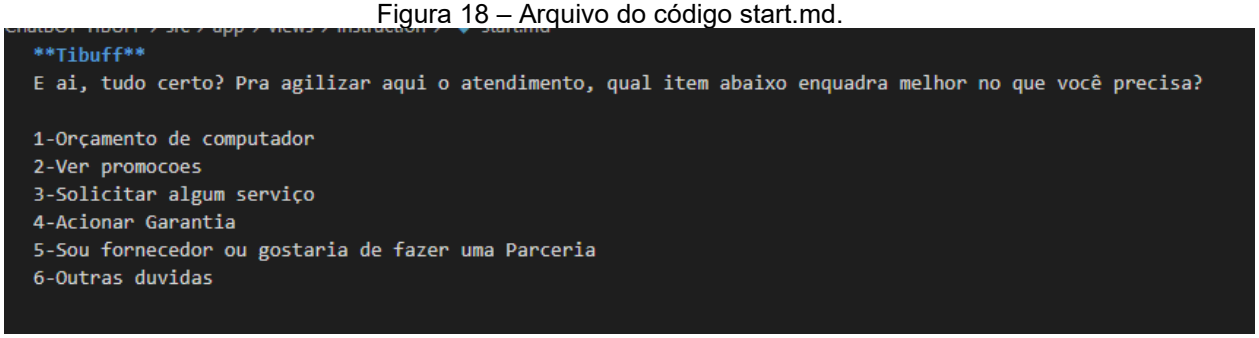
Fonte: Autor, 2021

d) Views

Pasta do código responsável por toda a parte que será visível no diálogo com o cliente. Pode ser facilmente alterada a qualquer momento.

Cada diálogo é representado por um arquivo .md (markdown).

Figura 18 – Arquivo do código start.md.



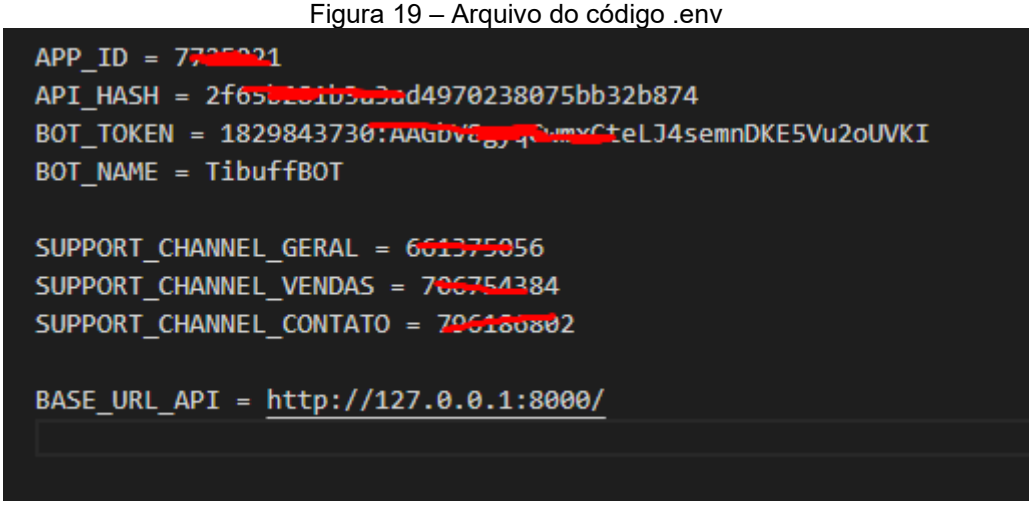
Fonte: Autor, 2021

e) Env

É uma ferramenta da biblioteca dotenv utilizada para orquestrar variáveis ambiente de um projeto.

Nesse projeto serão adicionadas no .env as credenciais PRIVADAS para conexão, como API, HASH, ID, NÚMERO SUPORTE, BOT_NAME, entre outros.

Figura 19 – Arquivo do código .env



Fonte: Autor, 2021

f) Init.py

Código responsável para configurar as credenciais inseridas no .env, e fazer a

comunicação com o Telegram e API Externa.

Figura 20 – Arquivo do código __init__.py

```python
from dotenv import dotenv_values
from telethon import TelegramClient

# Importante valores do dotenv
APP_ID: int = dotenv_values()["APP_ID"]
API_HASH: str = dotenv_values()["API_HASH"]
BOT_TOKEN: str = dotenv_values()["BOT_TOKEN"]
BOT_NAME: str = dotenv_values()["BOT_NAME"]
SUPPORT_CHANNEL_GERAL: int = dotenv_values()["SUPPORT_CHANNEL_GERAL"]
SUPPORT_CHANNEL_VENDAS: int = dotenv_values()["SUPPORT_CHANNEL_VENDAS"]
SUPPORT_CHANNEL_CONTATO: int = dotenv_values()["SUPPORT_CHANNEL_CONTATO"]

BASE_URL_API: str = dotenv_values()["BASE_URL_API"]

BOT = TelegramClient(BOT_NAME, APP_ID, API_HASH)
BOT.parse_mode = "md"
```

Fonte: Autor, 2021

g) Outros arquivos auxiliares.

O código possui também um módulo helper para auxílio em encontrar os arquivos em views, bem como arquivos em .json para melhor comunicação com dados de comando (Figura 21).

Figura 21 – Arquivo do código findFile_helper.py

```python
import codecs


def findFile(filename: str, folder: str):
    File = codecs.open(
        f"./src/app/views/{folder}/{filename}.md", "r", encoding="utf-8")
    File.readable()
    fileContent = File.read()
    File.close()
    return fileContent
```

Fonte: Autor, 2021

Para o funcionamento do Bot, o mesmo foi executado em servidor local. Recomenda-se que seja feito um deploy em um servidor como o heroku para o pleno desempenho constante do bot.

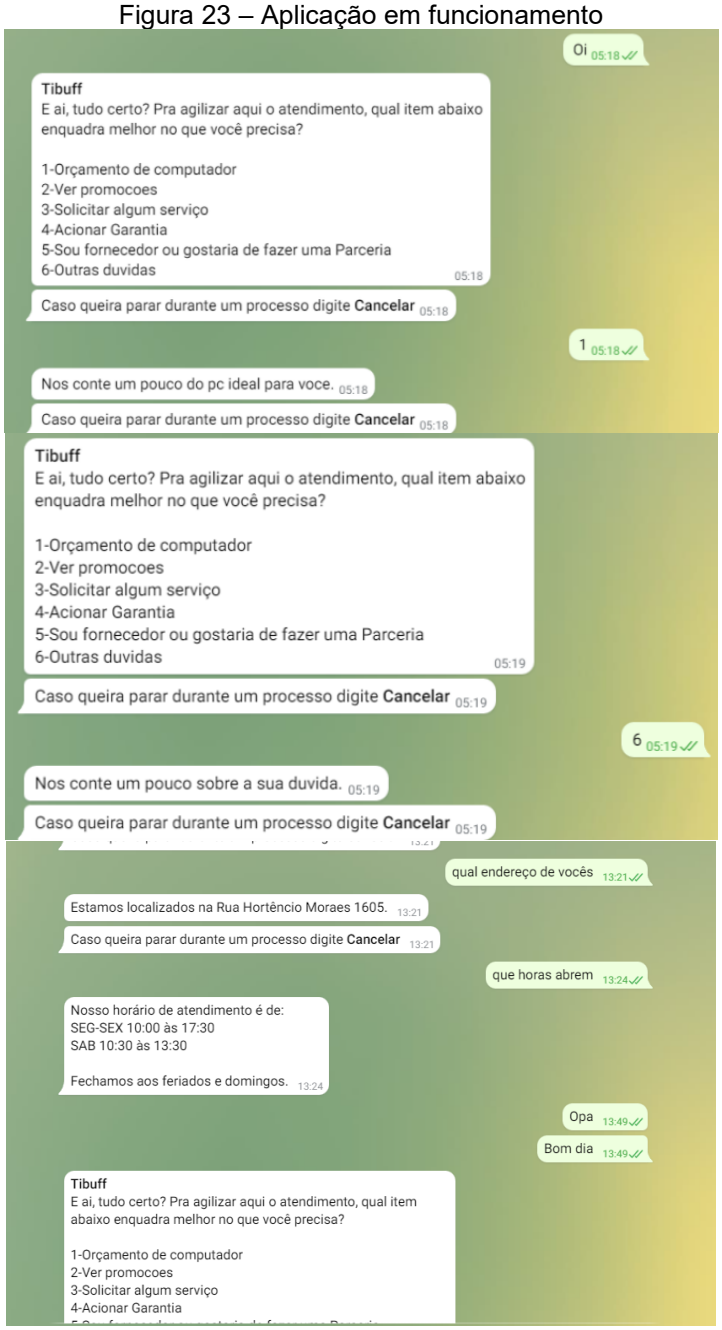
Figura 22 – Trecho do código Commands.json

```
[
  {
    "index": "0",
    "command": "start",
    "keysWords": ["bom dia", "boa tarde", "boa noite", "oi", "ola"]
  },
  {
    "index": "1",
    "command": "pc",
    "keysWords": ["orcamento", "pc", "laptop"]
  },
  {
    "index": "2",
    "command": "promocao",
    "keysWords": ["promocao"]
  },
  {
    "index": "3",
    "command": "servicos",
    "keysWords": ["limpeza", "manutencao", "servico"]
```

Fonte: Autor, 2021

Os resultados obtidos desse primeiro objeto podem ser vistos conforme Figura 23.

Figura 23 – Aplicação em funcionamento

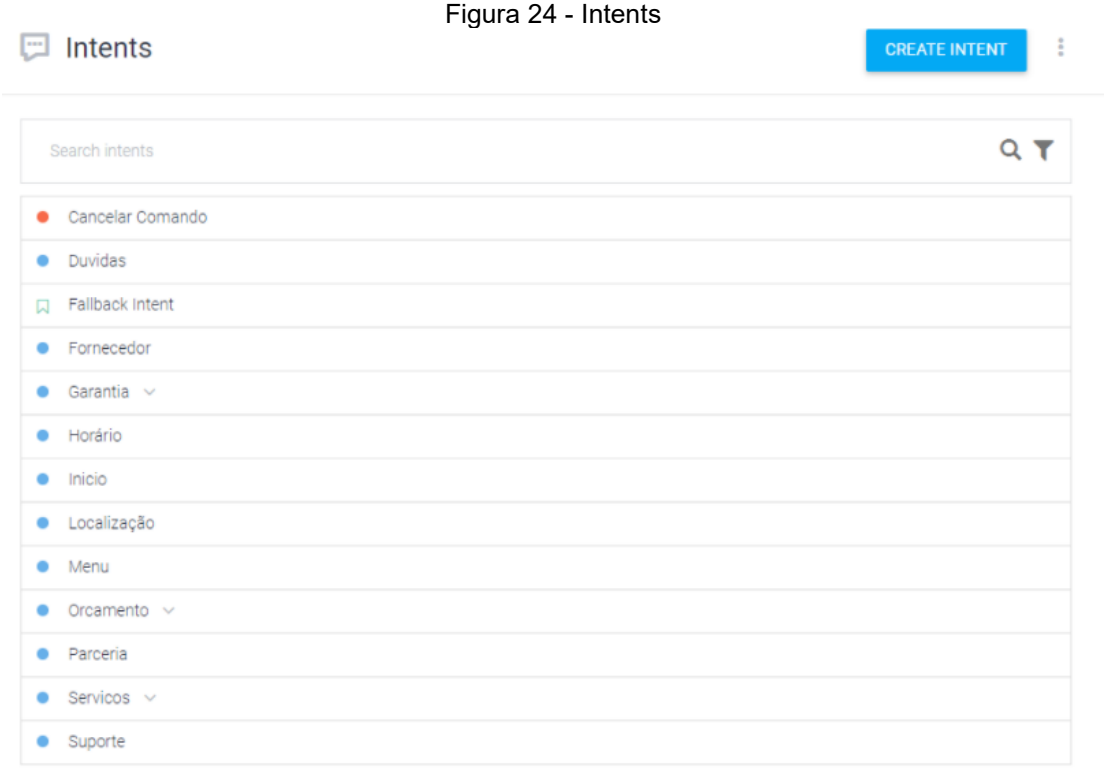


Fonte: Autor, 2021

3.5.3 Projeto DialogFlow/Whatsapp

O processo teve início com a criação de uma conta no DialogFlow, além da criação das intents (intenções) conforme Figura 24. Para a criação das intents basta utilizar o botão ‘CREATE INTENT’.

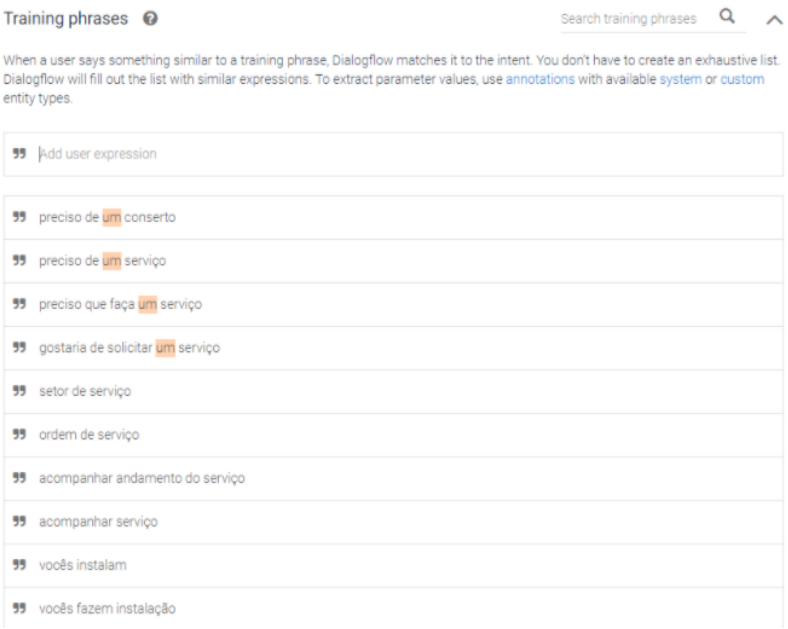
Figura 24 - Intents



Fonte: Autor, 2021

Algumas frases precisaram ser inicialmente treinadas para que acionem os comandos. Com a IA do Google, novas frases não exatamente como as escritas acionam os comandos por serem identificadas como similares, contribuindo bastante para a inteligência do bot (Figura 25).

Figura 25 – Treinamento de frases

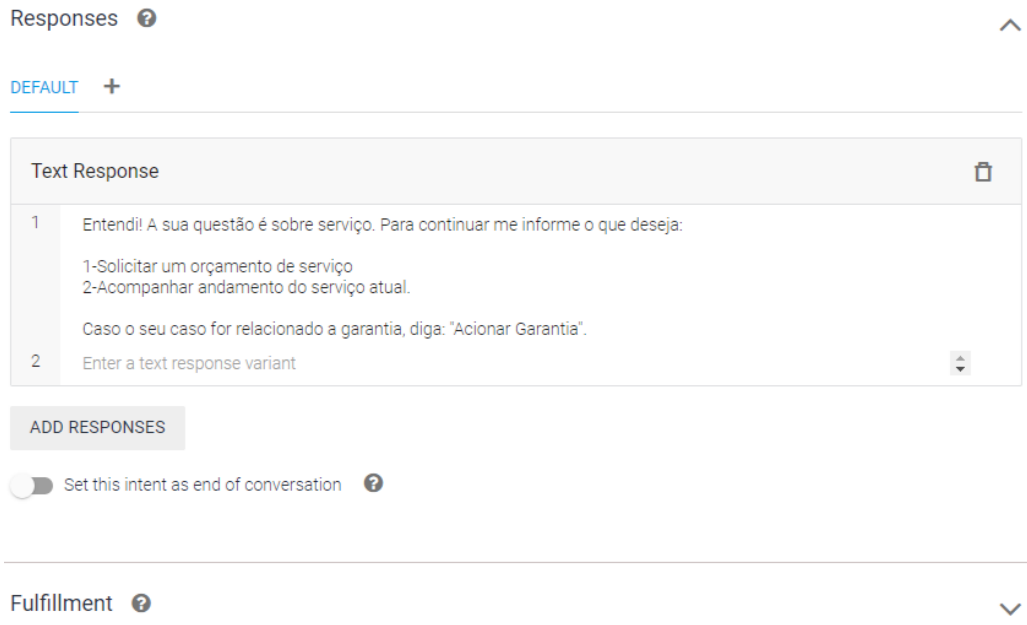


Fonte: Autor, 2021

De acordo com Figura 25, são configuradas então as respostas para cada uma das intenções, em alguns casos são respostas padrões (como por ex: local da loja), informações que não dinâmicas, já outras informações é necessário que façam a ativação do webhook, para que extraía informação via API de um outro serviço. Neste caso foi utilizado em comandos como serviço, parceria, fornecedores...

Seguiu-se com a criação das respostas para cada uma das intenções, conforme Figura 26.

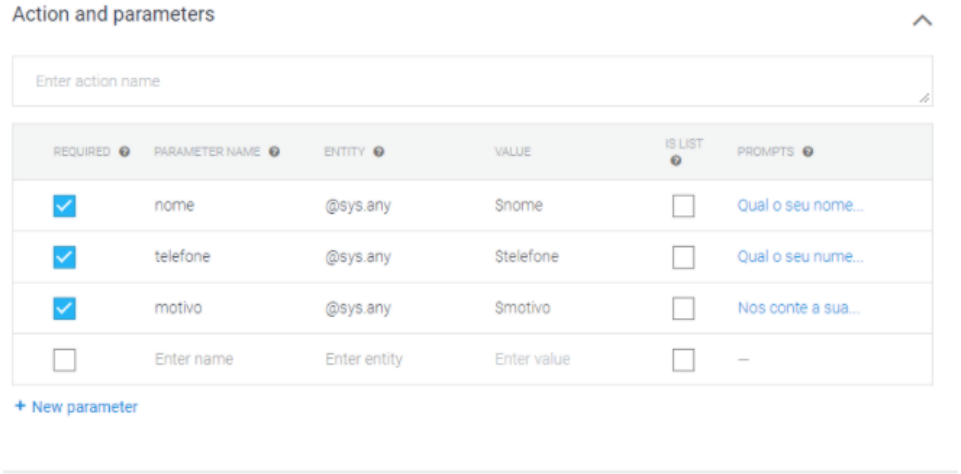
Figura 26 – Criação das intenções



Fonte: Autor, 2021

Para algumas intenções (Parceria, Suporte, Fornecedor) foi necessário também a criação de parametros e ações. Quando o cliente cai nessa intenção, alguns parametros são obrigatórios para serem identificados. Esse paramêtros são identificados pelas entidades (Figura 27).

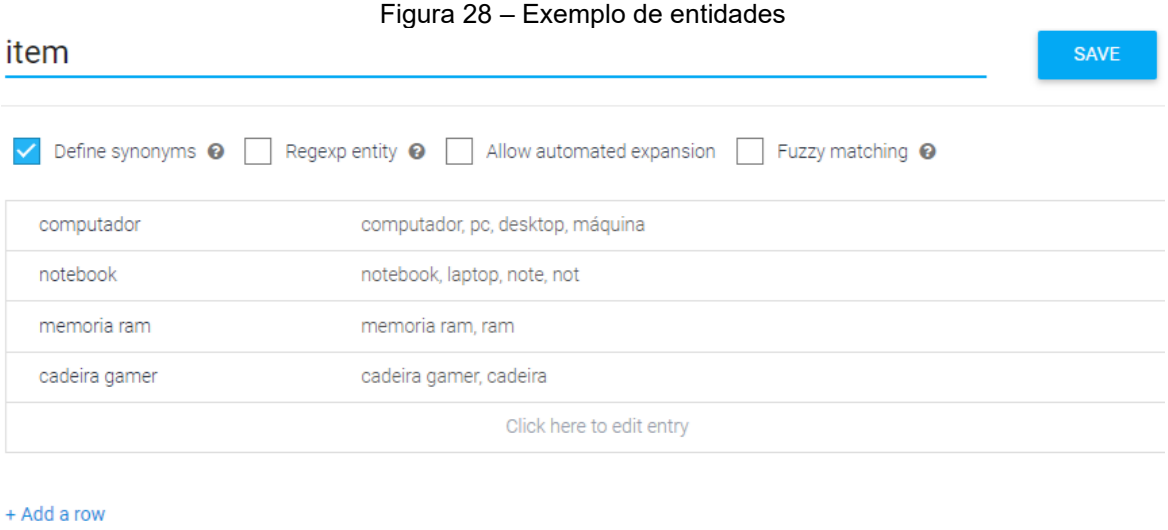
Figura 27 – Criação de parametros e ações



Fonte: Autor, 2021

Um exemplo de criação de entidades está ilustrado na Figura 28.

Figura 28 – Exemplo de entidades



Fonte: Autor, 2021

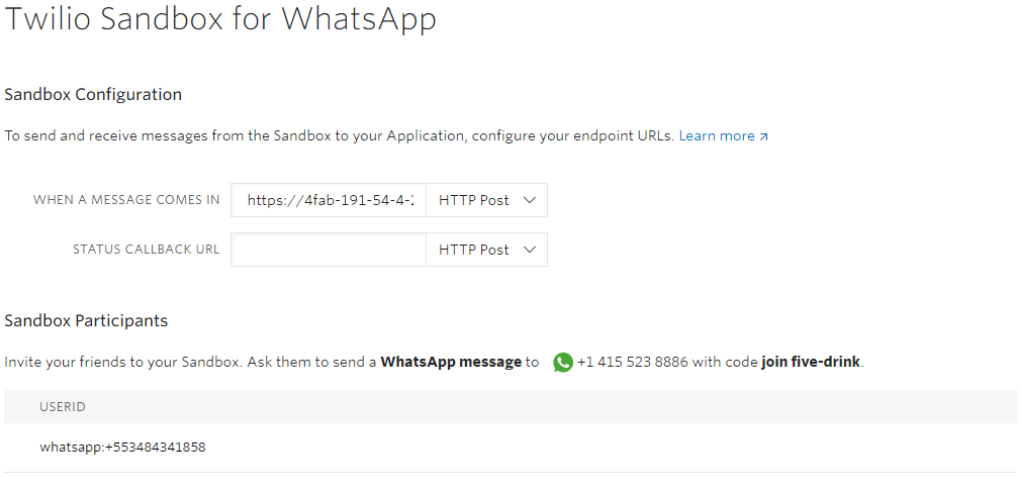
Para a integração com o WhatsApp foi necessário a criação de uma conta da Twilio, empresa que disponibiliza integração via API com WhatsApp.

Em seguida deve-se associar a conta com um número. A API é paga, portanto para o projeto utilizamos o projeto em SandBox (Testes).

Para sincronizar o DialogFlow com a Twilio foi também necessário a criação de API Externa que foi explicada anteriormente.

Na configuração do sandbox deve-se inserir o servidor para fazer as requisições, esse mesmo será explicado na Figura 29.

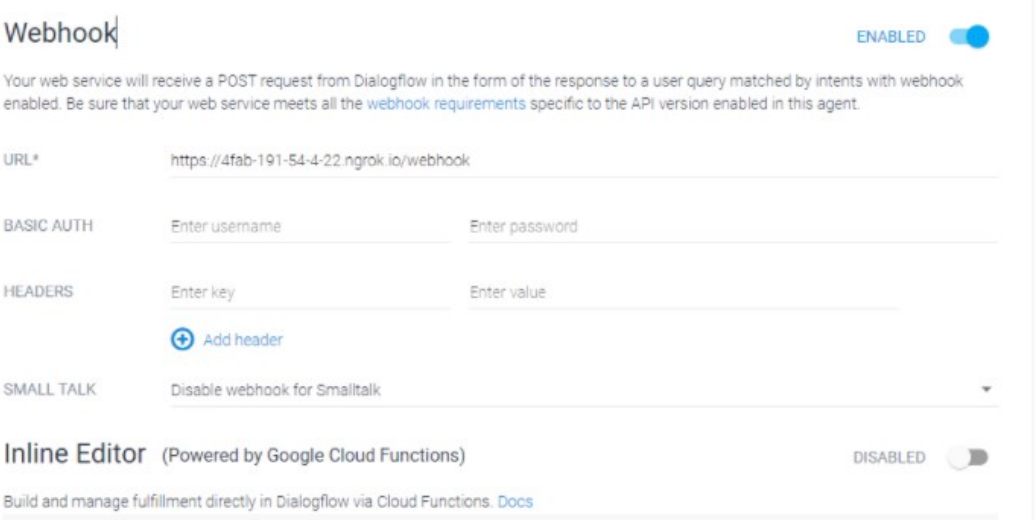
Figura 29 - Configuração sandbox da Twilio



Fonte: Autor, 2021

Foi necessário implementação do fullfiment com webhook para fazer requisição externa ao dialogflow (Figura 30).

Figura 30 - Implementação do fullfiment com webhook



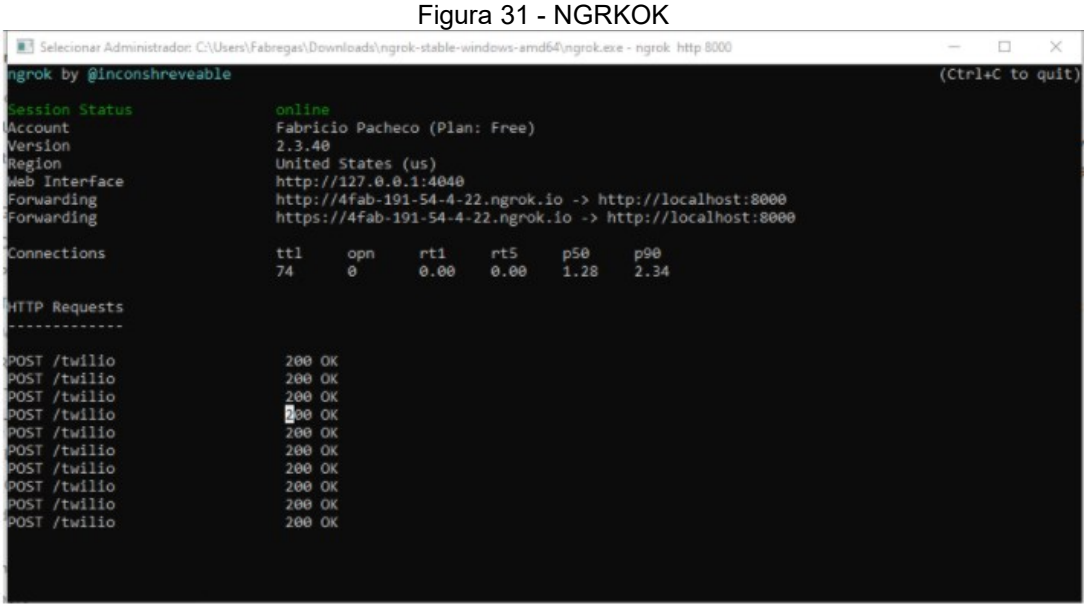
Fonte: Autor, 2021

Em URL é apontado o servidor no webhook do dialogflow, este mesmo endereço deve ser apontado na configuração do sandbox da Twilio. Para o projeto foi utilizado um servidor para Desenvolvimento chamado NGROK. (Recomenda-se para o projeto escalável a utilização de um servidor em nuvem como o HEROKU e fazendo um deploy com o github,

O NGROK é responsável por pegar o servidor local e transformá-lo em um servidor temporário em nuvem. Ao criar uma conta na NGROK é fornecido um token para autenticação e um tutorial a fim de colocar o servidor online.

A Figura 31 demostra o NGRKOK online fazendo as requisições da TWILIO.

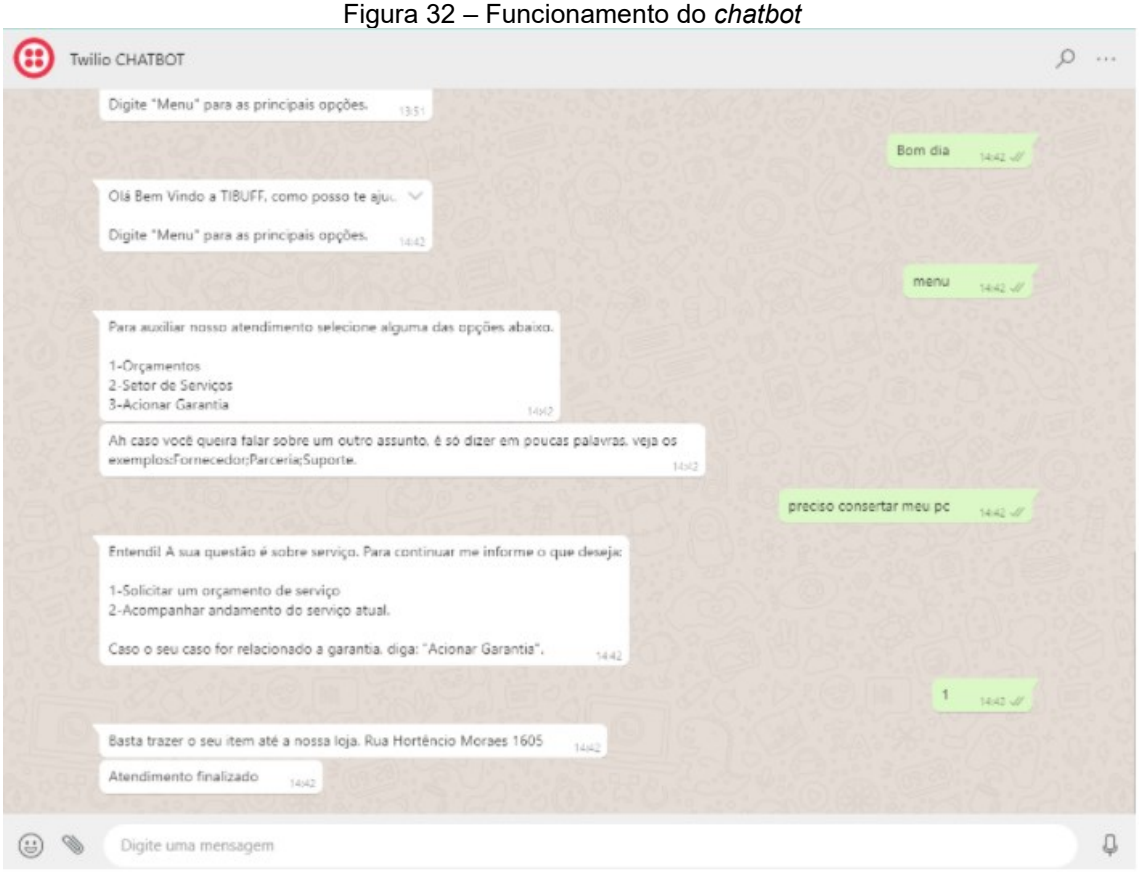
Figura 31 - NGRKOK



Fonte: Autor, 2021

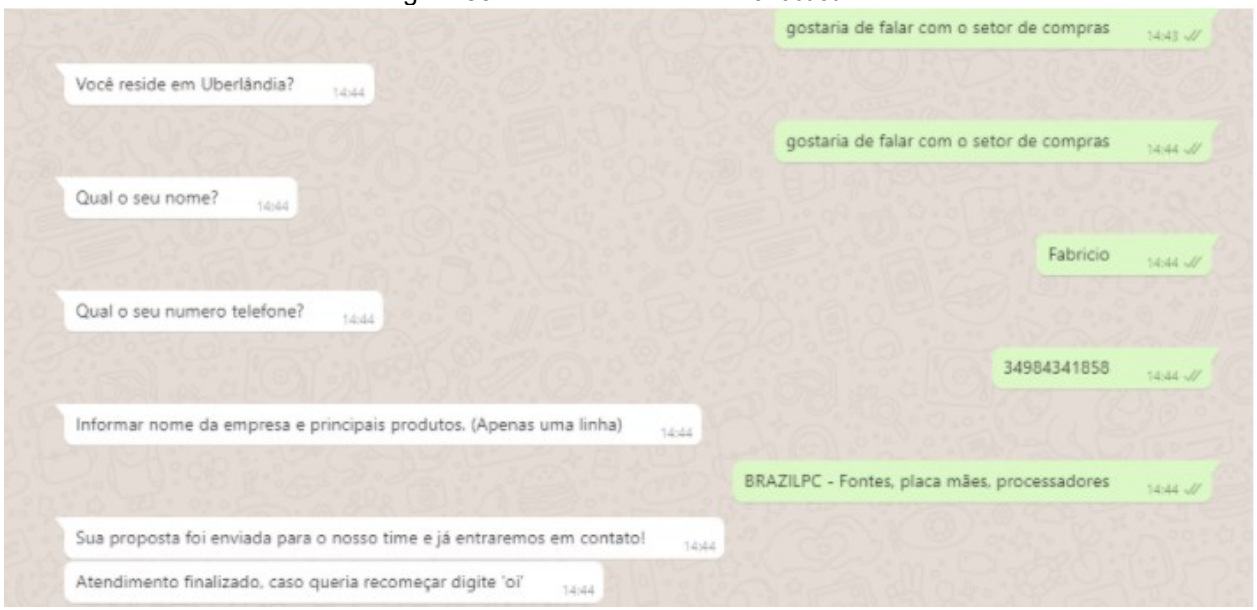
Por fim, apresenta-se o funcionamento do Bot, nas Figuras de 32 a 34.

Figura 32 – Funcionamento do *chatbot*



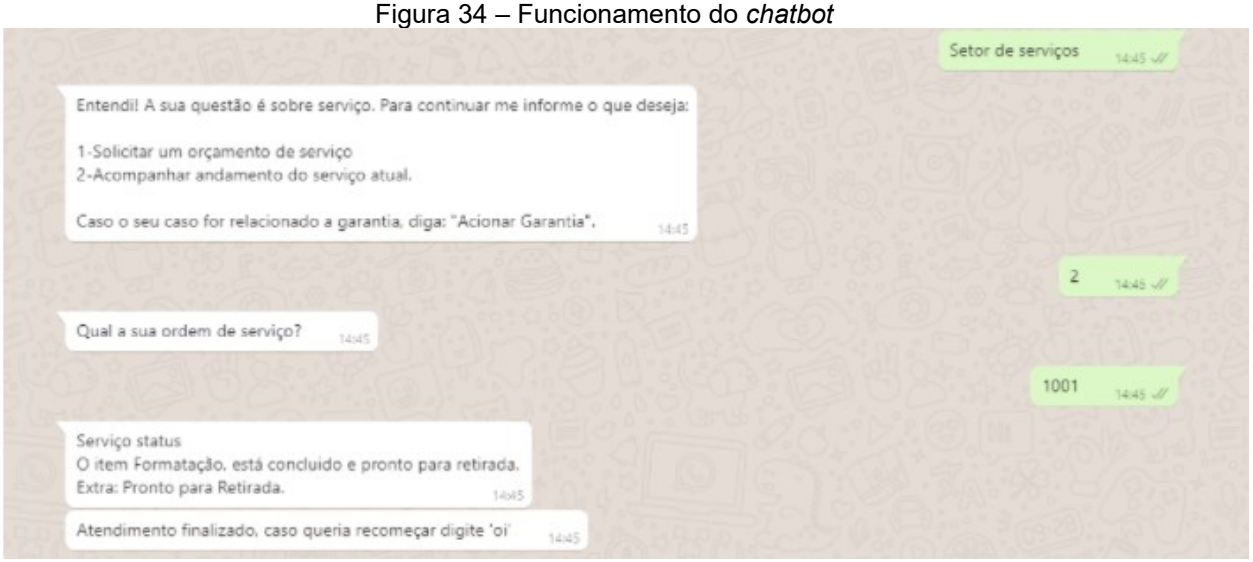
Fonte: Autor, 2021

Figura 33 – Funcionamento do *chatbot*



Fonte: Autor, 2021

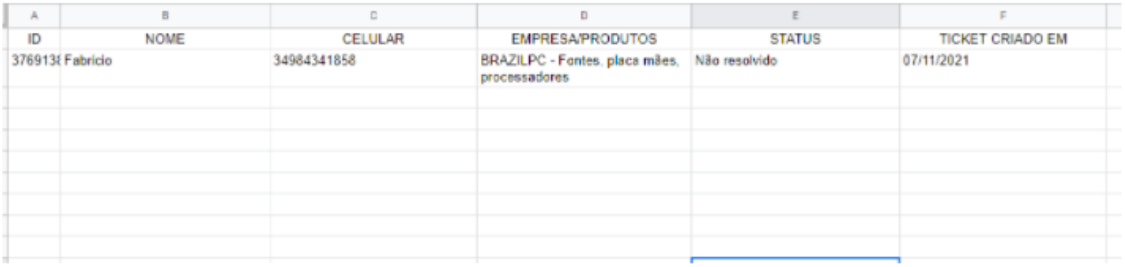
Figura 34 – Funcionamento do *chatbot*



Fonte: Autor, 2021

As Figuras 35 e 36 ilustram as planilhas sendo preenchidas automaticamente e buscando informações baseada nos diálogos.

Figura 35 – Planilhas



ID	NOME	CELULAR	EMPRESA/PRODUTOS	STATUS	TICKET CRIADO EM
376913(	Fabricio	34984341858	BRAZILPC - Fontes, placa mães, processadores	Não resolvido	07/11/2021

Fonte: Autor, 2021

Figura 36 – Planilhas

ID	PRODUTO	NOME	CELULAR	OBSERVAÇÃO	STATUS	TICKET CRIADO EM
1001	Formatação	Rodolfo	34984341858	Pronto para Retirada.	CONCLUIDO	
1002	Montagem	Daniel	34984429658	TESTE	EM PROCESSO	
					INICIANDO	

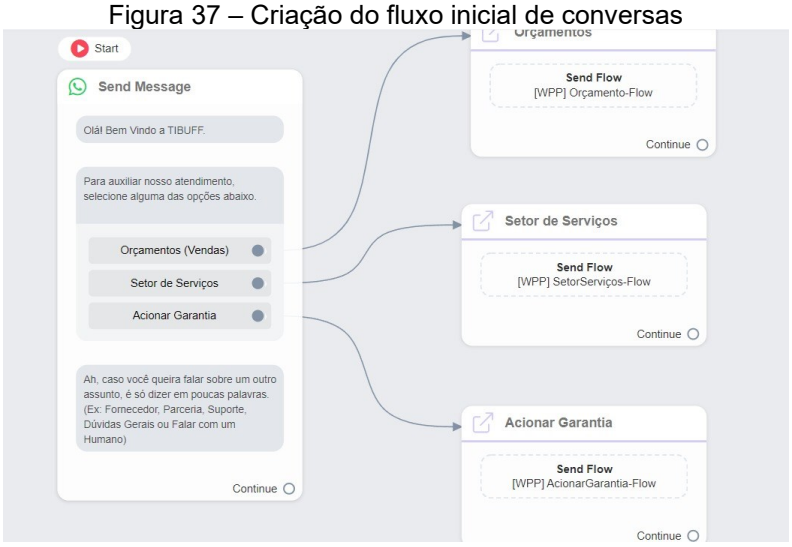
Fonte: Autor, 2021

3.5.4 Projeto ChatRace/Whatsapp

O processo teve início com o cadastro na plataforma CHATRACE e integração com a página do Instagram/Facebook.

Em seguida foram criados todos os fluxos para as conversas, sendo estabelecidos os seguintes termos: bem-vindo, acionar garantia, setor de serviços, orçamentos. Conforme ilustrado na Figura 37.

Figura 37 – Criação do fluxo inicial de conversas



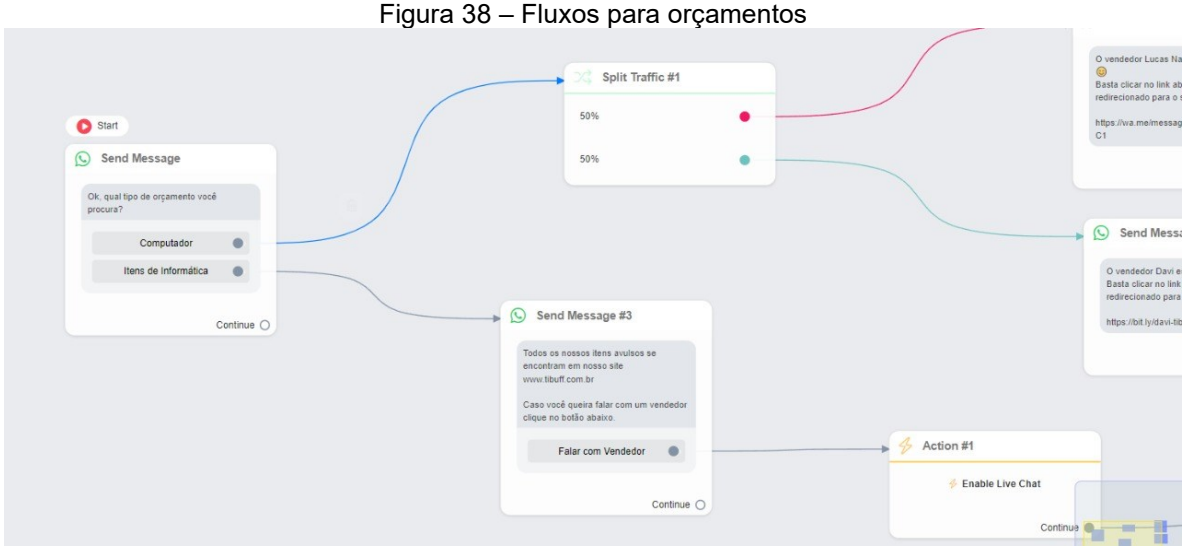
Fonte: Autor, 2021

Conforme Figura 38 o fluxo inicial da conversa visa fornecer o atendimento inicial ao cliente, filtrando sua necessidade de atendimento e o direcionando para o setor desejado. Esse será acionado com mensagens como 'bom dia', 'boa tarde ou 'olá'. Dessa forma o cliente poderá se guiar pelo fluxo recomendado ou mesmo digitar algo específico que esteja buscando.

Os fluxos se integram uns com os outros. O usuário então entra em um *flow*, porém com palavras chaves pode acionar facilmente a IA do software para entrar em novos *flows*. As Figuras de 38 a 43 ilustram, respectivamente, o fluxo para a opção orçamentos, o fluxo para serviços 1, o fluxo serviços 2, fluxo de garantia 1, fluxo de garantia 2, e fluxo de fallback.

Conforme Figura 38, na opção orçamento, duas opções são oferecidas nesse fluxo, orçamento para um computador, que será então destinado igualmente entre os vendedores do time da empresa, ou orçamento para periféricos, onde será informado o e-commerce da empresa. Para casos mais específicos é sempre oferecido a possiblidade de abrir o chat ao vivo com um atendente.

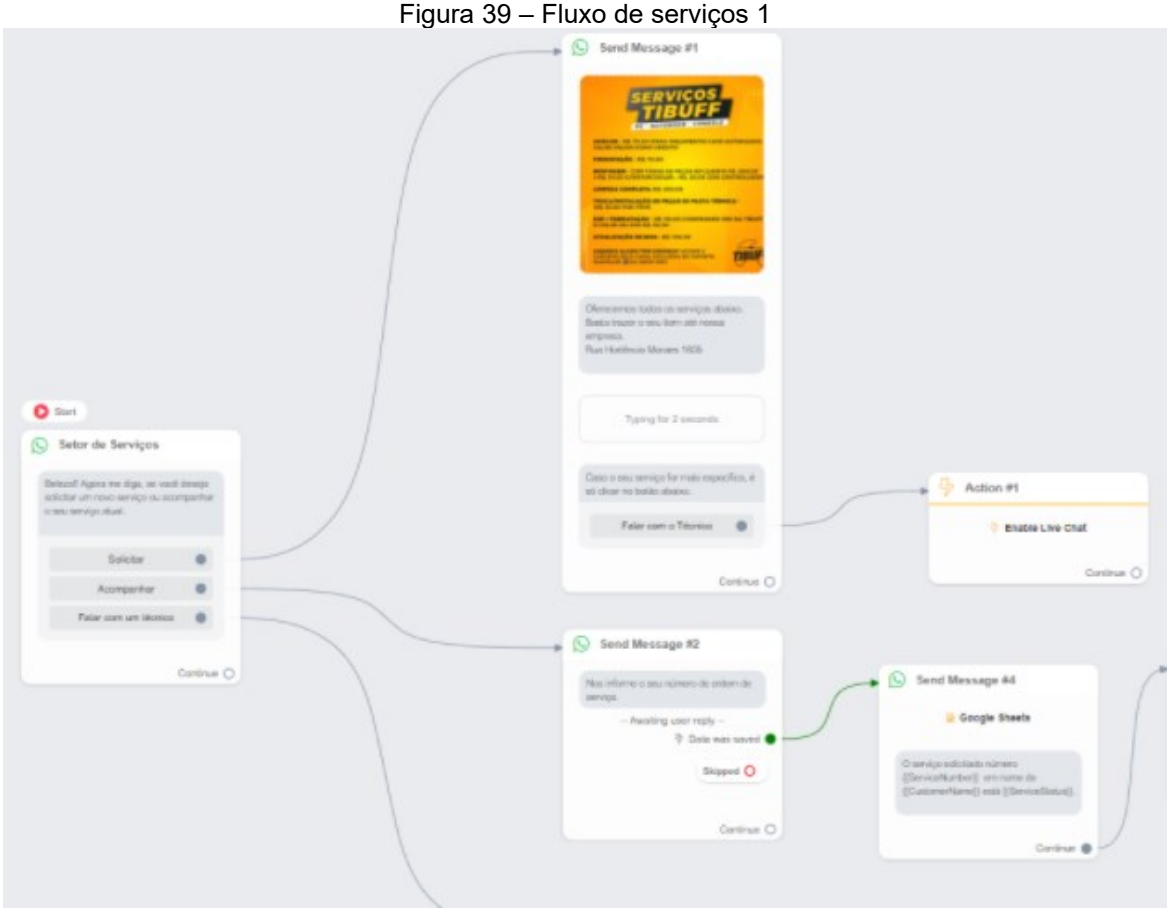
Figura 38 – Fluxos para orçamentos



Fonte: Autor, 2021

Conforme Figuras 39 e 40, no fluxo de serviço, o usuário pode solicitar ou acompanhar uma ordem de serviço. Ao solicitar, será enviada uma imagem com os serviços oferecidos e o processo para realizar o agendamento. Quando a demanda for por acompanhamento do serviço, os dados serão extraídos de uma planilha que do Google Planilhas que foi vinculada ao fluxo, por meio de integração pelo *software*.

Figura 39 – Fluxo de serviços 1



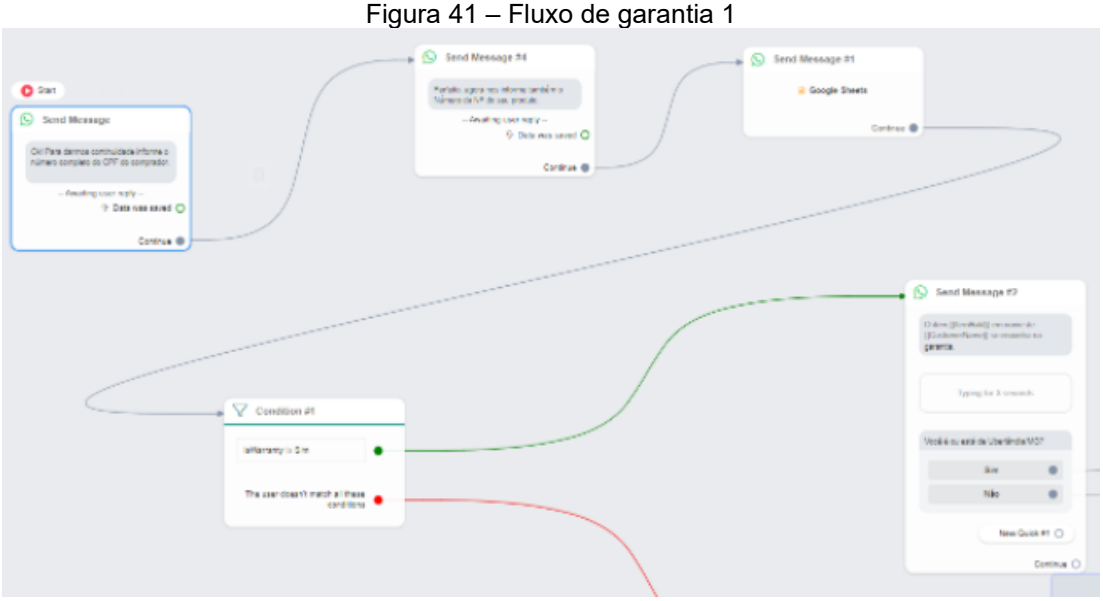
Fonte: Autor, 2021

Figura 40 – Fluxo de serviços 2

Fonte: Autor, 2021

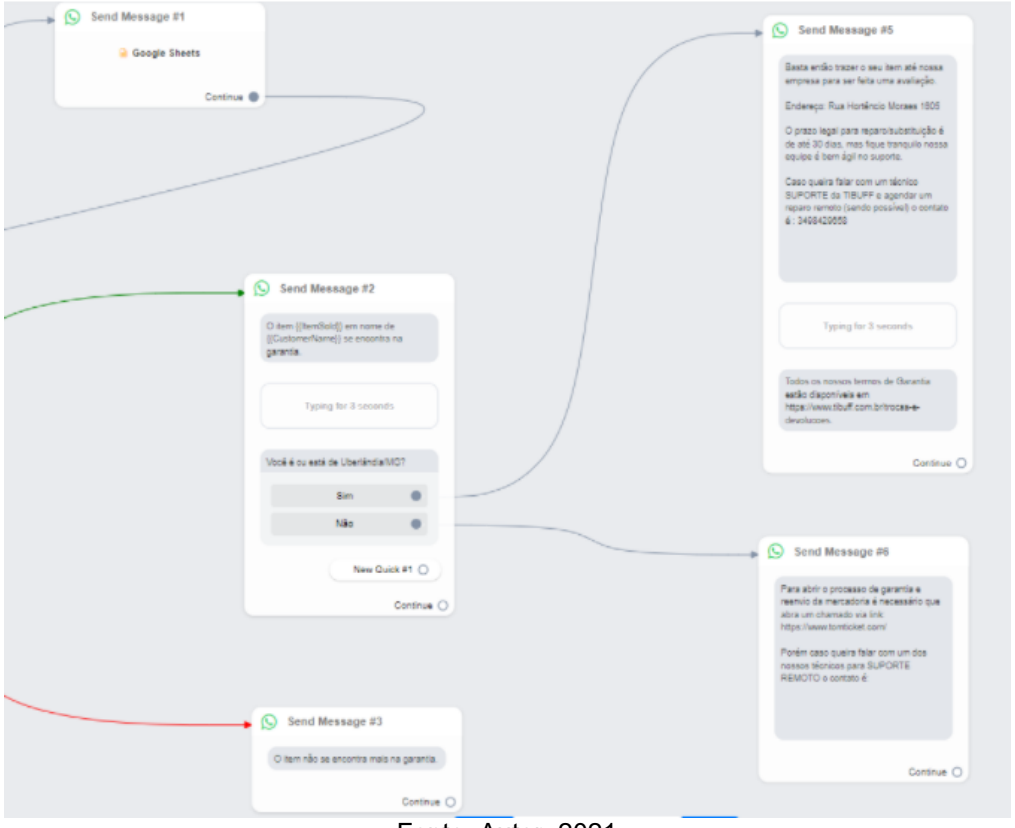
O fluxo de gantia é responsável por checar se a compra ainda está na garantia, os dados também são extraídos de uma planilha externa do Google Planilhas. Em caso de garantia, é enviado para o cliente os passos a ser seguidos para acioná-la. As Figuras 41 e 42 fazem ilustração do fluxo.

Figura 41 – Fluxo de garantia 1



Fonte: Autor, 2021

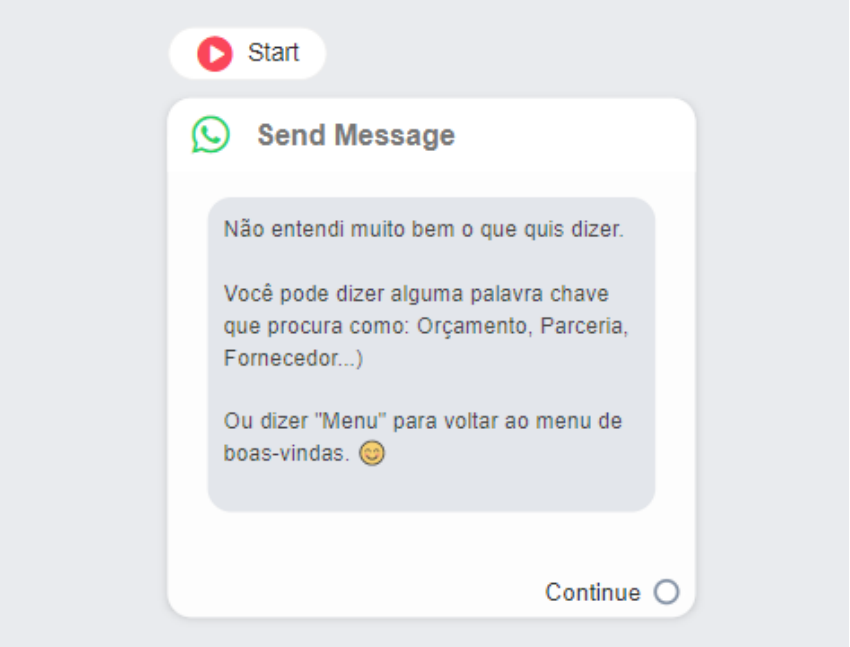
Figura 42 – Fluxo de garantia 2



Fonte: Autor, 2021

O fallback é o modo ativado para quando o chatbot não consiga entender o que o usuário quer dizer. Assim é encaminhado para o fallback para que repita a mensagem (Figura 43).

Figura 43 – Fluxo fallback



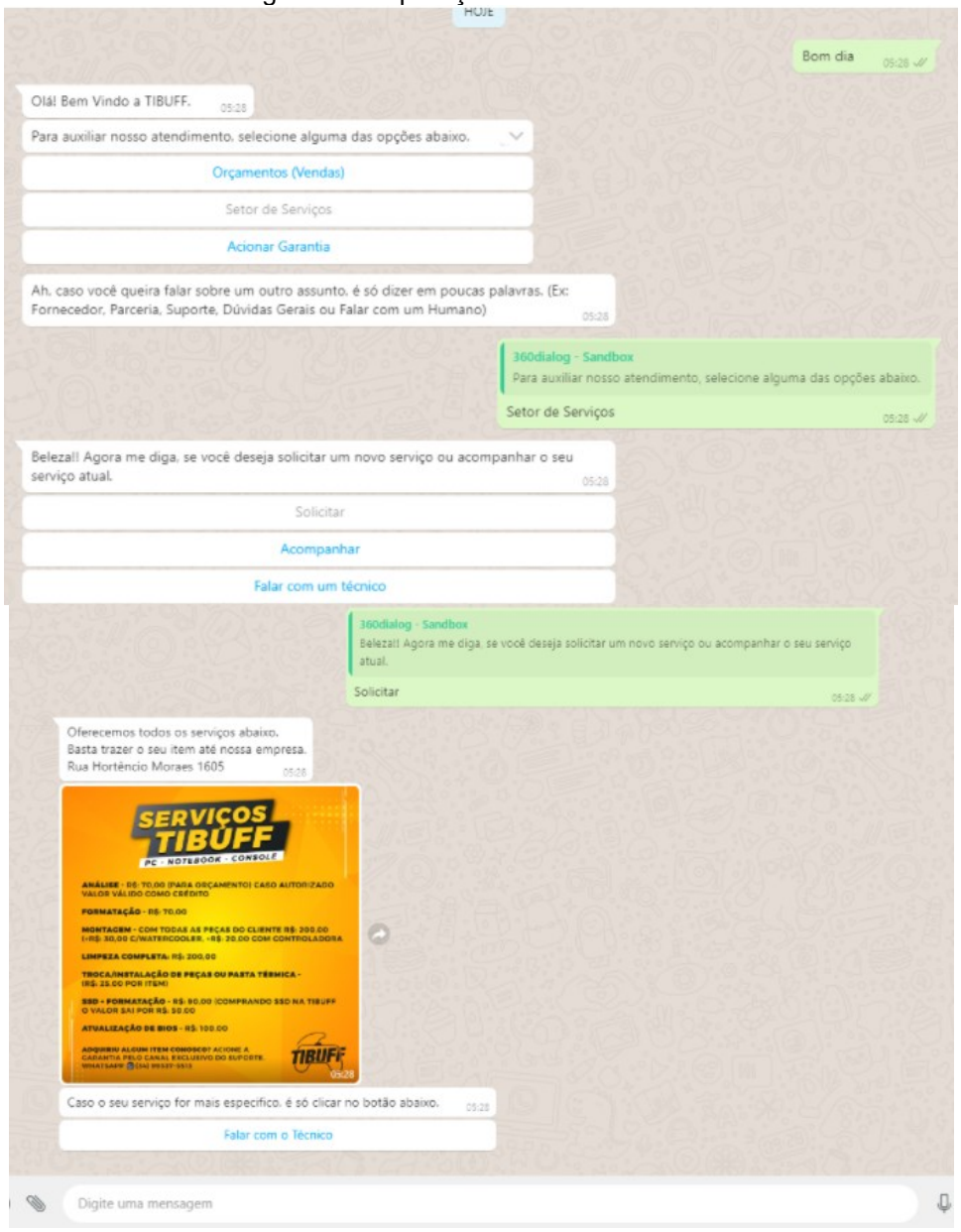
Fonte: Autor, 2021

Em todas as configurações insere-se as palavras-chave que serão direcionadas a fluxos e respotas diretas.

A interação com o Whatsapp foi realizada via API360WHATSAPP e o para o projeto foi utilizado o teste em SandBox. A integração é bem simples sendo feita pela plataforma da ChatRace integrando a um token individual.

A Figura 44 apresenta a aplicação em funcinamento.

Figura 44 – Aplicação em funcionamento



Fonte: Autor, 2021

Para esse projeto foi também utilizado um sistema hibrido de conversação, em alguns momentos onde o Bot não satisfaz a experiência do usuário, pode-se facilmente ser direcionado na mesma conversa para um atendente real, sem encaminhamentos para outros contatos . O atendimento é então continuado na plataforma do ChatRace.

4 RESULTADOS E DISCUSSÕES

O primeiro projeto criado, Projeto Python/Telethon (bot para Telegram), foi o sistema mais complexo, uma vez que é um sistema cru, o facilitador é a biblioteca do Telethon que faz a integração entre o Python e o Telegram. Os demais comandos, códigos, entendimentos, textos de reconhecimento são feitos manualmente com auxílio de códigos, assim demanda mais mão-de-obra e profissionais qualificados que tenham amplo conhecimento sobre programação. Além de estar mais suscetível a erros por ser feito quase que totalmente por meio de sistema manual e sem suporte de terceiros. No entanto é um projeto com custo de manutenção baixo pois não demanda sistemas terceiros, o único terceiro empregado é o servidor a fim de hospedagem. Ademais, a API também é aberta, não havendo também custos com esse quesito.

Já o segundo projeto desenvolvido, Projeto Dialogflow/Whatsapp, foi criado por meio do Dialogflow ferramenta do Google que permite a criação de chatbots, sendo mais facilitada pois já faz uso da inteligência do Google em processamento de linguagem natural, como por exemplo o banco de frases para perguntas e respostas. Além disso, ele já reconhece as intenções, entidades e contextos. A única integração necessária são informações de fora, impossíveis de haver respostas padrões, assim faz-se uso de uma API sincronizada as planilhas do Google utilizadas como fonte de busca para as respostas. É um sistema que se destaca pela facilidade para desenvolvimento, demandando pouco conhecimento em programação. Além disso, permite integração com diversos outros aplicativos, não somente o Whatsapp. O custo gasto nesse projeto se refere a API do Whatsapp, que diferente do Telegram, não é aberta, assim há a necessidade de pagamento.

O terceiro, Projeto ChatRace/Whatsapp, foi o que se mostrou mais simples quanto as etapas de desenvolvimento, em comparação aos outros dois anteriormente desenvolvidos. O sistema já conta com um banco de dados de inteligência artificial pré-definidos, facilitando o processo. Ademais é uma ferramenta muito visual onde é possível envio de imagens e criação de botões. No entanto, apresenta um custo pra utilização de torno de U$10 mais a API do Whatsapp, totalizando uma média de R$650,00 mês para manter o sistema no seu plano mais básico. Além disso, o ChatRace é um sistema hibrido que permite que o usuário a qualquer momento solicitar um atendente real, por meio de determinadas palavras-chave ou botões que

são ativados na conversação. O seu diálogo é então totalmente direcionado para um atendente que tem acesso a esta conversa dentro da plataforma do ChatRace.

Foram feitos vários testes com as três plataformas e estão em pleno funcionamento.

Os projetos obtiveram excelentes resultados quanto a sua aplicação prática e as requisições são satisfeitas durante o diálogo, com a utilização mais frequente é possível compreender outros termos que os usuários utilizam para se comunicar e aprimorar ainda mais a ferramenta.

Uma grande vantagem do DialogFlow é a exibição de todo o histórico dos diálogos que foram feitos e as compreensões do chatbot, sendo então uma ótima forma de melhoria para o agente.

Sobre os comparativos entre os agente criados podem ser avaliados de maneira mais visual conforme Tabela 1.

Tabela 1 - Comparativo entre os projetos elaborados

	Desenvolvimento IA (Qualidade)	Dificuldade Desenvolvimento	Integração com outros Canais	Escalabilidade	Valor Aproximado
Python TelegramBot	Complexo (Variável)	Médio-Avançado	Não	Sim	R$ 100,00/mês
DialogFlow WhatsApp (Twilio)	Integrado (Excelente)	Médio	Sim	Sim	R$0,20 /msg*
ChatRace WhatsApp (360Dialog)	Integrado (Básica)	Fácil	Sim, porém com variações	Sim	R$650,00/mês

Fonte: Autor, 2021

Pode-se identificar que há uma complexidade maior em desenvolver a inteligência artificial utilizando o Python. Percebe-se ainda uma diferença em dificudade de desenvolvimento e integração com outros canais. Por fim, os valores para implementação são variáveis. Desenvolvendo com Python, bem como as APIs, será necessário usar um servidor para fazer o deploy das aplicações, já para o DialogFlow e ChatRace também será necessário a aquisição da API do WhatsApp que tem um custo mensal (Twilio/360Dialog) e das plataformas de ChatBot (ChatRace/DialogFlow) cujos valores variam de acordo com quantidade de usuários ou requisições.

5 CONCLUSÃO E TRABALHOS FUTUROS

Com a realização deste estudo foi possível constatar a variedade de possibilidades existentes no mercado para execução do projeto de *chatbots*, bem como o progresso que essas aplicações vêm apresentando nos últimos anos. Foi também vastamente compreendido os conceitos intrínsecos e fundamentais para a criação de um agente conversacional.

Neste estudo, todos os projetos desenvolvidos apresentaram bons resultados quanto ao produto final, no entanto diferenciam-se frente a simplicidade para desenvolvimento do *chatbot*, integrações adicionais, opções de recursos dentro da conversa e custo.

O Projeto TelegramBot com Python/Telethon se mostrou mais trabalhoso porém destaca-se como o de menor custo. O Projeto WhatsApp Bot com DialogFlow apresenta um sistema de inteligência artificial muito bom, que facilita muito o processo de aprendizado do bot, bem como a qualidade do sistema final, além de, a depender do porte de atendimento do sistema não apresenta custo (somente o valor da API do WhatsApp), além de apresentar excelente integração. O Projeto WhatsApp Bot com ChatRace por sua vez é um sistema simples e muito interativo, no entanto apresenta um custo fixo elevado em relação aos demais.

Para esse estudo o Projeto ChatRace/Whatsapp foi o que se apresentou mais viável para a empresa, considerando a qualidade do sistema, facilidade de utilização e também sistema de chat ao vivo integrado.

Constatou-se ainda os fatores decisivos para avaliar os projetos: facilidade de criação tempo e habilidade necessários para desenvolver o projeto; personalização dos diálogos: quais as limitações de cada tipo de projeto; integração com outros serviços: serviços como Google Planilhas, sistemas de gestão, banco de imagens ; integração com outros canais: WhatsApp, Telegram, GoogleAssistente, WebChat, Messenger, Instagram; preço: valor para manter os serviços em funcionamento; escalabilidade: o qual propicio é para escalar em um nível grande de demanda; IA integrada/machine learning: inteligência para entender contextos, intenções, entidades, parâmetros e afins.

Portanto, trata-se de uma colaboração geral sobre o universo dos *chatbots* com aplicações práticas, sendo uma contribuição de alto valor para futuros trabalhos e projetos vinculados ao tema.

REFERÊNCIAS

AIS.pdf. Acesso em: 05 ago. 2021.

AMAZON. Amazon Lex Interfaces conversacionais para aplicações. 2018. Amazon Web Services, Inc. Disponível em: https://aws.amazon.com/pt/lex/. Acesso em: 1 nov. 2021.

ARAÚJO, E. F. S. Solução chatbot no ambiente acadêmico da UFRJ. Monografia (Engenharia de Computação e Informação). Escola Politécnica. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2020.

AYRES, M. Você sabe o que são e como surgiram os chatbots? (2018). Disponível em: https://hackel.com.br/historia-dos-chatbots/. Acesso em: 05 nov. 2021.
AZEVEDO, E. Desenvolvimento de Jogos e Aplicações em Realidade Virtual. 3. ed., Rio de Janeiro: Elsevier, 2005.

BRANDES, B. Dialogflow (api.ai) - breve introdução da plataforma. 2017. Disponível em: https://medium.com/botsbrasil/api-ai-breve-introdu%C3%A7%C3%A3o-da-plataforma-ecb2d77107a2. Acesso em: 1 nov. 2021.

CAMARA JUNIOR, A. T. D. Processamento de linguagem natural para indexação automática semântico-ontológica. Tese (Doutor do Programa de Pós-Graduação em Ciência da Informação). Universidade de Brasília. 2013.

CHARNIAK, E.; MCDERMOTT, D. Introduction to ai. Reading (Mass.): Addison, 1985.

CIRIACO, D. O que é Tegram. (2015). Disponível em: https://canaltech.com.br/apps/o-que-e-telegram/. Acesso em: 06 nov. 2021.

CLAUDINHO, B. S. Canais de comunicação pensados para o consumidor 4.0 estudo de caso: lança perfume. Monografia (Bacharel em Publicidade e Propaganda). Universidade do Sul de Santa Catarina. Palhoça. 2018.

CLAUDINO, B. S. Canais de comunicação pensados para o consumidor 4.0 estudo de caso: lança perfume. Monografia (Bacharel em Publicidade e Propaganda). Universidade do Sul de Santa Catarina. Palhoça. 2018.

COMARELLA, Rafaela Lunardi; CAFÉ, Ligia Maria Arruda. Chatterbot: conceito, características, tipologia e construção. Informação & Sociedade: Estudos, João Pessoa, v.18, n. 2, p.55-67, maio 2008.

COSTA, M. B. O que significa WhatsApp: entenda a origem do nome. (2021). Disponível em: https://canaltech.com.br/apps/o-que-significa-whatsapp/. Acesso em: 06 nov. 2021.

CUNHA, I. M., KOBASHI, N. Y. Análise documentária e inteligência artificial. Revista Brasileira de Biblioteconomia e Documentação, São Paulo, 24(1/4), 38-62, 1991.

DALE, R. The return of the chatbots. Natural Language Engineering, Cambridge University Press, v. 22, n. 5, p. 811–817, 2010. https://doi.org/10.1017/S1351324916000243

DIAS, W. S. SUSI – Uma Proposta de Chatbot para o Atendimento de Usuários do Ministério da Saúde. Monografia (Especialização em Informática do Departamento de Ciência da Computação). Universidade Federal de Minas Gerais. Brasília. 2019.

EBIT/NIELSEN/BEXS BANCO. Relatório Webshoppers 43 2020. Disponível em: https://www.mobiletime.com.br/wp-content/uploads/2021/03/Webshoppers_43.pdf. Acesso em: 06 ago. 2021.

FARIAS, S. B. Uma análise da utilização de chatbots de atendimento a clientes e o desenvolvimento à luz do dialogflow. Monografia (Bacharel em Ciência da Computação). Universidade Estadual Da Paraíba. Campina Grande. 2020.

FARINON, J. L. Análise e classificação de conteúdo textual. Monografia (Bacharel em Ciência da Computação). Universidade de Santa Cruz do Sul. Santa Cruz do Sul. 2015.

FILHO, Eustáquio Cézar Pereira. O Uso do Processamento de Linguagem Natural na Construção de Chatterbots. 2009. 46 f. Monografia (Especialização) - Curso de Ciências da Computação, Universidade Federal de Goiás, Catalão, 2009.

FRANKLIN, Stan; GRAESSER, Art. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: third international workshop on agent theories, architectures, and languages, 3., 1996, Berlin: Springer-verlag, 1996. Disponível em: http://ccrg.cs.memphis.edu/assets/papers/Is it an Agent, or just a Program – A Taxonomy.htm. Acesso em: 05 ago. 2021.

GHIDINI, I.; MATTOS, V. W. Desenvolvimento e aplicação de um chatbot para auxiliar o antendimento ao cliente. Monografia (Graduação em Sistemas de Informação). Universidade do Sul de Santa Catarina. Palhoça. 2018.

HADDAD, R. BOT Framework e Integração com Aplicações. 2018. https://msdn.microsoft.com/pt-br/mt721312.aspx. Acesso em: 1 nov. 2021.

HIPPISLEY, A. Lexical analysis. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.) Handbook of natural language processing. 2. ed. Boca Raton: Chapman & Hall/CRC, 2010.
HIRT, A. C. et al. Inteligência artificial: percepção dos usuários em relação ao seu uso. Faculdade Modelo, 2019.

http://gsigma.ufsc.br/~popov/aulas/Publicacoes/Zambiasi_Pinheiro_VCOTB2016_AN
INBOT. História dos Chatbots. (2018). Disponível em: https://in.bot/chatbots/historia-dos-bots.php. Acesso em: 05 nov. 2021.

INTERCOM. Relatório de tendências de suporte ao cliente de 2021. (2020). Disponível em: https://www.intercom.com/. Acesso em: 06 ago. 2021.

KOTLER, Philip e ARMSTRONG, Gary. Princípios de Marketing. 15. ed. São Paulo: 2014.

KURZWEIL, R. et al. The age of intelligent machines. [S.l.]: MIT press Cambridge, MA, 1990. v. 579.

LJUNGLÖF, P.; WIRÉN, M. Syntactic parsing. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.) Handbook of natural language processing. 2. ed. Boca Raton: Chapman & Hall/CRC, 2010.

LOPES, M. V. O marketing para o consumidor 4.0 - Estudo de caso sobre a personagem lu, do magazine luiza , como influenciadora virtual. Monografia (Bacharel em Publicidade e Propaganda). Universidade do Sul de Santa Catarina. Palhoça. 2020.

LUGER, G. Inteligência Artificial. 6. ed., São Paulo: Pearson, 2014.
MACIEL, R. **94% das empresas que usam bots já recuperaram o investimento feito na tecnologia.** Disponível em: https://canaltech.com.br/. Acesso em: 11 ago. 2021.

MALDONADO, M.; ALULEMA, D.; MOROCHO, D.; PROAÑO, M. System for monitoring natural disasters using natural language processing in the social network twitter. p. 1–6, 2016. https://doi.org/10.1109/CCST.2016.7815686

MANDELLI, A. S. Qualidade no atendimento ao cliente. Monografia (Bacharel em Administração). Universidade do Extremo Sul Catarinense – UNESC. Criciúma. 2014.

MCC-ENET. Referência em métricas e indicadores do consumo online no Brasil. (2020). Disponível em: https://www.mccenet.com.br/. Acesso em: 06 ago. 2021.

MCC-ENET. Referência em métricas e indicadores do consumo online no Brasil. (2021). Disponível em: https://www.mccenet.com.br/. Acesso em: 06 ago. 2021.

MELLISH, C.; PAN, J. Z. Natural language directed inference from ontologies. Artificial intelligence, v. 172, p. 1285–1315, 2008. https://doi.org/10.1016/j.artint.2008.01.003

MITTMANN, A. Implementação do chatterbot ELIZA na linguagem multiparadigma Oz. Universidade Federal de Santa Catarina. Disponível em: https://projetos.inf.ufsc.br/arquivos_projetos/projeto_304/TCC-Adiel.pdf. Acesso em: 05 ago. 2021.

O'GRADY, S. The RedMonk Programming Language Rankings: January 2018. Disponível em: https://redmonk.com/sogrady/2018/03/07/language-rankings-1-18/. Acesso em: 06 nov. 2021.

OLIVEIRA, F. A.; NAVAUX, P. O. A. Processamento de linguagem natural: princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa. Rio Grande do Sul, 2004.

OTHERO, Gabriel de Ávila; MENUZZI, Sérgio de Moura. Linguística computacional: teoria & prática. São Paulo: Parábola, c2005.

OVERVIEW. Developer Survey Results. (2018). Disponível em: https://insights.stackoverflow.com/survey/2018#overview. Acesso em: 06 nov. 2021.

PEREIRA, G.; PINHEIRO, M. A. Intercom – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação XIX Congresso de Ciências da Comunicação na Região Sul – Cascavel - PR – 31/05 a 02/06/2018 Conversando com robôs: O uso de chatbots na comunicação de marcas no Facebook Messenger.

POLLAK, M.; ANDERST-KOTSIS, G. Email monotoring and management with MS social bots. In: Proceedings oh the 19h International Conference on Information Integration and Web-based Applications & Service. IiWAS'17. [S.l]: ACM Press. 2017. https://doi.org/10.1145/3151759.3151799

Preeti dan BrahmaleenKaurSidhu. "Natural language processing," Int.J.Computer Technology & Applications, vol. 4, no. 5, hal. 751–758, 2013.

PYPI. Descrição do Projeto. (2018). Disponível em: https://pypi.org/project/Telethon/. Acesso em: 06 nov. 2021.2018
SANTOS, S. S. Desenvolvimento do chatbot ellen como ferramenta de alerta e acompanhamento para pessoas com doenças crônicas não transmissíveis. Monografia (Bacharel em Engenharia Biomédica). Universidade Federal do Rio Grande do Norte. Natal. 2018.

Saulo Popov; PINHEIRO, Patricia Leandra Barrufi. Persona Ex Machina: A Inteligência Artificial Aplicada na Cena Teatral. Disponível em:

SERCOM E QUALIBEST. Quais são os canais de SAC mais usados? (2020). Disponível em :https://selectra.net.br/anatel/noticias/consumidor/sercom-qualibest pesquisa-canais-sac-mais-usados. Acesso em: 11 ago. 2021.

SETZER, V. W. Alan Turing e a Ciência da Computação. Departamento de Ciência da Computação da USP. Disponível em: http://www.ime.usp.br/~vwsetzer/Turing-teatro.html. Acesso em: 05 ago. 2021.

SHIOZAWA, R. S. C. Qualidade no atendimento e tecnologia de informação. São Paulo: Atas. 1993.

SILVA, A. Consumidor 4.0: conheça o perfil desse público e veja como atendê-lo. (2020). Disponível em: https://www.docusign.com.br/blog/consumidor-4-0-conheca-o-perfil-desse-publico-e-veja-como-atende-lo. Acesso em: 06 ago. 2021.

SOUSA, D. L. A excelencia no atendimento ao cliente: um desafio estratégico em serviços bancários. Monografia (Bacharel em Admionistração). Universidade de Brasília. Brasília. 2011.

SOUZA, I. Atendimento ao cliente em e-commerce: como garantir a satisfação dos visitantes (2018-1). Disponível em: https://rockcontent.com/br/blog/atendimento-ao-

cliente-em-ecommerce/. Acesso em: 06 ago. 2021.

SOUZA, I. Consumidor 4.0: sua empresa já está preparada para atendê-lo? (2018-2). Disponível em: https://rockcontent.com/br/blog/consumidor-4-0/. Acesso em: 06 ago. 2021.

VENNERS, B. A fabricação de Python - Uma conversa com Guido van Rossum, Parte I. (2003). Disponível em: https://www.artima.com/articles/the-making-of-python. Acesso em: 06 nov. 2021.

WENI. História do chatbot: saiba como tudo começou. (2018). Disponível em: https://weni.ai/blog/historia-do-chatbot-saiba-como-tudo-comecou/. Acesso em: 06 nov. 2021.