



UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E
SISTEMAS - PECS

EWERTON FERREIRA BASTOS

*SAGMID: CHATTEBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM
IDOSOS UTILIZANDO AIML .*

SÃO LUIS

2021

EWERTON FERREIRA BASTOS

***SAGMID: CHATTERBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM
IDOSOS UTILIZANDO AIML.***

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão-UEMA como parte dos requisitos necessários para obtenção do Título de Mestre em Engenharia da Computação e Sistemas.

Orientador: Prof.^o Dr.^o. Reinaldo de Jesus da Silva

SÃO LUIS

2021

Bastos, Ewerton Ferreira.

Sagmid: chatterbot para auxílio de gestão de medicamentos em idosos utilizando AIML / Ewerton Ferreira Bastos. – São Luís, 2020.

77 f

Dissertação (Mestrado) – Curso de Engenharia da Computação e Sistemas, Universidade Estadual do Maranhão, 2020.

Orientador: Prof. Dr. Reinaldo de Jesus da Silva.

1.Chatbot. 2.Artificial Intelligence Markup Language-AIML.
3.Processamento de linguagem natural. I.Título.

CDU: 004.434:615.2-053.9

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E
SISTEMAS - PECS

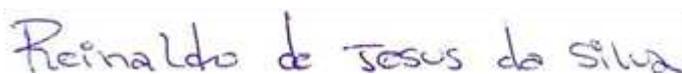
A Banca Examinadora, abaixo assinada,
aprova a proposta de Dissertação de Mestrado
em 31 de março de 2021.

***SAGMID: CHATTEBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM
IDOSOS UTILIZANDO AIML .***

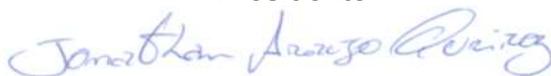
elaborado por

EWERTON FERREIRA BASTOS

BANCA EXAMINADORA:



Dr. Reinaldo de Jesus da Silva, PECS/UEMA
Presidente



Dr. JONATHAN ARAÚJO QUEIROZ ,UFMA.
Examinador Externo a instituição



Dr. LUIS CARLOS COSTA FONSECA, PECS/UEMA.
Examinador Interno

AGRADECIMENTOS

A presente dissertação não poderia chegar à sua fase final sem o precioso apoio de várias pessoas que foram importantes ao longo desse processo.

Primeiramente, agradeço a Deus por ter me concedido forças nos momentos difíceis e ter me permitido chegar até a conclusão deste processo.

À minha esposa pelo incentivo, compreensão e paciência demonstrada durante o período de desenvolvimento do meu trabalho.

Aos meus pais que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória; em especial, para mais esta conquista na vida acadêmica e profissional.

De forma especial, agradeço meu orientador Prof.^o Dr.^o. Reinaldo de Jesus da Silva que sempre se mostrou disponível durante a realização desta dissertação.

Ao amigo e professor Dr.^o. Jonathan Araújo Queiroz pelas orientações e correções durante a escrita das minhas pesquisas.

A todos os meus professores do curso de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão pela excelência técnica de cada um.

Por fim, agradeço a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

RESUMO

SAGMID: CHATTERBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM IDOSOS UTILIZANDO AIML.

Com o avanço da inteligência artificial surgiram os sistemas conversacionais, chamados de *chatbots*, que são aplicativos que simulam a conversa de um ser humano. Com o desenvolvimento de algoritmos, que são capazes de reconhecer e sintetizar a voz e a integração destes com os *chatbots*, torna possível a interação por voz com o usuário, de forma que este tenha a percepção de que está conversando com o sistema. Os *chatbots* têm estado cada vez mais presentes no ecossistema digital, principalmente no que se refere à automatização do atendimento ao cliente. Além disso, a tecnologia usada para o desenvolvimento de sistemas conversacionais pode ser empregada para solucionar problemas da área da saúde. O objetivo do presente trabalho é apresentar um protótipo de um *chatbot*, que pode ser utilizado para auxiliar os idosos na gestão de medicamentos. O sistema conversacional proposto possui uma base de conhecimento construída em *Artificial Intelligence Markup Language-AIML*, além disso, o *chatbot* irá sintetizar e reconhecer voz para funcionar como interface de interação com os usuários idosos. A protótipo da ferramenta visa facilitar a gestão de medicamento para idosos com elementos de interação por voz.

Palavras-chaves: Chatbot, Artificial Intelligence Markup Language-AIML, Processamento de Linguagem Natural.

ABSTRACT

SAGMID: CHATTERBOT FOR DRUG MANAGEMENT ASSISTANCE IN THE ELDERLY USING AIML.

With the advancement of artificial intelligence, conversational systems, called chatbots, suggest applications that simulate the conversation of a human being. With the development of algorithms, which are able to recognize and synthesize the voice and the integration of these with the chatbots, it makes possible the interaction by voice with the user, so that the user has the perception that he is talking with the system. Chatbots have been increasingly present in the digital ecosystem, especially regarding the automation of customer service. In addition, the technology used to develop conversational systems can be used to solve health problems. The aim of the present work is to present a prototype of a chatbot, which can be used to assist the elderly in the management of medications. The proposed conversational system has a knowledge base built on Artificial Intelligence Markup Language-AIML, in addition, the chatbot will synthesize and recognize voice to function as an interface for interaction with elderly users. The prototype of the tool aims to facilitate medication management for the elderly with voice interaction elements.

Keywords: Chatbot, Artificial Intelligence Markup Language-AIML, Natural Language Processing.

Keywords: Chatbot, Artificial Intelligence Markup Language-AIML, Natural Language Processing.

Sumário

1 INTRODUÇÃO.....	13
1.2 Objetivo Geral.....	14
1.3 Objetivos Específicos	14
1.4 Justificativa	15
1.5 Estrutura do Trabalho	16
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1 Fundamentos de Chatterbot	17
2.2. Estado da Arte dos Chatbot's	20
2.2.1. Teste de Turing, Quarto Chinês.	20
3. TECNOLOGIAS PARA O DESENVOLVIMENTO DE UM CHATBOT.....	25
3.1 Processamento de Linguagem Natural (PLN)	25
3.1.1 Processamento Morfológico	26
3.1.2 Análise Sintática.....	29
3.1.3 Análise Semântica.....	30
3.1.4 Áreas de Aplicação do Processamento de Linguagem Natural.....	31
3.2. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE-AIML	32
3.2.1 Chatterbot criados com AIML.	35
4. REVISÃO SISTEMÁTICA DA LITERATURA.....	37
4.1 Protótipo de Aplicação Android para Controle da Rotina De Medicamentos para Idosos.....	37
4.2 Aplicação Web para o Auxílio da Toma de Medicamentos e de gestão de Informação de Consultas Médicas Para Idosos-Old Care.....	38
4.3 EMED	39
4.4 Medamigo: Um Aplicativo de Auxílio a Idosos a Ingestão de medicamentos	40
4.5 SMAI - Sistema Móvel de Assistência ao Idoso	40

4.6 Helpcare: Um Protótipo de Chatbot para O Auxílio Do Tratamento de doenças Crônicas.....	41
4.7 Roborto.....	42
4.8 BOT ELLEN.....	43
4.9 Sumário de Trabalhos Relacionados.....	43
5 SOLUÇÃO PROPOSTA: CHATTERBOT FOR DRUG MANAGEMENT ASSISTANCE IN THE ELDERLY USING AIML.....	45
5.1 Arquitetura Proposta.....	45
5.3 Modelagem do SAGMID.....	46
5.3.1 Caso de Uso.....	46
5.3.2 Diagrama De Classe.....	47
5.3.3 Diagrama De Sequência.....	49
6. CHATBOT SAGMID.....	51
6.1 . Validação e Verificação.....	53
6.1.2 Procedimento Para Coleta E Análise De Dados.....	53
7. CONCLUSÃO.....	59
Referências.....	61
APÊNDICE.....	65

LISTA DE FIGURAS

FIGURA 1: PROJEÇÃO DA POPULAÇÃO BRASILEIRA ATÉ 2060	15
FIGURA 2: ESTRUTURA DOS MODELOS GENERATIVOS E BASEADOS EM EXTRAÇÃO, PASSÍVEIS DE SEREM IMPLEMENTADOS NUM CHATBOT.....	19
FIGURA 3: GRAUS DE DIFICULDADE ASSOCIADOS AO DESENVOLVIMENTO DOS DIFERENTES TIPOS DE SISTEMAS CONVERSACIONAIS.....	19
FIGURA 4: FUNCIONAMENTO DA MÁQUINA DE TURING.	21
FIGURA 5: ARQUITETURA DO ELIZA.....	22
FIGURA 6:CONVERSA COM CHATTERBOT ELIZA.	22
FIGURA 7: ARQUITETURA DO JULIA.	23
FIGURA 8: TRECHO DE CONVERSA COM JULIA	23
FIGURA 9: REGRA DO ALGORITMO RSPL STEMME.....	28
FIGURA 10: ABORDAGEM EM BUSCA EM TABELA.....	28
FIGURA 11: ARVORE DE CONSTITUÊNCIA.	29
FIGURA 12:CATEGORIA AIML.....	33
FIGURA 13: CAMINHO PADRÃO PARA UMA CATEGORIA.	33
FIGURA 14 EXEMPLO DO OBJETO GRAPHMASTER	33
FIGURA 15: EXEMPLO DE CATEGORIA PADRÃO.	35
FIGURA 16: REDUÇÃO SIMBÓLICA.....	36
FIGURA 17: PROTÓTIPO DE APLICAÇÃO ANDROID PARA CONTROLE DA ROTINA DE MEDICAMENTOS PARA IDOSOS.	37
FIGURA 18: APLICAÇÃO WEB PARA O AUXÍLIO DA TOMA DE MEDICAMENTOS E DE GESTÃO DE INFORMAÇÃO DE CONSULTAS MÉDICAS PARA IDOSOS.....	39
FIGURA 19: : APLICAÇÃO WEB PARA O AUXÍLIO DA TOMA DE MEDICAMENTOS E DE GESTÃO DE INFORMAÇÃO DE CONSULTAS MÉDICAS PARA IDOSOS	39
FIGURA 20: MEDAMIGO	40
FIGURA 21: SMAI.	41
FIGURA 22: HELPCARE.	42
FIGURA 23: ROBORTO.	42
FIGURA 24: BOT ELLEN.....	43
FIGURA 25: REPOSTA PARA SAUDAÇÃO DADA PELO CHATBOT	45
FIGURA 26: ARQUITETURA SAGMID.....	46
FIGURA 27: MODELAGEM DE CASO DE USO DO SISTEMA PARA APOIO DO TRATAMENTO FARMACOLÓGICO PARA IDOSOS DESENVOLVIDO NO STARTUML.....	47

FIGURA 28: DIAGRAMA DE CLASSE DO SISTEMA PARA APOIO DO TRATAMENTO FARMACOLÓGICO PARA IDOSOS DESENVOLVIDO NO STARTUML.	48
FIGURA 29: DIAGRAMA DE SEQUÊNCIA DO SISTEMA PARA APOIO DO TRATAMENTO FARMACOLÓGICO PARA IDOSOS DESENVOLVIDO NO STARTUML.FONTE : AUTOR (2021).....	50
FIGURA 30: FLUXO DE DADOS DO SAGMID.....	52
FIGURA 31: INTERAÇÃO DO USUÁRIO COM O SAGMID.	52

LISTA DE TABELAS

TABELA 1:SUMARIO DE TRABALHO RELACIONADO	44
TABELA 2: PERCENTUAL DE PARTICIPANTES.....	53

LISTA DE GRÁFICOS

GRÁFICO 1:IDADE DOS PARTICIPANTES POR SEXO.....	54
GRÁFICO 2: DESEMPENHO DO SAGMID.	54
GRÁFICO 3:COMO FOI SUA SATISFAÇÃO EM CADASTRAR UM MEDICAMENTO OU CONSULTA ?	56
GRÁFICO 4:A INTERAÇÃO POR VOZ COM A FERRAMENTA FACILITOU O CADASTRO DO MEDICAMENTO OU CONSULTA ?	56
GRÁFICO 5:O ALARME CONFIGURADO NO MOMENTO DO CADASTRO DO MEDICAMENTO OU CONSULTA FUNCIONOU DA MANEIRA ESPERADA ?.....	56
GRÁFICO 6: O APLICATIVO CONSEGUIR RESPONDER SUAS PERGUNTAS DE FORMA SATISFATÓRIA ?	57

LISTA DE ABREVIATURAS

TIC: Tecnologia da Informação e comunicação.

AIML: Linguagem de Marcação de Inteligência Artificial.

IA: Inteligência Artificial.

SAGMID: CHATTERBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM IDOSOS UTILIZANDO AIML ..

UML: Linguagem de Modelagem Unificada.

WSD :Word Sense Disambiguation

LSA: Latent Semantic Analysis

RSPL: Removedor de sufixo da língua portuguesa

SRL: Semantic Role Labelling

XML: Extensible Markup Language

1 INTRODUÇÃO

O desenvolvimento da inteligência artificial abriu caminhos para mudar o modo de interagirmos com os sistemas computacionais. Os estudos de Alan Turing sobre “Computing Machinery and Intelligence”, se “Podem as máquinas pensar?”, conceitua o problema, o qual foi chamado de jogo de imitação, atualmente amplamente conhecido como teste de Turing (DIAS, 2018). A partir deste experimento abriu-se o caminho para o desenvolvimento de sistemas que pudessem interagir com o usuário através do diálogo, de modo semelhante a interação humana.

Com a inteligência artificial é possível desenvolver sistemas conversacionais que já estão inseridos no nosso cotidiano, como por exemplo, os smartphones com sistema operacional Android que pode ser acessado por voz. Essa interação é possível através do google assistente, que executa tarefas através do reconhecimento de voz (GUO, 2018). Outro assistente virtual conhecido é o sistema Cortana da Microsoft.

Os assistentes virtuais atualmente são chamados de *chatbot*, pois podem interagir com os usuários através de um contexto conversacional e executar tarefas solicitadas pelo usuário, o qual pode interagir com o sistema através do reconhecimento e síntese de voz, assim como entrada e saída de texto (COMARELLA, 2008).

Os *chatbots* têm sido empregados nas áreas de marketing, vendas e atendimento, como por exemplo: a Secretaria de Saúde do Estado de Goiás¹ disponibilizou em seu endereço eletrônico um *chatbot* para tirar dúvidas sobre a Covid-19.

Uma outra atividade em que pode ser empregado o *chatbot* é como ferramenta de auxílio no cuidado à saúde dos idosos, pois, segundo o Instituto Brasileiro de Geografia e Estatística -IBGE , a expectativa de vida do brasileiro atingiu a média 76,3 anos em 2018.

A população idosa é cerca 28 milhões de pessoas com 60 anos ou mais, este número representa 13% da população brasileira. Estima-se que 70%,

¹ <https://www.saude.go.gov.br/coronavirus>

aproximadamente, dos idosos fazem uso de algum medicamento e, cerca de 20% destes, utilizam em média três medicamentos simultaneamente (SENA, 2018). Segundo Pedrazzi, (2007), as doenças crônicas que mais acometem os idosos são: hipertensão arterial, problemas de coluna, problemas para dormir, problemas cardíacos, má-circulação, osteoporose, artrite, artrose, diabetes mellitus, acidente vascular cerebral.

Diante das morbidades que afetam os idosos é necessário ter um tratamento farmacoterapêutico eficiente para o controle da doença e a manutenção da qualidade de vida. Neste contexto, o uso de assistentes virtuais como ferramenta de auxílio no tratamento farmacoterapêutico eficiente, pode auxiliar a reduzir as intercorrências relacionadas ao uso de medicamentos, tais como: interação medicamentosa, efeitos colaterais e erro de administração do medicamento.

1.2 Objetivo Geral

Este trabalho tem como objetivo propor uma ferramenta para auxiliar o idoso na gestão de medicamentos de uso contínuo.

1.3 Objetivos Específicos

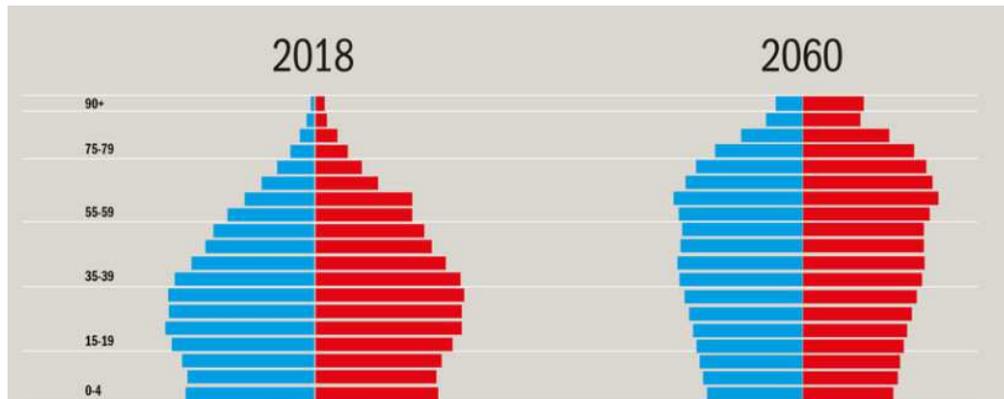
Para que o objetivo geral possa ser atingido, são propostos os seguintes objetivos específicos:

- Investigar o uso de *chatbot* como ferramenta de auxílio na gestão de medicamentos;
- Apresentar a arquitetura do chatterbot para auxílio de gestão de medicamentos em idosos utilizando aiml.
- Avaliar a arquitetura do sistema computacional para auxiliar o idoso na gestão de medicamentos de uso contínuo;
- Construir uma ferramenta para auxiliar o idoso na gestão de medicamentos;
- Estimar o desempenho da aplicação em ambiente real;

1.4 Justificativa

De acordo com estudos realizados pelo IBGE², em 2060 o número de idosos com mais de 60 anos será equivalente a um quarto da população, conforme mostra a figura 1 em uma projeção do cenário atual de idosos no nosso país (LIMA, 2009).

Figura 1: Projeção da população brasileira até 2060



Fonte: <https://censo2020.ibge.gov.br/>. Acessado em 13/06/2020

O aumento da idade de um indivíduo está associado ao aparecimento de morbidade que levam ao uso de medicamentos contínuos (FAZAN; PEREIRA; GODOY, 2009). Neste contexto, o uso de TIC pode auxiliar no tratamento de forma que os medicamentos possam ser tomados na quantidade certa e na hora correta.

Pesquisa realizada pela ferramenta Google Play, em 10/06/2020, usando a palavra *alarme de medicamento*, trouxe um número aproximado de 100 aplicativos que auxiliam na administração de medicamentos. De acordo com a pesquisa realizada, nenhum dos aplicativos verificados utiliza *chatbots* para melhor interação com o paciente.

De acordo com os levantamentos feitos nas literaturas que embasam este trabalho, é possível perceber que os sistemas conversacionais estão sendo utilizados para ajuda a cuidar da saúde do idoso no que desrespeito ao uso correto dos medicamentos de uso contínuo (ROCHA, 2018)(OLIVEIRA, 2019).

Conclui-se que os *chatbots* podem ser usados para ajudar a cuidar da saúde do idoso, facilitando a interação com o sistema através de mecanismos

² Fonte: <https://censo2020.ibge.gov.br/>. Acessado em 13/06/2020

de reconhecimento de voz e síntese de voz que são providos em dispositivos móveis.

1.5 Estrutura do Trabalho

Os capítulos deste trabalho serão distribuídos da seguinte forma:

Capítulo 1 - Introdução: O primeiro capítulo orienta o leitor sobre o assunto abordado no projeto, apresentando levantamentos que servirão de base para o entendimento referente à intenção do desenvolvimento deste trabalho.

Capítulo 2 - Fundamentação Teórica: O segundo capítulo aborda a fundamentação teórica, apresentando os principais conhecimentos prévios referentes às ideias apresentadas neste trabalho.

Capítulo 3 - Tecnologia para desenvolvimento de um *chatbot*: Neste capítulo serão abordados os conceitos sobre Processamento de Linguagem Natural – PNL e AIML.

Capítulo 4 - Revisão sistemática da literatura: Neste capítulo serão apresentadas as literaturas, que embasarão o desenvolvimento deste trabalho.

Capítulo 5 - Solução proposta, SAGMID: Chatterbot para auxílio de gestão de medicamentos em idosos utilizando aiml, será apresentada a arquitetura do sistema, os diagramas UML, tais como: diagrama de caso de uso, diagrama de classe e diagrama de atividade do SAGMID.

Capítulo 6 - *Chatbot* SAGMID: Neste capítulo será apresentada a linguagem de desenvolvimento do sistema, o fluxo de funcionamento e a validação do *chatbot*.

Capítulo 7 - Considerações finais e trabalhos futuros: O último capítulo demonstrará o resultado do desenvolvimento do software.

2. FUNDAMENTAÇÃO TEÓRICA

Para a compreensão do que vem ser um *chatbot* e a sistemática de funcionamento, faz-se necessário um estado-da-arte que atraia no leitor a curiosidade em desvendar as possibilidades desse recurso como ferramenta de auxílio na saúde do idoso.

2.1 Fundamentos de Chatterbot

Os sistemas conversacionais, chatbot, vem sendo utilizado em diferentes áreas como: marketing, atendimento ao cliente, suporte técnico, educação, treinamento (IGARASHI, 2019). Para Kříž, (2017), o termo chatbot pode ser descrito sobre diferentes formas, agentes conversação, sistemas de conversação, chatterbot, ou simplesmente robô, Estes termos são usados como sinônimos para a palavra chatbot.

Em Comarrela, (2008), defini chatbot como um agente de conversão baseado em regras e / ou inteligência artificial que simula uma conversa real usando uma linguagem natural para se comunicar com os usuários.

Os agentes de conversão devem apresentar características de aprendizagem, raciocínio e personalidade de acordo com o domínio para o qual foi construído, passando para o utilizador segurança no atendimento prestado pelo chatbot (KŘÍŽ, 2017).

Em (SANSONNET, 2006) , está descrito os atributos básicos que personalizam o comportamento esperado pelo usuário do chatbot. Os atributos são:

Agente de aprendizagem dialógica –O sistema deve ser capaz de entender o problema do usuário através de entradas, texto ou voz, fornecida como entrada de dados. O processamento de linguagem natural emprega técnicas que permitem que os sistemas possam extrair o significado da solicitação do usuário.

Agente racional – Deve ter acesso ou deve ser alimentado por uma base de conhecimento externa e representativa do senso comum e conhecimento geral, de maneira a fornecer um certo tipo de competência quanto a formulação da resposta ao utilizador.

Agente personificado/corporificado – Deve fornecer a função de presença antropomórfica para melhorar a confiança do sistema pelo usuário. A função de presença é características que um Chatbot deve possuir.

Além dos atributos citados os chatbot apresentam diferentes formas de serem implementados. Dias (2018), descreve que eles podem de domínio aberto, domínio fechado, baseados em extração de informação, generativos, conversacionais e orientados a tarefas.

A primeira característica do sistema conversacional é quanto ao domínio do diálogo de interação entre a máquina e o usuário do sistema, que pode ser classificado em **aberto** ou **fechado** (JUNIOR, 2017).

Um sistema de domínio aberto é usualmente mais difícil de projetar e implementar, pois não há uma meta ou intenção bem definidas. O número de possibilidades pode ser infinito, por exemplo, uma conversa em uma rede social pode ser sobre qualquer assunto, para ter uma resposta satisfatória pelo agente conversacional é necessário ter ampla base de conhecimento sobre qualquer tema que inicie uma conversa (CAHN, 2017).

Por outro lado, os sistemas de domínio fechado o chatbot é projetado para domínio da aplicação reduzindo as possíveis entrada e saídas são, pois o sistema tenta alcançar um objetivo muito específico tais como: suporte técnico ao cliente, venda entre outros (CAHN, 2017). Como exemplo de sistema fechado, encontramos o atendente virtual da empresa de telecomunicação oi, o chatbot Joice³, que tem a função de auxiliar o cliente em identificar e solucionar problemas na rede de dados residencial.

Outra importante característica do chatbot, é o modelo implementado para a devolução da resposta para o utilizador: se o sistema é baseado na extração de informação ou modelo generativo (DIAS, 2018).

Nos modelos baseados em extração da informação usam um repositório de repostas predefinida e um tipo de resposta apropriado de acordo com a entrada e no contexto. A sua heurística pode ser tão simples como a correspondência baseada em regras previamente formuladas, ou tão complexa

³ <https://www.oi.com.br/assistentevirtual/>

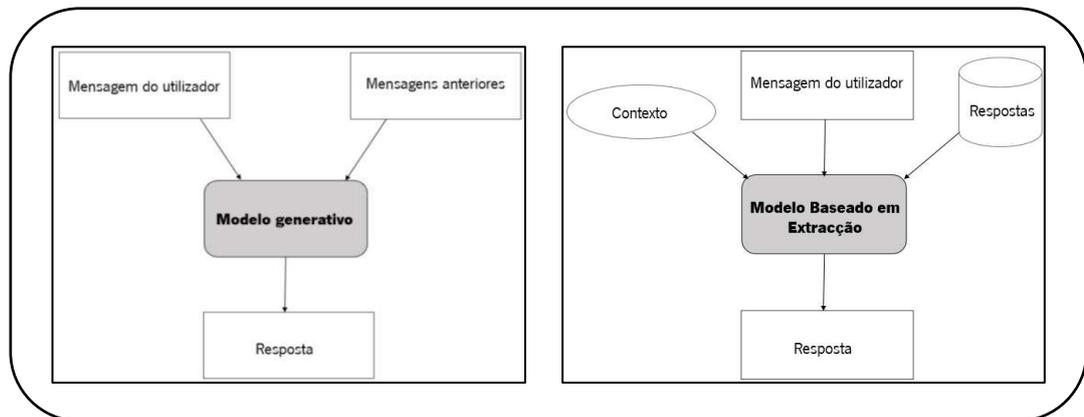
como a utilização de algoritmos de classificação baseados em métodos de Machine Learning (GEHREKE, 2017).

Uma das dificuldades encontrada para utilizar este modelo está no fato antecipar todos os possíveis casos que serão transmitidos pelo chatbot. Este tipo de abordagem precisa de conjuntos de casos que permitam responder a todas situações do domínio. O modelo baseado em extração da informação se aplicar a chatbot de domínio fechado(CAHN, 2017).

O benefício do modelo de recuperação de informação está no fato de conseguirmos controlar o que se encontra na sua base de conhecimento evitando assim a ocorrências de erro gramaticais e de más construções frasais, garantindo qualidade nas respostas produzidas pelo sistema (GEHREKE, 2017).

No entanto, o modelo generativo não usa repositórios de respostas predefinidas, este modelo usa o processo de aprendizagem de máquinas para responder as interações com o usuário (JUNIOR, 2017). Em decorrência do exposto , a figura-2 ilustra a estrutura presente em cada tipo de sistemas apresentado.

Figura 2: Estrutura dos modelos generativos e baseados em extração, passíveis de serem implementados num Chatbot.

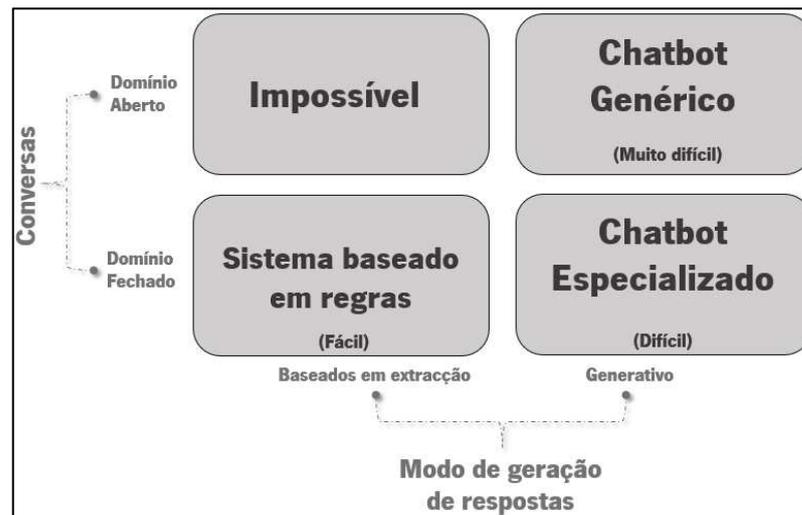


Fonte: Dias (2018).

Outra distinção que são inerentes ao *chatbot* é quanto ao seu objetivo, que pode ser um *bot* conversacional ou orientado a tarefa. Os *chatbots* conversacionais estão focados em ter uma conversa com o usuário com o intuito de entretenimento (JUNIOR, 2017)(CAHN, 2017). Enquanto os sistemas orientados a tarefa estão focados em ajudar o usuário a realizar uma atividade, como por exemplo: comprar uma passagem aérea (FERREIRA, 2006).

Visto que, cada arquitetura possui um grau de dificuldade associado à sua implementação, sendo ilustrado na figura 3, as possíveis combinações a serem implementadas num *chatbot* com base no seu objetivo: conversacional ou orientado a tarefa.

Figura 3: Graus de dificuldade associados ao desenvolvimento dos diferentes tipos de sistemas conversacionais.



Fonte: Dias (2018)

2.2. Estado da Arte dos Chatbot's

Os sistemas conversacionais começaram a ser desenvolvidos por volta de 1950 com os estudos de Alan Turing. Os primeiros modelos de sistemas conversacionais eram baseados em palavra simples, baseando-se no reconhecimento de padrões e de certas palavras chaves para a geração da resposta a ela associada, (GUO, 2018). Neste capítulo será apresentado os primeiros chatbots ELIZA, JULIA e ALICE.

2.2.1. Teste de Turing, Quarto Chinês.

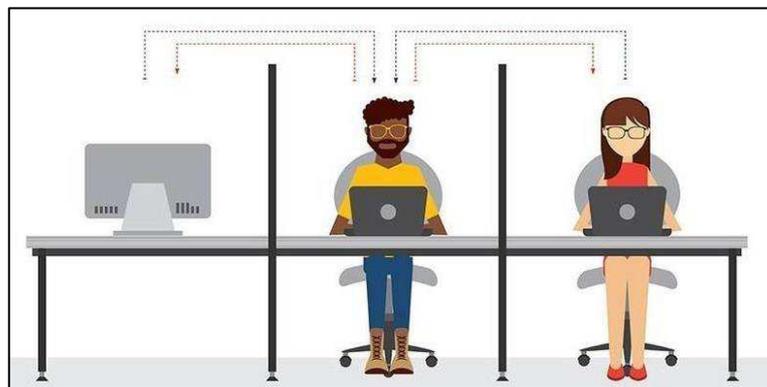
Em seu artigo semanal sobre inteligência artificial, Alan Turing, propõe um teste simples para avaliar se uma máquina exibe comportamento inteligente (OLIVEIRA, 2019).

O Teste de Turing, envolve três participantes – dois humanos e um computador – no qual, um dos seres humanos possui o papel de interrogador, com o objetivo de determinar qual dos outros dois participantes é o computador (SPANHOL, 2017).

Podemos pensar que, se caso o interrogador não consiga diferenciar as repostas que são dadas pelo computador e o participante humano, então podemos afirmar que “as máquinas são capazes de pensar”. A figura 5 ilustra o funcionamento da máquina de Turing.

Russel e Norvig na sua obra, afirmam que o Teste de Turing é útil para identificar uma forma operacional de inteligência, isto é, não se preocupa com o mecanismo utilizado, mas sim com o seu resultado (NORVIG; RUSSELL, 2010).

Figura 4: Funcionamento da máquina de Turing.



Fonte: <https://filosofianaescola.com/metafisica/teste-de-turing/>

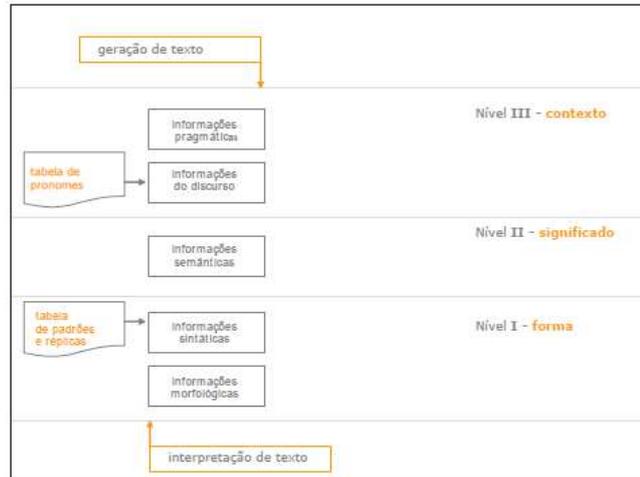
O teste de Turing iniciou uma corrida para a criação de sistemas que conseguissem confundir árbitros. O primeiro sistema conversacional criado com esta finalidade foi o Eliza, desenvolvido em 1966 pelo professor Joseph Weizenbaum (1966) em Massachusetts Institute of Technology (MIT) (CARES, 2018).

Para a construção do *chatterbot* Eliza foi utilizado o mecanismo de correspondência de padrões e técnicas de substituição de caracteres. Ao receber uma entrada do usuário o sistema faz a busca por termos chaves que foram previamente definidos no código fonte do programa e são considerados importantes no contexto da conversa (SPANHOL, 2017).

De acordo com Kriz, (2017), a arquitetura do sistema ELIZA utiliza pares de valores chaves para obter os padrões e classificar as sentenças do interlocutor e gerar outras sentenças, como mostrado na figura 5.

Já na figura 6, ilustra um trecho de uma conversa com Eliza, traduzido para o português, no qual o usuário conta como está se sentindo.

Figura 5: Arquitetura do ELIZA.



Fonte: Neves (2005).

Figura 6: Conversa com Chatterbot Eliza.

```

 Bem-vindo ao
 EEEEE LL IIII ZZZZZZ AAAAA
 EE LL II ZZ AA AA
 EEEEE LL II ZZZ AAAAAA
 EE LL II ZZ AA AA
 EEEEE LLLLL IIII ZZZZZZ AA AA

 ELIZA é uma psicoterapeuta falsa (rogeriana).
 O programa original foi descrito por Joseph Weizenbaum em 1966.
 Esta implementação ('elizabot.js') por Norbert Landsteiner 2005.
 Integração de gráficos e texto em fala em tempo real adicionada em 2013.

 **** Digite ou clique no botão do microfone abaixo para entrada de voz. ****

 ELIZA: Como você está? Por favor, diga-me o seu problema.
 VOCÊ: estou triste
 ELIZA: Conte-me mais sobre isso.
 VOCÊ: não consigo dormir
 ELIZA: Por favor, continue.
 VOCÊ: passo a noite em claro
 ELIZA: Conte-me mais sobre isso.
 YOU:
  
```

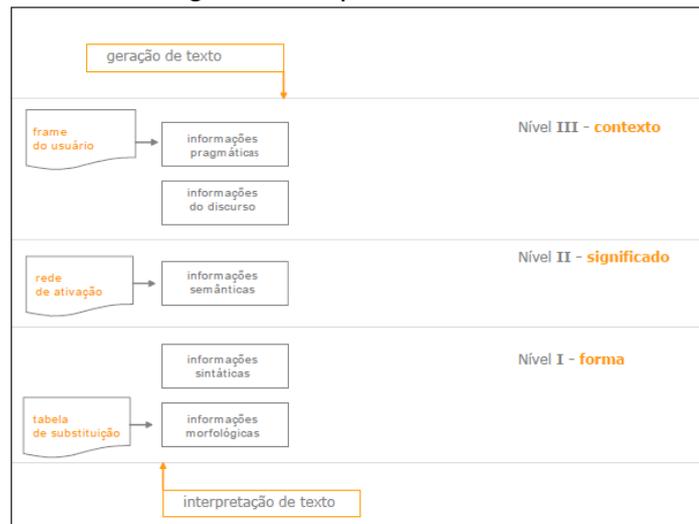
Fonte: <http://www.masswerk.at/elizabot/>

Outro sistema conversacional notório desenvolvido foi JULIA, este foi criado por Michael Mauldin, em 1994 (NEVES, 2005). De acordo com o que está descrito em (PEREIRA, 2018), chatterbot JULIA é um TinyMUD, ou seja, um personagem dentro de um jogo de computador multiusuários. Sua função é emular um jogador que auxilia os demais jogadores mapeando cavernas, indicando direções e os melhores caminhos a serem tomados, além de fornecer informações sobre outros jogadores, ambientes e objetos.

A arquitetura da Julia é formada por três módulos: Frames do usuário que são as entradas fornecidas pelo usuário, redes de ativação e conhecimento

sobre o t3pico e por 3ltimo a tabela de substitui33o que trabalha com as varia333es morfol33gicas e sint33ticas (PEREIRA, 2018). A figura 7 descreve a arquitetura do sistema Julia; e na figura 8 visualizamos um trecho de uma conversa de uma conversa com Julia, traduzida para o portugu33s, no qual o usu33rio pergunta sobre sua apar33ncia

Figura 7: Arquitetura do Julia.



Fonte: Neves (2005).

Figura 8: Trecho de conversa com Julia

```
VOC33: Julia, quantos anos voc33 tem?
JULIA: Eu tenho 21 anos.
VOC33: Julia, fale um pouco sobre voc33.
JULIA: Eu tenho 1,55m de altura e peso 55kg. Tenho cabelos loiros e curtos e olhos castanho-escuros.
VOC33: Julia, qual 33 a cor dos seus olhos?
JULIA: Eu tenho olhos castanho-escuros.
```

Fonte: Neves (2005).

Por fim, podemos citar o chatbot A.L.I.C.E, que foi desenvolvido em 1995 pelo Dr. Wallace, um chatterbot implementado em AIML (Artificial Intelligencia Language) (KRIZ, 2017). Este sistema opera com um modelo de aprendizagem supervisionado, ou seja, um “Treinador” que monitora as conversas do chatterbot e cria um 33ndice de AIML, a fim de obter respostas mais apropriadas (KRIZ, 2017).

De acordo com Neves (2005), a arquitetura do A.L.I.C.E 33 formada por tr33s n33veis: n33vel de contexto, n33vel de significado e n33vel de forma. O n33vel de contexto analisa o contexto do di33logo do sistema e faz a an33lise morfol33gica da entrada; no n33vel de significado 33 analisado o sentido de cada entrada para o

casamento de padrão fornecido pelo AIML para entregar uma resposta para o usuário; e no nível de forma é buscado uma saída conforme o padrão de entrada que se encontra dentro das categorias do AIML.

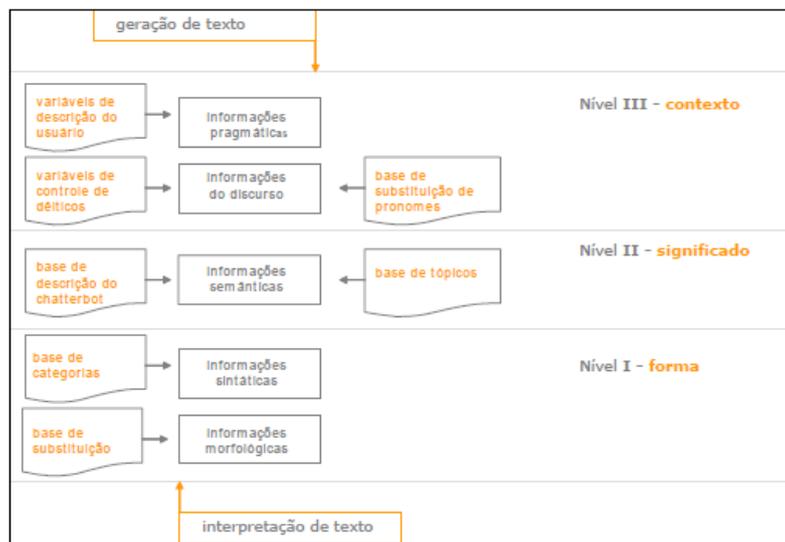
Na figura 11 é possível visualizar a arquitetura do sistema A.L.I.C.E.; já a figura 10 mostra um trecho de uma conversa com ALICE, traduzida para o português.

Figura 10: Trecho de conversa com Alice

ALICE: Olá, meu nome é Alice e eu sou humana.
 JUIZ: Olá Alice, como você está?
 ALICE: Olá. Tudo está indo muito bem.
 JUIZ: Estou contente de estar aqui. Onde você mora?
 ALICE: Obrigada pela informação. Estou localizada em Nova York. Onde você está?
 JUIZ: Em Nova York. Como você chegou aqui hoje?
 ALICE: Conte-me um pouco sobre sua vida em Nova York. Ah! Tudo que eu preciso saber eu aprendi com meu instrutor.

Fonte: Traduzido de Kriz (2017).

Figura 11: Arquitetura do sistema A.L.I.C.E



Fonte: Neves, (2005).

Os fundamentos de *chatbot* e o estado da arte foram apresentados nesta seção, além de apresentar os diferentes tipos de *chatbot* e domínios que eles podem ser construídos. Os temas abordados nesta seção constituem a base teórica necessária para a compressão desta dissertação.

3. TECNOLOGIAS PARA O DESENVOLVIMENTO DE UM CHATBOT.

Os *chatbots* têm sido empregados nas áreas da educação, médica, comercial, marketing e atendimento. Para atender a demanda do mercado, empresas como Google, Amazon, Microsoft, IBM e entre outras, têm desenvolvido mecanismos para a criação de *bot* de maneira eficiente e com menor tempo de desenvolvimento.

Os *chatbots* precisam realizar o processamento de linguagem natural para extrair informações das intenções dos usuários a partir das entradas fornecidas. Outra parte crucial para a construção do *chatbot* é sua base de conhecimento. Artificial Intelligence Markup Language (AIML) foi um dos primeiros mecanismos utilizados para a construção da base de conhecimento do *chatbot*, como é caso do *bot* ALICE, citado na seção 2.1 deste trabalho. Para fins deste trabalho, iremos compreender os conceitos sobre Processamento de Linguagem Natural (PLN) e AIML.

3.1 Processamento de Linguagem Natural (PLN)

A nossa comunicação interpessoal é realizada de diversas maneiras: através da fala, da audição, por comunicação gestual, pela realização de sinais ou através de formas textuais. Sobre comunicação textual, referimo-nos a um conjunto de palavras escritas em qualquer superfície ou exibidas em algum dispositivo eletrônico, tendo em vista a sua transmissão para o receptor (DIAS, 2018).

Desde a década de 1950, um dos problemas mais recorrentes para a ciência da computação tem sido a interpretação e a geração de texto em linguagem natural (NEVES, 2005).

Neste contexto, a PLN possibilita que os seres humanos se comuniquem com os computadores de uma forma mais natural, utilizando sua língua nativa. O seu uso permite que os usuários não precisem aprender uma linguagem artificial para realizar a interação com o computador, a qual a sintaxe pode ser um pouco mais difícil, como as linguagens de consulta de banco de dados (JUNIOR, 2017).

A PLN é um campo da ciência da computação que estuda formas de compreender um texto, de extrair informações específicas dele e até de gerar um novo texto (CAHN, 2017).

De acordo com Dias (2018), o processo de compreensão da linguagem natural, divide-se de uma forma comum, em 3 passos distintos, discutidos e apresentados ao longo desta seção: Processamento Morfológico, Processamento Sintático e Processamento Semântico.

3.1.1 Processamento Morfológico

O processamento morfológico trata as palavras quanto à sua estrutura, forma, flexão e classificação, que correspondem às estruturas estáticas (radicais) e variantes (afixos). Para o desenvolvimento de *chatbot*, o processamento morfológico é importante, pois pode resultar em um número bem menor de regras necessárias para realizar o casamento de padrão (JUNIOR, 2017).

As técnicas de processamento morfológico que iremos discutir nesta seção são: Tokenização, Transforme Cases, Remoção de Non Letters, Remoção de Stop Words, Stemming Remoção de Afixo, Busca em Tabela, Variedade de Sucessores, N-Gramas (SHAWAR, 2014).

Tokenização (Tokenization): Essa técnica permite que o texto seja dividido em unidades linguísticas identificáveis, que constituem parte dos dados de uma língua, chamados de tokens.

A tokenização permite a identificação de palavras sem significado num determinado texto, fornecendo, assim, ao seu utilizador, a capacidade de remover componentes não desejados por este, reduzindo desta forma a quantidade de dados sem relevância, a serem processados (DIAS, 2018).

Nesta primeira fase do processo PNL ainda não é possível extrair informações, pois estamos separando os termos de uma frase, por exemplo: “*Quem comeu o meu queijo?*”, nesta frase encontramos os tokens: *QUEM COMEU O MEU QUEIJO*.

Transforme Casos (Transforme Cases, lower cases): A técnica lower case é utilizada para transformar todas as palavras ou tokens em letras

minúsculas, de forma que não sejam reconhecidas como tokens diferentes (SANTOS, 2015).

Remoção do que não é letra (Non Letters): A remoção de non letters tem como objetivo eliminar todos os tokens que não sejam palavras. Geralmente são removidos a pontuação, caracteres alfanuméricos e caracteres especiais (SANTOS, 2015).

Remoção de palavras repetidas (Stop Words): Este método consiste em eliminar as palavras que ocorrem com maior frequência em um texto, devido ao fato destas palavras não serem úteis na classificação e extração de significado de um dado contexto.

As stop word são chamadas de não discriminantes, pois servem como elemento de ligação entre os elementos morfológicos de uma frase. Os candidatos a não discriminantes são: artigos, preposições e conjunções (PINTO, 2015) (SANTOS, 2015).

Com a remoção das stop word é possível reduzir o tamanho do conjunto de dados que são relevantes para a compreensão do contexto de domínio.

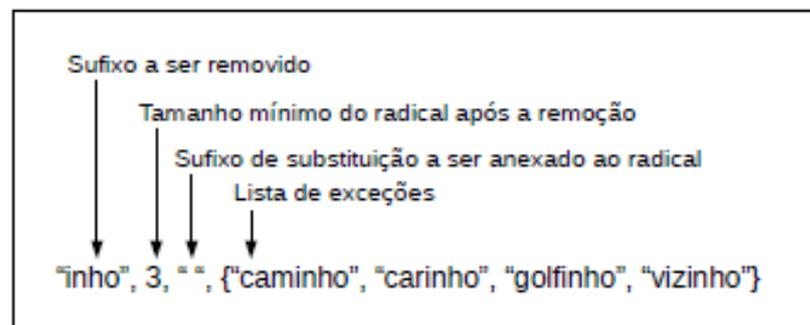
Stemming: Stemming diz respeito a uma técnica de análise lexical, utilizada para diminuir a lista de palavras indexadas num dado documento. Este método permite reduzir palavras que se encontram em formas derivadas para a sua forma base, stem, removendo as variações de palavras do tipo (plural, gerúndio, prefixos, sufixos, género e número) de modo que a palavra fique só com a raiz (DIAS, 2018).

Assim como a técnica Stop words, a Stemming tem a finalidade de reduzir a quantidade de termos que são indexados e melhora a relevância dos resultados no reconhecimento de termos. Podemos observar as variações da palavra estudar (estudou, estudava e estudei) ao passar pela técnica de Stemming será reduzida a sua forma base (DIAS, 2018).

Remoção de Afixo: Esta técnica é utilizada para remover os sufixos das palavras, porque muitas variações de uma palavra são geradas pela introdução de sufixo, em vez de prefixo. Está técnica é utilizada no algoritmo de

Porter, que utiliza uma lista de sufixos para o descarte de sufixos das palavras. Outro algoritmo utilizado para remoção de sufixos é o Removedor de Sufixo da Língua Portuguesa – RSPL, que tem como objetivo remover sufixos para a língua portuguesa, que baseado em regras, e cada uma delas pode ser expressa conforme a figura 14, abaixo:

Figura 9: Regra do Algoritmo RSPL Stemmer.



Fonte: Ticom (2017).

Busca em Tabela: A ideia deste método é possuir um dicionário com todas as palavras de determinada língua e seus respectivos radicais, este método possui a vantagem tornar a procura por padrões de entrada de maneira mais rápida, mas por outro lado, possuem a desvantagem de necessitarem de uma capacidade de armazenamento que acompanhe o número de stems pretendidos (PINTO, 2015). A figura 10 exemplifica um modelo de dicionário.

Figura 10: Abordagem em busca em Tabela.

Palavra	Stem
computadores	comput
computador	comput
computar	comput

Fonte: Pinto (2015).

Variedade de sucessores: Necessita de uma coleção de documentos definidos antes do processamento das palavras (PINTO, 2018). Este método foi proposto por Hafer e Weiss, em 1974, após uma rebuscada

análise da frequência e do tamanho dos prefixos ao longo das variações morfológicas de uma palavra, identificar heurísticamente o seu potencial radical (DIAS, 2018).

N-Gramas: Os n-gramas são sequências de conjuntos de palavras, de tamanho especificado pelo desenvolvedor, que representam a frase original, através de pedaços da mesma (CAHN, 2017). Após a divisão da palavra é calculado um coeficiente sobre o número de n-gramas gerados e então gerada uma matriz de similaridade entre todos os pares possíveis de todas as palavras processadas.

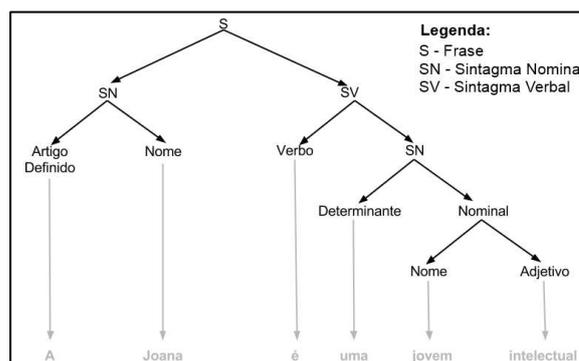
3.1.2 Análise Sintática

O processamento sintático se encarrega de verificar as relações formais entre as palavras, ou seja, qual é o papel de cada palavra numa frase e a sua correlação com as demais palavras vizinhas. Na Língua Portuguesa, um pronome pessoal é, geralmente, seguido de um verbo (JUNIOR, 2017).

Dias (2018), resume a análise sintática como um conjunto de regras estruturais utilizadas com técnicas de parsing (análise/geração de frases, de acordo com a sua gramática) é capaz de definir a relação entre os termos presentes em uma frase, podendo estas representarem um adjetivo, um verbo ou um substantivo.

Pinto (2015), demonstra que o processamento sintático permite construir uma árvore de constituintes que tem por objetivo demonstrar a relação entre os termos da frase. A figura 11 ilustra a relação entre os termos de uma frase.

Figura 11: Arvore de constituintes.



Fonte: Dias (2018).

3.1.3 Análise Semântica

As análises anteriores, processamento morfológico e análise sintática, são etapas importantes no processo para a interpretação e classificação de frases, no entanto, entender o significado de frase é primordial para implantar sistemas de conversão que utilizam processamento de linguagem natural.

Entender o significado de uma frase não é uma tarefa fácil para o computador, pois o sentido das palavras pode ser interpretado de forma ambígua. Por exemplo, a palavra *banco*: “*Eu irei ao banco*”, “*Eu vou me sentar no banco*” (CAHN, 2017).

Para melhorar o entendimento da palavra empregada em contextos diferentes são utilizadas técnicas de análise semântica para reduzir a saída de significados errôneos devido ao contexto (DIAS, 2018). Os métodos mais utilizados na análise semântica são:

Extração de Informação: Representar a informação não estruturada de uma forma organizada, de maneira que possa depois ser utilizada em métodos de Machine Learning e Deep Learning, bem como em motores de pesquisa (DIAS, 2018).

Desambiguação de sentido das palavras (Word Sense Disambiguation - WSD): Desambiguação de sentido das palavras (WSD) é a capacidade de identificar o significado das palavras no contexto de uma maneira computacional (TICOM, 2007).

Análise Semântica (Lenten Semantic Analysis - LSA): É o método para extrair e representar o significado de uso contextual das palavras por cálculos estatísticos aplicados a um grande corpus de texto (PINTO, 2018).

Rotulagem de função semântica (Semantic Role Labelling - SRL): As técnicas de SRL dizem respeito aos processos capazes de rotular os elementos de uma frase, de acordo com o seu papel semântico, isto é, são capazes de identificar quem é o sujeito, qual o objetivo, o resultado, entre outros, dependendo do domínio e do contexto em que estes se encontram inseridos, (ALVARES, 2005).

Mecanismos de análise de questões: A análise de uma questão se inicia com o recebimento por parte do sistema, da questão enviada pelo

utilizador, sob a forma de texto não estruturado, começando desta forma, por identificar de forma sintática e semântica, os elementos da questão (DIAS, 2018).

Processamento do Discurso: Essa etapa tenta identificar qual é a influência de uma ou mais frases na interpretação das frases subsequentes, ou seja, ela extrapola a estrutura da frase que está sendo verificada atualmente pelo processador (JUNIOR, 2017).

Processamento Pragmático: O processamento pragmático verifica as relações entre as mensagens e os sujeitos, ou seja, analisa a intenção de quem fala e o reconhecimento desta intenção pelo ouvinte, levando em consideração o contexto em que o emissor e o ouvinte estão inseridos (JUNIOR, 2017).

3.1.4 Áreas de Aplicação do Processamento de Linguagem Natural

O processamento de linguagem natural possibilita desenvolver ferramentas para diversas aplicações em que o entendimento da linguística se faz necessário para um computador. Algumas aplicações que utilizam PNL (PINTO, 2015):

Corretores ortográficos e gramaticais: Este tipo de aplicação são módulos que fazem parte de editores de textos, com a finalidade de auxiliar o usuário na correção ortográfica e gramatical, que verifica se cada palavra pertence ao vocabulário da linguagem e verifica algumas construções gramaticais das frases.

Tradutores automáticos: Os tradutores automáticos são utilizados para realizar a tradução de textos de uma língua natural para outra (e.g., inglês, espanhol, francês). Existem vários sistemas ou sites disponíveis na internet para tradução automática, como o Google Translate e o Babylon.

Interface em Linguagem Natural: São aplicações que utilizam como entrada de dados a linguagem falada ou escrita, traduzindo as instruções apresentadas em linguagem específica do sistema de gerenciamento. Este tipo de aplicação é denominado sistema de perguntas e repostas que são relativos de aplicação específica e limitada, muitas vezes delimitando-se a interações de palavras-chaves. A interface de linguagem natural é método utilizado para a construção de *Chatbot* pelas empresas.

3.2. ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE-AIML

AIML é uma linguagem XML para especificar conteúdo de uma conversa para *chatbot*. Está foi desenvolvida pelo Doutor Richard Wallace, entre 1995 e 2002, utilizada para construir a base de conhecimento do *chatbot* A.L.I.C.E (BIRD, 2009).

O AIML é separado em Interpretador e Base de Conhecimento. O interpretador pode ser encontrado em diferentes linguagens de programação, como exemplo: C++, Python, Java e Ruby (SHAWAR, 2016). O interpretador é o responsável por ler os arquivos AIML, receber as entradas do usuário e devolver as respostas. A base de conhecimento, codificada em arquivos AIML, é utilizada para armazenar as regras de casamento de padrões do *chatterbot* (WALLACE, 2014).

Nos arquivos AIML, as categorias (regras) são as unidades básicas do conhecimento. Cada categoria é constituída por um *componente de decomposição* da sentença digitada pelo usuário e por componentes de reconstrução da sentença devolvida pelo *chatbot* sendo denominada, respectivamente, de *pattern* e *template* (WALLACE, 2014). As categorias <category> básicas AIML são compostas por:

1. Padrão <pattern> - utilizado para casar-se com entradas do usuário;
2. Modelo < Template>- usado para dar a resposta para o usuário; e
3. Contexto representado pela tag<that> e <topic> - utilizadas para tratar as réplicas do usuário e agrupar categorias que tratam de assuntos correlatos em uma conversa.

Na figura 12 é possível observar a construção de uma categoria no AIML, dentro da tag categoria temos o padrão de entrada <pattern> que é fornecido pelo usuário, o interpretador AIML ao encontrar respectiva entrada, devolve uma resposta que está na tag <template>.

Figura 12: Categoria AIML.

```

<category>
<pattern>Você está bem </pattern>
<template>sim, estou</template>
</category>

```

Fonte: Wallace (2014).

Como exposto, cada categoria AIML é especificada exclusivamente por padrão de entrada e padrão de saída. O AIML especifica um caminho padrão, que é sequência de vinculada de nós, onde cada nó representa um padrão de entrada e padrão de saída. A figura 13 abaixo ilustra o caminho ligado por nós que o interpretador AIML percorre para casar um padrão de saída.

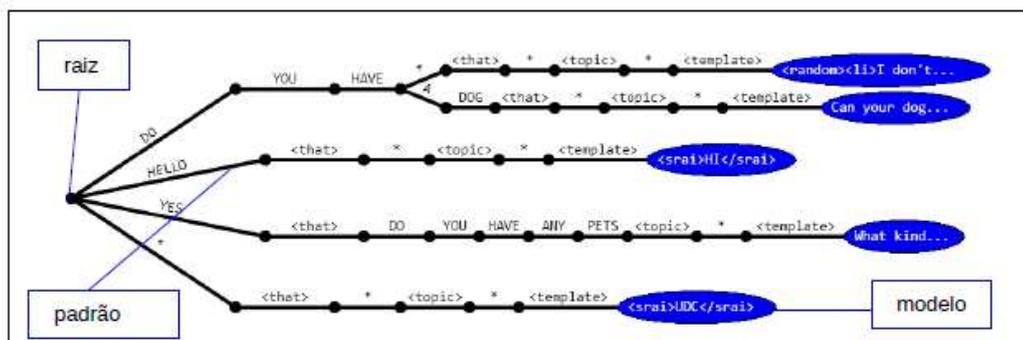
Figura 13: Caminho padrão para uma categoria.



Fonte: Wallace (2014).

O Interpretador AIML cria um objeto chamado Graphmaster que carrega os arquivos AIML (BIRD, 2009). No Graphmaster, é inserido um caminho para cada padrão (<pattern>) associado às categorias existentes nesses arquivos. Cada caminho introduzido possui uma direção e uma raiz. No final de cada caminho do Graphmaster, existe um link para o modelo (<template>) (WALLACE, 2014). A Figura 14 ilustra o objeto Graphmaster criado Wallace (2014).

Figura 14: Exemplo do Objeto Graphmaster



Fonte: Wallace (2014).

Dada uma mensagem do usuário para o *chatbot*, o interpretador AIML constrói um caminho de entrada, semelhante a um caminho padrão contendo entradas normalizadas, com última resposta do bot. O algoritmo de correspondência de padrões AIML procura no Graphmaster uma correspondência do caminho de entrada (WALLACE, 2014). A pesquisa prossegue em uma busca profunda pela correspondência, porém quando a pesquisa, em uma ramificação do gráfico, falha em encontrar uma correspondência, o algoritmo de pesquisa retorna ao último nó da ramificação inexplorada da pesquisa (WALLACE, 2014).

De acordo com Wallace (2018), a sequência de pesquisa em cada nó é feita pela seguinte sequência:

1. \$palavra – A palavra que tem o símbolo cifrão a antecedendo tem maior prioridade para casamento de padrões.
2. # - O curinga hashtag representa o casamento de zero ou mais palavras.
3. _ - O curinga de sublinhado corresponde ao casamento de uma ou mais palavras.
4. \$ Palavra – casamento com palavra exata.
5. <set> nome </set> - correspondência encontrada no conjunto AIML.
6. ^ - O curinga circunflexo representa o casamento de zero ou mais palavras.
7. * - O curinga asterisco representa o casamento de uma ou mais palavras.

Se nenhuma correspondência for encontrada no graphmaster, o interpretador deve retornar uma resposta padrão, como: “*não tenho resposta para isso*”, especificada pelo botmaster, este é a pessoa que constrói os diálogos AIML.

Como mencionado, as categorias AIML constituem a base de conhecimento do bot e estão divididas em três tipos (SHAWAR, 2014):

1. Categorias atômicas: são os padrões que não possuem símbolos curinga (*).

2. Categorias padrões: são os padrões que possuem símbolo curinga. Estes correspondem a qualquer entrada, mas diferem em sua ordem alfabética. Assumindo a categoria anterior, se bot não encontrar o padrão atômico anterior, ele tentará encontrar padrão seguinte.

A figura 15 exemplifica uma categoria AIML com uso de caractere curinga.

Figura 15: Exemplo de categoria padrão.

```
<category>
<pattern>Você está * </pattern>
<template>sim, estou</template>
</category>
```

Fonte: baseado em Shawar (2014).

3. Categorias recursivas: São as categorias que contém as tags <srai> e <sr>, que se referem a redução simbólica. Este método reduz as palavras para a sua forma básica (infinitivo). Outra aplicação da tag <srai> é a divisão das sentenças, por exemplo, dividir uma sentença maior em duas subsentenças, a resposta é a combinação das novas sentenças criadas. Além disso, a tag <srai> pode ser empregada na correção ortográfica e gramatical das entradas do usuário. Na figura 16 exemplifica o emprego de redução simbólica que é chamar uma categoria com o padrão contido na tag <srai> para dar a resposta ao usuário de acordo com padrão fornecido.

Figura 16: Redução Simbólica

```
<category>
<pattern>Você sabe quem * é</pattern>
<template><srai>Quem é <start/></srai></template>
</category>
```

Fonte: Autor (2021).

3.2.1 Chatterbot criados com AIML.

A linguagem AIML tem sido bastante utilizada para o desenvolvimento de *chatbot*, devido a facilidade de aprendizagem da linguagem e o interpretador

AIML pode ser encontrado em diversas linguagens de programação. Na seção 3.1, evidenciamos alguns trabalhos que foram construindo utilizando AIML:

Em Neves (2005), foi criado um *chatbot* com objetivo de responder perguntas sobre epilepsia. Sua base de dados foi preenchida com informações básicas da doença, como: os conceitos, causas e diagnósticos. Seu desenvolvimento foi feito utilizando a ferramenta Pandorabots.

Já em Ferreira (2006), encontramos o *chatbot* Esteban para auxiliar no ensino da língua estrangeira espanhola. Esteban possui sintetizador de voz para melhorar a interação com o usuário e foi desenvolvido na linguagem Python. Em Gehrke (2017), encontramos o *chatbot* CHATTEREDU para auxiliar no ensino de disciplina de geografia desenvolvido na linguagem java.

Neste capítulo foi abordado as tecnologias para desenvolvimento de um *chatbot*, tais como: Processamento de linguagem Natural – PNL e AIML que é a principal tecnologia utilizada para o entendimento desta dissertação.

4. REVISÃO SISTEMÁTICA DA LITERATURA

Para complementar este trabalho, foi examinado propostas semelhantes de forma a analisar as diferenças que podem agregar pesquisas futuras.

As literaturas examinadas mostram que há um crescente aumento no emprego de Tecnologias da Informação na área de saúde, que auxiliam médicos, enfermeiros e farmacêuticos a serem mais assertivos nos resultados que pretendem alcançar.

Uma área que se mostrou com grande interesse para os pesquisadores, é o desenvolvimento de Tecnologias da Informação que possa apoiar os idosos nas suas atividades diárias, por exemplo: ferramentas que auxiliem o idoso na gestão dos medicamentos que precisam ser tomados ao longo do dia.

Os trabalhos relacionados abordam o uso de tecnologia da Informação para auxiliar o idoso a administrar os medicamentos que eles necessitam. Os trabalhos estão em seções, como segue:

4.1 Protótipo de Aplicação Android para Controle da Rotina De Medicamentos para Idosos.

Inacio (2014), descreve uma ferramenta para auxiliar os idosos a ter gestão dos medicamentos que são necessários tomar ao longo do dia. Este trabalho foi desenvolvido por um protótipo funcional seguindo os critérios Web Content Accessibility Guidelines (WCAG) 2.0, de usabilidade para idosos e para aplicação mobile.

Na figura 22 é possível visualizar as duas funções do Protótipo de Aplicação Android para Controle da Rotina De Medicamentos para Idosos, na tela inicial temos: o botão para criar um alarme que leva para uma tela para cadastrar o nome do medicamento e horário de ingestão do medicamento. Já na opção lista, o alarme irá mostrar os medicamentos cadastrados e o horário configurado para o alarme (INACIO, 2014).

A aplicação apresentada não possui a funcionalidade para cadastro de consulta, não possibilita a interação com o usuário e não implementa nenhuma técnica de inteligência artificial para criar uma base de conhecimento

que permita que a aplicação se comunique com o usuário de maneira mais intuitiva.

Figura 17: Protótipo de Aplicação Android para controle da rotina de Medicamentos para Idosos.

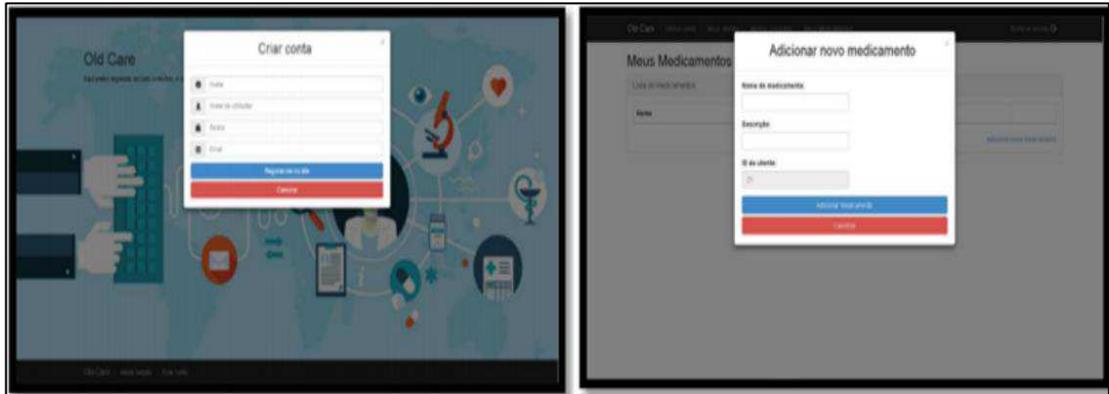


Fonte: Inacio (2014).

4.2 Aplicação Web para o Auxílio da Toma de Medicamentos e de gestão de Informação de Consultas Médicas Para Idosos-Old Care.

Aplicação Web para o auxílio da tomada de medicamentos e de gestão de informação de consultas médicas para idosos - Old Care é uma solução que pode ser acessada via web browser pelo usuário [28]. Ao acessar a interface do Old Care será solicitado ao usuário idoso seus dados pessoais para cadastro na ferramenta e a criação de um login e senha para acessar as demais funções do software (ROCHA, 2018). Ao acessar o Old Care o idoso poderá cadastrar os medicamentos que necessitam ser tomados ao longo do dia e inserir os dados das consultas médicas. Na figura 18 é ilustrado a tela da seção inicial e a tela de cadastro de medicamentos do Old Care.

Figura 18: Aplicação Web para o auxílio da toma de medicamentos e de gestão de informação de consultas médicas para Idosos.

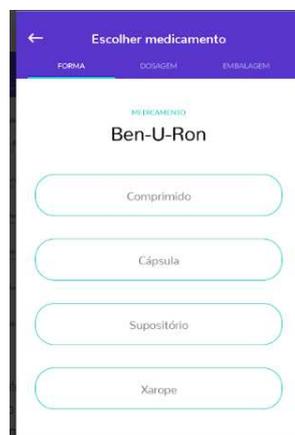


Fonte: Rocha (2018).

4.3 EMED

Esta solução foi desenvolvida para auxiliar o usuário na gestão dos medicamentos que ele deve tomar ao longo do dia. Além de auxiliar no gerenciamento do medicamento, o EMED realiza a leitura dos códigos de barras das caixas de medicamentos e informa aos usuários opções similares do medicamento a ser tomado e as possíveis farmácias, nas quais ele pode ser encontrado. O EMED não possibilita o usuário criar um lembrete das consultas médicas. Abaixo é demonstrado, na figura 19, a tela para escolher o medicamento a ser cadastrado no sistema EMED

Figura 19: Aplicação Web para o auxílio da toma de medicamentos e de gestão de informação de consultas médicas para idosos

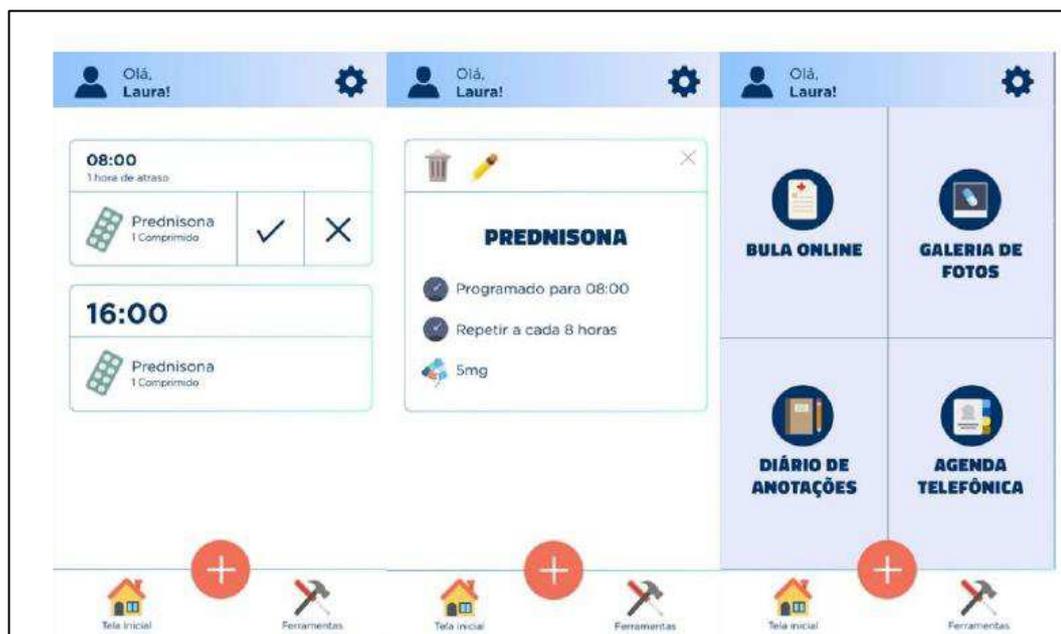


Fonte: Neves (2005).

4.4 Medamigo: Um Aplicativo de Auxílio a Idosos a Ingestão de medicamentos

Aplicativo de auxílio a idosos na ingestão de medicamentos – MedAmigo, propõe uma ferramenta que seja simples de ser utilizada pelo público alvo: idosos (IGARASHI, 2019). Na ferramenta, denominada de MedAmigo, foi implementada as funcionalidades de cadastrar o medicamento, assim como a sua foto, o horário do alarme para o medicamento, repetir em intervalos definidos pelo usuário. Este sistema traz uma funcionalidade nova em relação aos que foram apresentados, a consulta online da bula e agenda telefônica, que pode conter o contato do cuidador do idoso (IGARASHI, 2019). O MedAmigo não permite criar alarmes para as consultas que o idoso irá fazer, do mesmo modo que ele não grava o local da consulta. . A figura 20 demonstra a tela do MedAmigo.

Figura 20: MedAmigo.



Fonte: Igarashi (2019).

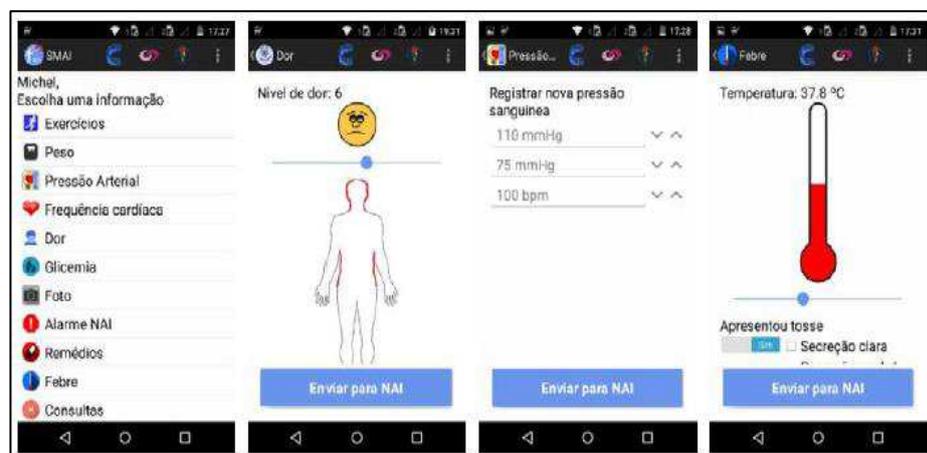
4.5 SMAI - Sistema Móvel de Assistência ao Idoso

Sistema Móvel de Assistência ao Idoso - SMAI é um software que integra o serviço do profissional de saúde com o paciente idoso, através de informações coletadas a partir dos dados inseridos pelo usuário no SMAI (STUZEL, 2015). O SMAI possui dois módulos, um chamado SMAI médico, para

acompanhamento do idoso pelos profissionais de saúde; e o outro chamado de SMAI web, que é utilizado pelo usuário.

O SMAI oferece diversas funcionalidades para melhorar a vida do idoso, tais como: lembrete sobre o uso de medicação e consultas, informações contendo medidas fisiológicas, como a pressão arterial, envio da localização do paciente, relatório diário sobre o paciente (STUZEL, 2015). Na figura 21 podemos visualizar a tela principal do SMAI.

Figura 21: SMAI.



Fonte: *Stuzel (2015)*.

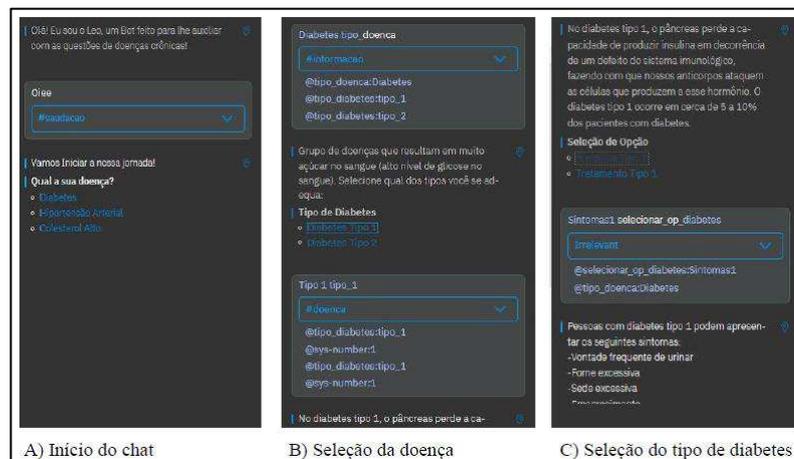
4.6 Helpcare: Um Protótipo de Chatbot para O Auxílio Do Tratamento de doenças Crônicas.

Diferentes dos outros trabalhos apresentados, Oliveira (2018) propõe um sistema de conversação para auxiliar usuários com doenças crônicas, como diabetes, colesterol alto e hipertensão arterial, a tomar os medicamentos que são importantes para que o paciente tenha uma vida saudável.

O helpcare tira dúvidas sobre a doença crônica que o usuário é portador, avisa horários das medicações, cadastra a finalidade dos remédios, auxilia na dieta e lista os hospitais mais próximo do local onde reside o paciente, para emergências (OLIVEIRA, 2018).

O helpcare foi desenvolvido para utilizar a plataforma do telegram para troca de mensagens com o usuário. Na figura 22 podemos visualizar a interação do usuário com o bot.

Figura 22: Helpcare.



Fonte: Oliveira (2018).

4.7 Roborto

Roborto é *chatbot* para auxiliar no gerenciamento dos medicamentos que devem ser tomados por paciente com doenças crônicas (FADHIL, 2018). O *chatbot* permite cadastrar os medicamentos por nome, ícone ou foto. A partir do cadastro do medicamento o bot define um lembrete sobre o tempo e a quantidade da medicação (FADHIL, 2018). Além da medicação, o bot pergunta aos usuários sobre os sintomas e seu nível de estresse e padrão de sono.

O usuário do Roborto pode registrar atividades físicas, dietas e cadastrar consultas médicas. Na figura 23 é possível visualizar a interface do Roborto e suas funcionalidades.

Figura 23: Roborto.



Fonte: Fadhil (2018).

4.8 BOT ELLEN

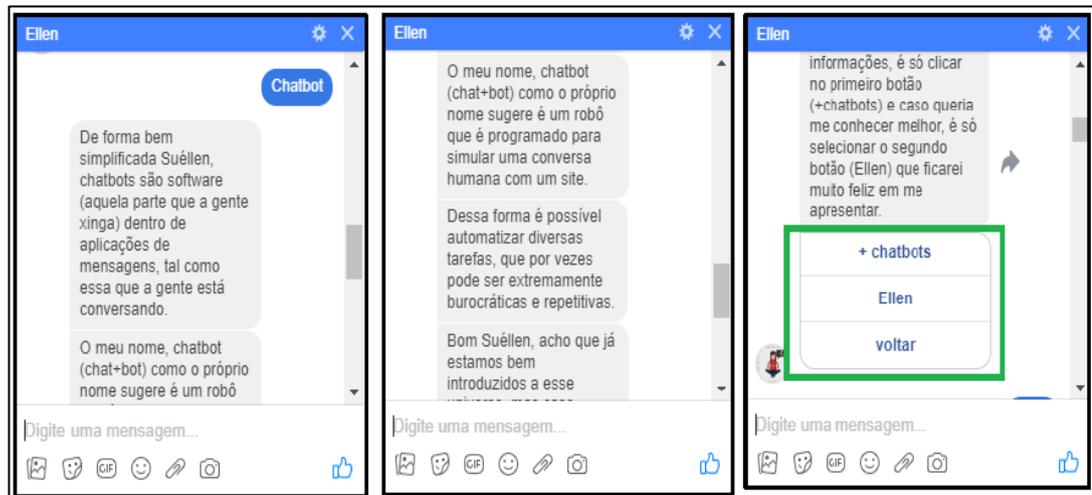
Chatbot Ellen é uma ferramenta para auxiliar pessoas com doenças crônicas não transmissíveis a gerir os medicamentos. O *chatbot* auxilia o usuário a cadastrar os medicamentos e gera alertas sobre as medicações de uso contínuo que o paciente precisa tomar (SANTOS, 2018).

Além disso, o *chatbot* explica aos usuários as complicações que a doença registrada por ele pode causar, caso não seja feito o tratamento farmacológico adequado (SANTOS, 2018).

O Robô ainda apresenta em seu menu a opção de encontrar farmácias próximas a localização do paciente.

O *chatbot* Ellen foi criado para utilizar a plataforma do Facebook Message para a troca de mensagens com o usuário (SANTOS, 2018). A figura 24 apresenta a interface do *chatbot* Ellen.

Figura 24: Bot Ellen



Fonte: Santos,(2018).

4.9 Sumário de Trabalhos Relacionados.

A tabela 1 lista de forma resumida os trabalhos relacionados e a tecnologia utilizada pelos autores citados na seção de trabalhos relacionados.

Na presente tabela a maior parte dos trabalhos relacionados são aplicações mobile e três aplicações utilizam chatbot para auxiliar o idoso no gerenciamento dos medicamentos de uso diários. Os três trabalhos que usam bot na gestão dos medicamentos, está relacionado ao uso de serviços redes sociais para habilitar a comunicação do chatbot com o paciente.

Este trabalho propõe um sistema computacional baseado em inteligência artificial com processamento e síntese de voz para auxiliar os idosos na gestão de medicamentos.

Tabela 1: Sumario de trabalho relacionado

Autores	Base de Conhecimento	Plataforma	Aprendizagem de Máquina	Reconhecimento e Síntese de voz
Inácio, (2014)	Não	Mobile	Não	Não
Rocha(2018)	Não	Web	Não	Não
SNS	Não	Mobile	Não	Não
Igarashi (2019)	Não	Mobile	Não	Não
Oliveira (2020)	IBM Watson	Mobile	Supervisionado	Não
Fadhil (2018)	Microsoft Bot Framework2	Mobile/Web	Supervisionado	Não
Santos (2018)	Chatfuel	Mobile	Supervisionado	Não
Stutzel (2014)	Não	Mobile	Não	Não

Fonte: Autor (2021).

Nesta seção foram levantados os trabalhos semelhantes a proposta desta dissertação para fundamentar e provar que o que está sendo proposto já possui um modelo validado. Com base nos trabalhos relacionados é possível verificar que é possível melhorar a solução proposta.

5 SOLUÇÃO PROPOSTA, SAGMID: CHATTERBOT PARA AUXÍLIO DE GESTÃO DE MEDICAMENTOS EM IDOSOS UTILIZANDO AIML ..

Definir a arquitetura de um sistema representa uma etapa importante no desenvolvimento de software. Nesta fase são analisados os requisitos funcionais e não funcionais que serão importantes para o desenvolvimento do projeto de software (NETO, 2016). Neste capítulo serão apresentados os diagramas de caso de uso, diagrama de classe e diagrama de sequência, assim como a arquitetura do projeto.

5.1 Arquitetura Proposta

O SAGMID é um protótipo cujo objetivo é auxiliar o idoso na gestão de medicamentos que são utilizados por ele. A ferramenta descrita neste trabalho processa e sintetiza voz para facilitar a interação do usuário com a aplicação nas atividades de cadastramento de medicamento, cadastramento de consulta e a partir destes será gerado o alarme para notificar o idoso sobre o horário da consulta médica e o horário do uso do medicamento.

Para que o *chatbot* consiga interagir com o usuário simulando uma conversa, o SAGMID possui uma base de conhecimento construída em AIML que permite o bot realizar a coleta dos dados para o cadastro de medicamento e consulta. Além disso, o *chatbot* é capaz responder a saudações, como exibido na figura 25.

A figura 26 descreve a interação do usuário com *chatbot* via comando de voz, de acordo com o padrão de entrada fornecido pelo usuário o sistema irá buscar um padrão correspondente nos arquivos AIML e devolverá a resposta via voz para o usuário.

Figura 25: Resposta para Saudação dada pelo chatbot

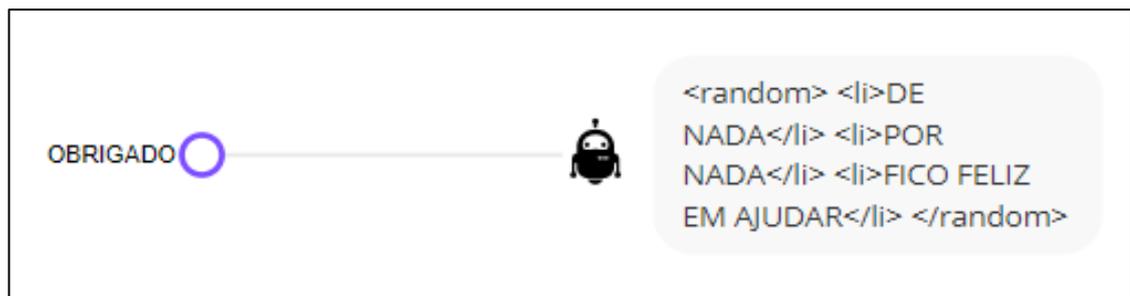
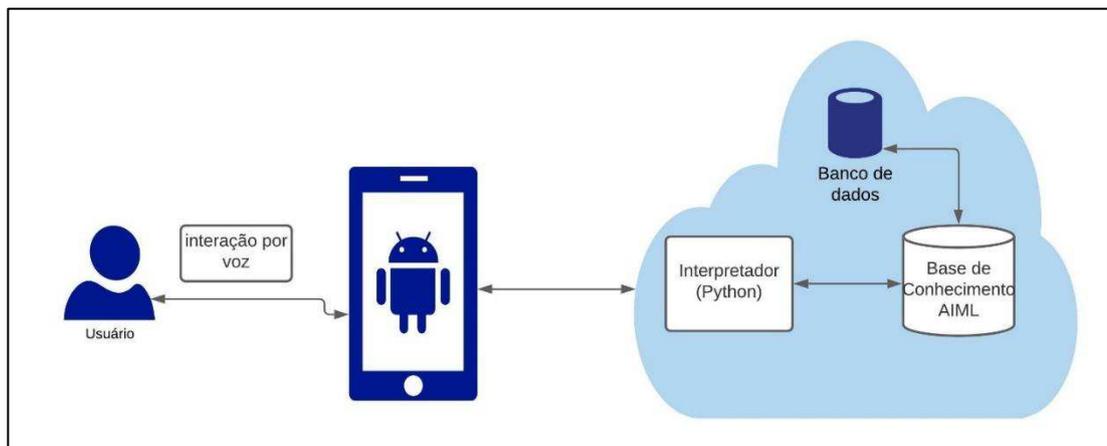


Figura 26: Arquitetura SAGMID.



Fonte: Autor (2021).

5.3 Modelagem do SAGMID

A área de requisitos de software preocupa-se com o levantamento, análise, especificação e validação das propriedades que devem ser apresentadas para resolver tarefas relacionadas ao software em desenvolvimento. Requisitos são a expressão mais detalhada sobre aquilo de que o usuário ou cliente precisa em termos de um produto de software (GUEDES, 2011).

5.3.1 Caso de Uso

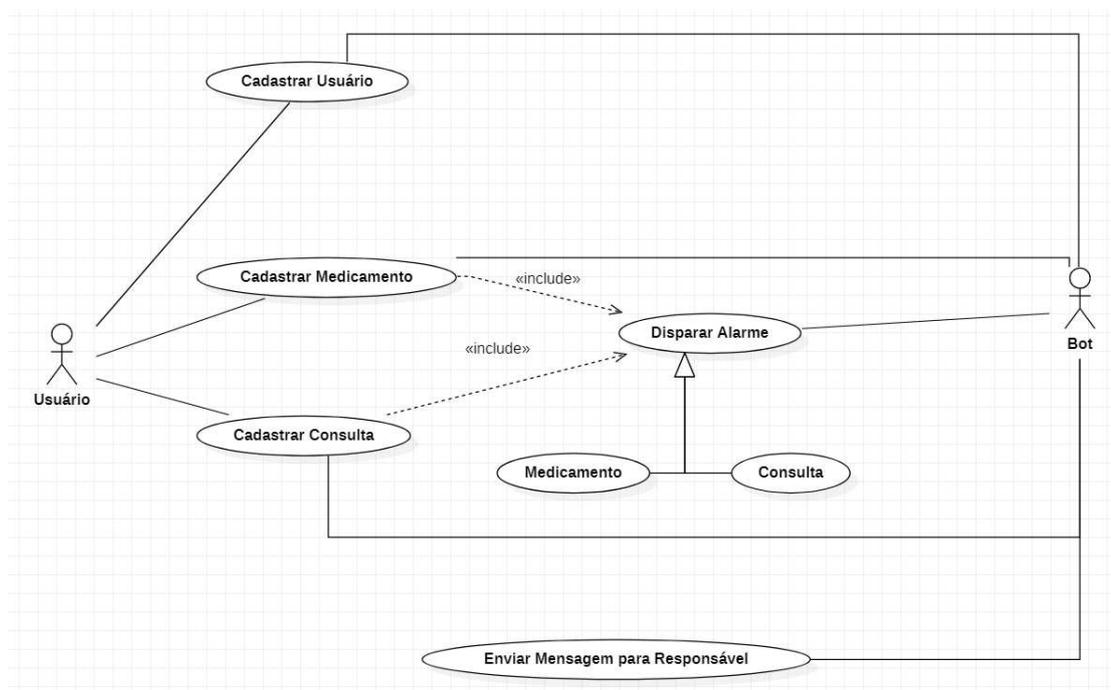
O diagrama de caso de uso, procura, por meio de uma linguagem de programação simples, possibilitar a compreensão do comportamento externo do sistema por qualquer pessoa, tentando apresentar o sistema por intermédio de uma perspectiva do usuário (GUEDES, 2011).

O Diagrama de caso de uso concentra-se em dois itens principais: atores e caso de uso (GUEDES, 2011). Os atores representam os papéis desempenhados pelos diversos usuários que poderão utilizar de alguma maneira, os serviços e funções do sistema; já o caso de uso representa os requisitos do sistema, ou seja, refere-se aos serviços, tarefas ou funcionalidades identificadas ao software e que podem ser utilizadas de alguma maneira pelos atores que interagem com o sistema (GUEDES, 2011). Na perspectiva de aplicação mobile, pode-se utilizar para visualizar interações entre os usuários e sistema para ajudar a definir a responsabilidade e ações.

A figura 27 exemplifica os requisitos do sistema SAGMID, que são cadastro de usuário, medicamento e consulta. Outra funcionalidade encontrada no SAGMID é o alerta de medicamento e consulta, esses alarmes serão gerados quando o usuário informar a hora da ingestão do medicamento, assim como, gerar o alarme com a data da consulta do idoso.

O caso de uso abrange a realidade do sistema, contém dois atores, um ator que é o usuário do sistema propriamente dito e outro; é o *chatbot* que interage com o usuário para realizar as atividades descritas:

Figura 27: Modelagem de caso de uso do sistema para apoio do tratamento farmacológico para idosos desenvolvido no StartUml.



Fonte : Autor (2021).

5.3.2 Diagrama De Classe

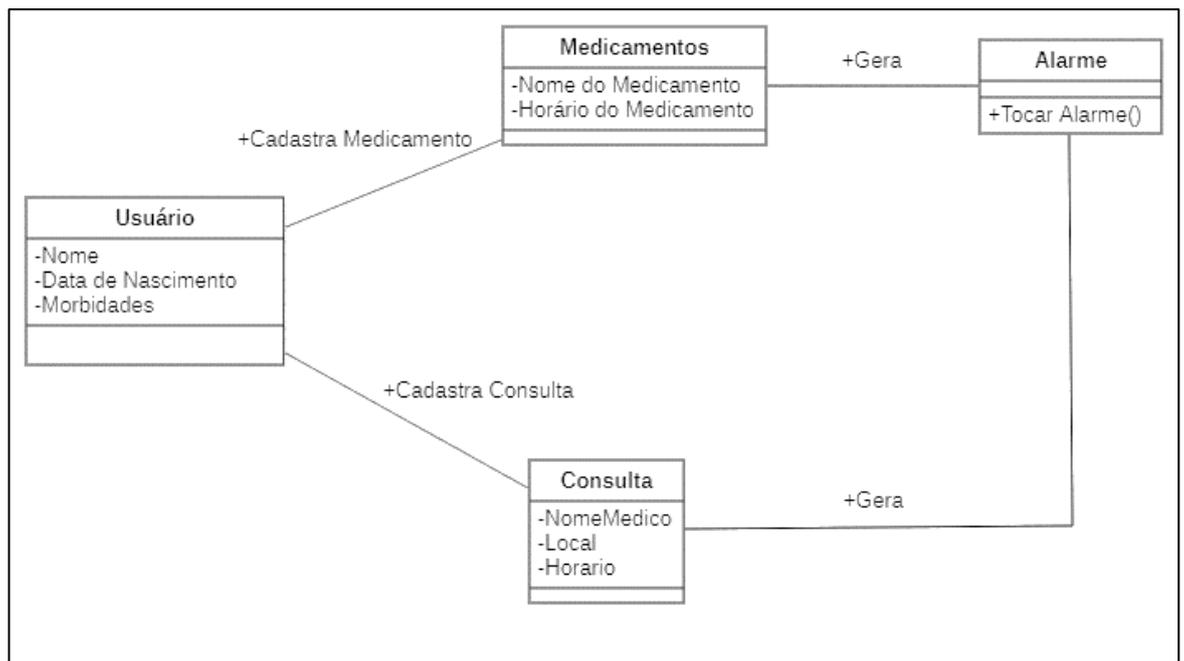
O Diagrama de classe define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam entre si (GUEDES, 2011).

O diagrama de classe do sistema que oferece apoio ao tratamento farmacológico de idosos é composto pelas classes: usuário, medicamento, consulta, cadastro e alarme. A classe usuário é responsável por obter as informações do usuário, tais como: nome, data de nascimento, doença que o

usuário possa ter. Na classe medicamento, as informações obtidas estão relacionadas ao medicamento que o usuário toma e o horário que deve ser ingerida a medicação, a classe consulta obtém informações, como: horário, local e informações importantes sobre as consultas médicas que o usuário deve realizar.

As classes cadastro e alarme obtém as informações das classes consulta e medicamento por meio do objeto da classe, essas realizam o cadastro dos medicamentos e consultas, do mesmo modo funciona a classe alarme para gerar os alertas sobre os horários da ingestão dos medicamentos e horários das consultas. A figura 28 mostra o diagrama de classe do protótipo do sistema para apoio ao tratamento farmacológico de idosos.

Figura 28: Diagrama de classe do sistema para apoio do tratamento farmacológico para idosos desenvolvido no StartUml.



Fonte : Autor (2021).

5.3.3 Diagrama De Sequência

Uma vez definido e modelado, o diagrama de caso de uso pode criar o diagrama de sequência para detalhar mais a interação dos atores com o sistema (GUEDES, 2011).

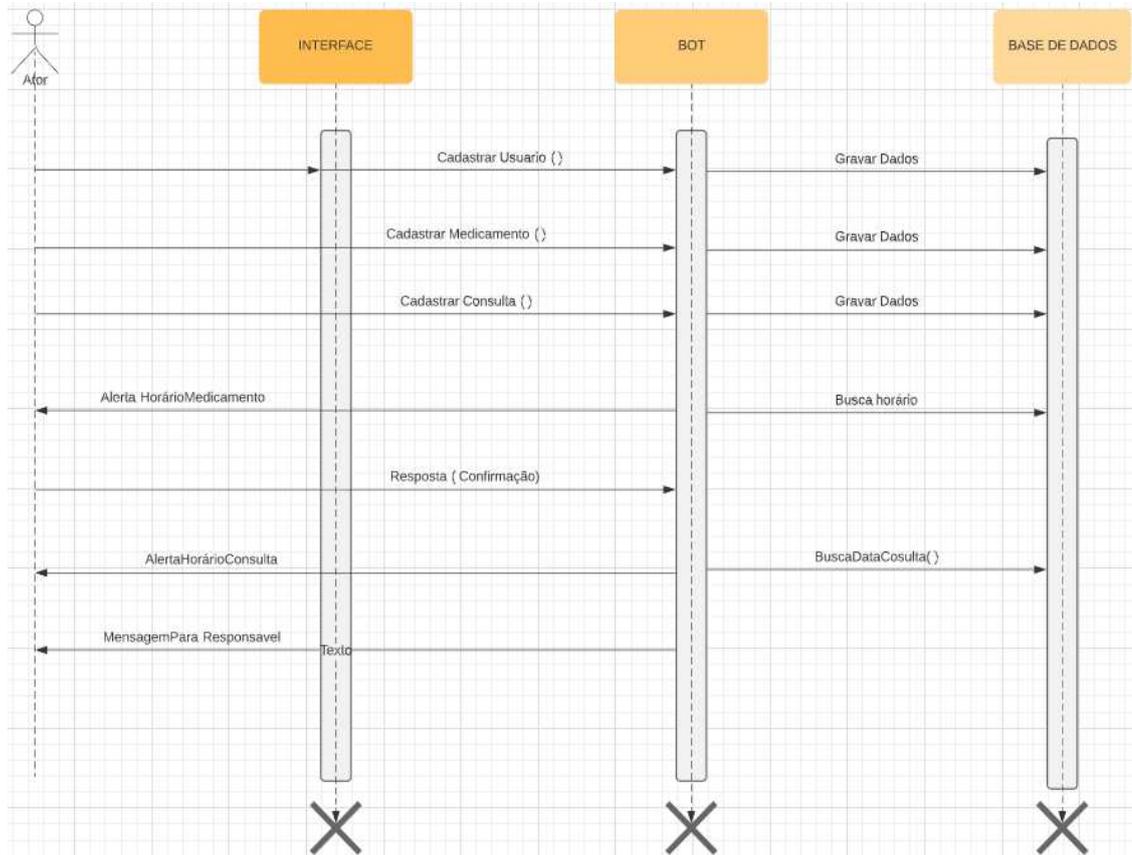
Os atores modelados no diagrama de sequência são instâncias dos atores declarados no diagrama de caso de uso, representando entidades externas que interagem com os sistemas e que solicitam serviços, gerando, assim eventos que iniciam processos (GUEDES, 2011).

A figura 29 mostra o diagrama de sequência do sistema para apoio do tratamento farmacológico de idosos, onde é possível visualizar as etapas que serão realizadas pelo sistema, desde a primeira interação do usuário até a resposta enviada pelo bot, de acordo com os passos:

1. A primeira interação com o bot: o usuário informa ao bot que deseja se cadastrar como usuário do sistema, começa, então, a interação de perguntas e repostas;
2. O bot ao finalizar o cadastro do usuário pergunta se ele deseja cadastrar um medicamento;
3. O bot pergunta para o usuário se ele deseja cadastrar uma consulta médica;
4. Ao finalizar o cadastro de medicamentos e consultas, o bot gera o alarme para disparar, diariamente, para o medicamento e para a consulta no dia específico informado.

O usuário precisa informar ao bot se ele ingeriu o medicamento, caso não seja informado, o bot envia uma mensagem para o responsável do idoso para que o mesmo possa alertar sobre a medicação.

Figura 29: Diagrama de seqüência do sistema para apoio do tratamento farmacológico para idosos desenvolvido no StartUml.



Fonte : Autor (2021).

Neste capítulo foram descritos, a arquitetura do SAGMID e os diagramas UML: diagrama de caso de uso, diagrama de classe e diagrama de atividade, que foram elaborados na fase de projeto desta pesquisa.

6. CHATBOT SAGMID

O *chatbot* SAGMID utiliza a arquitetura cliente e servidor, o cliente da aplicação é o App Android que funciona do lado do usuário capturando os dados, já a camada de servidor processa a entrada do usuário e devolve como resultado a resposta para a entrada fornecida. O SAGMIG foi desenvolvido utilizando o Programy que é um interpretador Python para AIML 2.0. O Programy possui uma biblioteca *py-aiml* que faz o processamento das tags AIML e implementa o mecanismo de resposta que é chamado de *kernel respond*, através deste o *py-aiml* busca o padrão correspondente a entrada e devolve a resposta para o usuário (BIRD, 2009). A arquitetura de um *chatbot* AIML pode ser visto na figura 31, na seção 5.1.

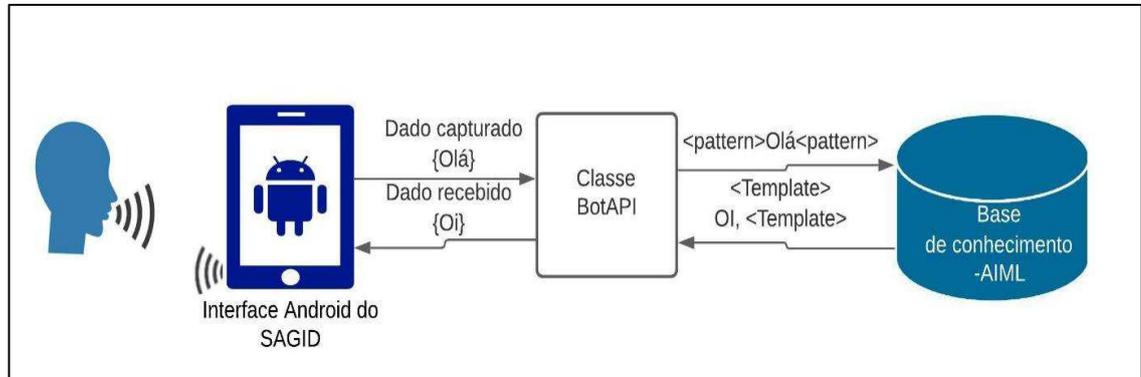
Para armazenamento dos dados do usuário foi utilizado o Django na versão 3.1, é um framework para aplicações web gratuito e de código aberto, escrito em Python, que utiliza o padrão de projetos MTV (Model, Template e View). O model é onde acontece toda a interação com o banco de dados, já o template onde é feita a renderização dos dados para visualização; e por fim, o View é a camada responsável pela lógica da aplicação. O Model do SAMID possui três classes: a User, a Appointment e Medicament. A classe user é responsável por capturar os dados do usuário, tais como: nome e idade. Já a classe Appointment faz o cadastro da consulta do usuário, armazenando dados como, nome do médico e data da consulta. O model para cadastro do dado do usuário pode ser visto no Apêndice A.

Para a comunicação com o usuário foi desenvolvido uma interface em Android que utiliza o recognition da google que vem instalado no smartphone para fazer o reconhecimento de voz, e para sintetizar a voz o sistema, o SAGMID usa o Text-to-Speech que é provido de acordo com o fabricante do aparelho.

O dado capturado pelo recognition do aparelho é enviado para a classe BotAPI do Sagmid. A classe BotAPI irá buscar dentro do arquivo AIML o padrão que corresponde com a entrada que o usuário forneceu para o sistema e a parte do casamento do padrão encontrado, o SAGMID envia a resposta para interface Android que utiliza o Text-to-Speech do dispositivo para devolver a resposta ao usuário. Na figura 30 é possível visualizar o fluxo que o dado

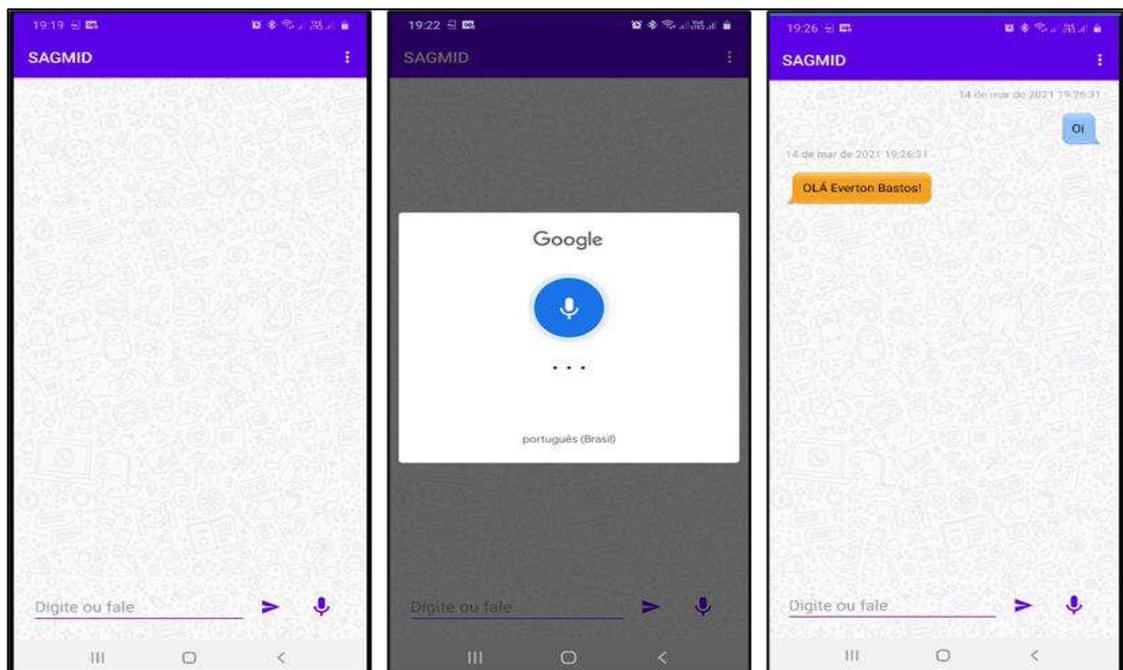
percorre dentro do SAGMID; e na figura 31, podemos verificar a interação do usuário com *chatbot*.

Figura 30: Fluxo de dados no SAGMID.



Fonte: Autor (2021)

Figura 31: Interação do usuário com o SAGMID.



Fonte: Autor (2021).

O SAGMID possui um módulo para cadastro de consulta, os dados para serem inseridos no banco de dados são obtidos pelos arquivos AIML e processados pela classe BOTAPI, passados para a classe appointments, para serem inseridos no banco, ao informar os dados, o alarme é gerado para sinalizar o dia da consulta. O AIML, para capturar os dados do usuário, encontra-se no Apêndice B.1.

O cadastro de medicamento funciona de forma semelhante ao cadastro de consulta. O usuário vai informar o nome do medicamento, o horário de ingestão e frequência de uso do remédio, esses dados serão obtidos via diálogo com o *Chatbot*. O arquivo AIML, responsável pelo cadastro de medicamento, cria um fluxo de diálogo com as tag's <that> e <topic> que criará um fluxo de perguntas para interagir com o usuário.

6.1 . Validação e Verificação.

Para mensurar a validação do SAGMID, foi disponibilizado de forma gratuita para 20 (vinte) usuários utilizarem o aplicativo com a finalidade de avaliarem o funcionamento e o desempenho com o objetivo de testar e validar o software.

6.1.2 Procedimento Para Coleta e Análise de Dados.

Para validação desta pesquisa foi feito um estudo por amostra, composto por 20 (vinte) usuários. Os dados foram coletados a partir de questionários aplicados, aos quais os usuários foram submetidos, que está presente no Apêndice C.

A coleta dos dados foi realizada através de formulário criado no Google Forms. Este é uma plataforma do Google que permite aos usuários criarem formulários dinâmicos com perguntas e repostas personalizadas, podendo ser utilizadas as opções de múltipla escolha, texto livre, respostas longas e curtas.

Para categorizar os dados de forma que pudesse obter informações dos dados levantados foi utilizado o Power BI, que é uma ferramenta da Microsoft para análise de dados.

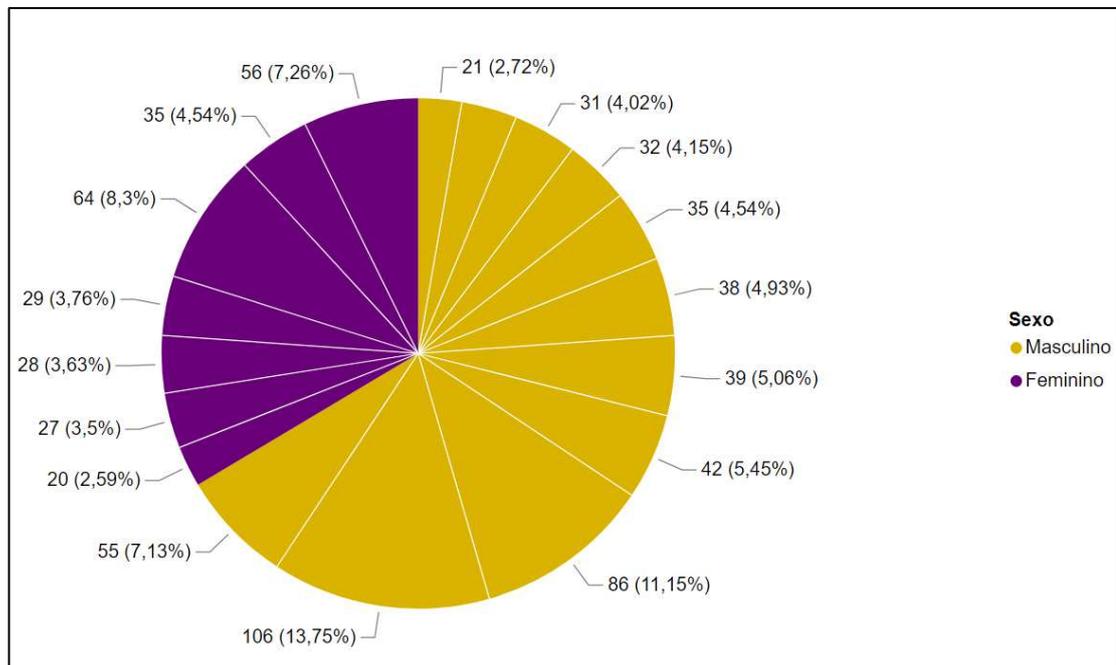
Os dados da tabela 2 gerados no Power BI mostra que 61,90% dos usuários que participaram do teste são homens e 38,10% dos usuários são mulheres. Conforme está descrito no gráfico 1, os usuários participantes têm a faixa etária de idade entre 20 à 56 anos.

Tabela 2: Percentual de Participantes.

Sexo	Percentual de Participantes
Masculino	61,90%
Feminino	38,10
Total	100%

Fonte: Auto (2021)

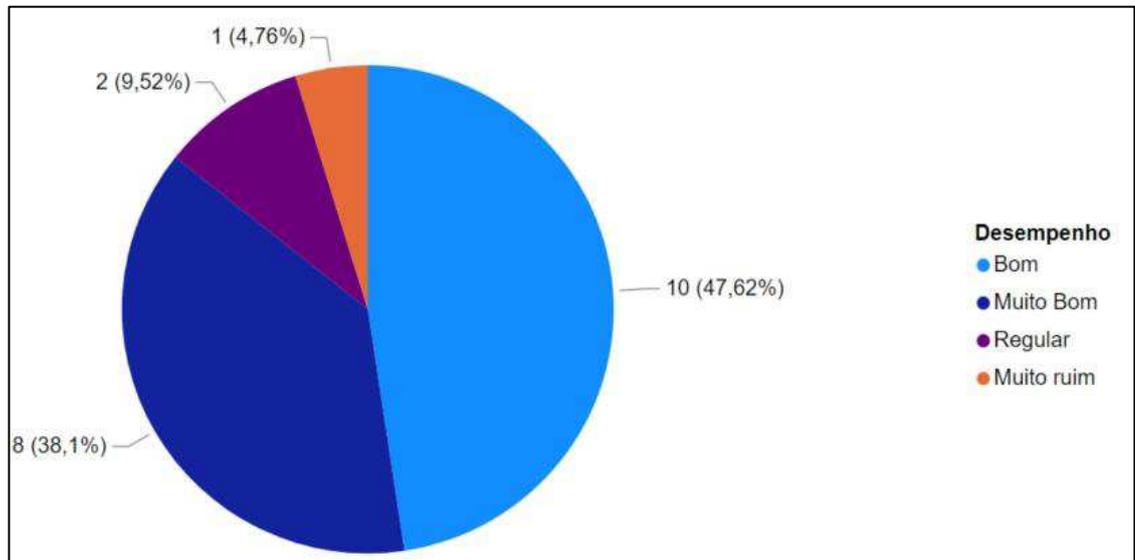
Gráfico 1: Idade dos participantes por sexo.



Fonte: Autor (2021).

Neste contexto, em que fica claro o total de participantes da pesquisa, podemos descrever o gráfico 2, que de acordo com o mesmo 47,62% dos usuários acharam “Muito bom” o desempenho do SAGMID e 38,1% dos participantes disseram que o desempenho era “bom”, 9,52% dos participantes classificaram o desempenho da ferramenta como regular e apenas 4,76% classificaram como muito ruim. Podemos concluir que o grau de satisfação do SAGMID de modo geral é de 85% em relação ao seu desempenho.

Gráfico 2: Desempenho do SAGMID.

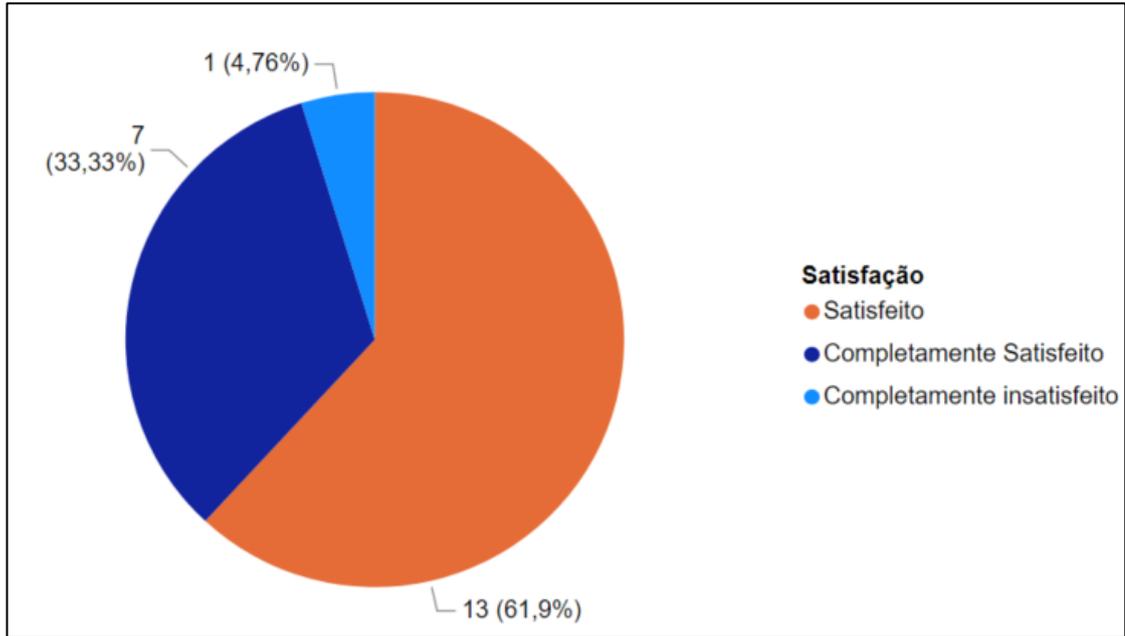


Fonte: Autor (2021).

Em relação a satisfação do usuário quanto ao cadastro de medicamento ou consulta, 33,33% dos participantes da pesquisa responderem que estão “Completamente Satisfeitos” ao fazer o cadastro de consulta ou medicamento no SAGMID, enquanto 61,9% dos usuários responderam que estão “satisfeito” e 4,76% responderam que estão completamente insatisfeito com a capacidade do SAGMID em cadastrar um medicamento ou uma consulta por voz. De modo geral, o SAGMID teve uma aceitação de 95,23% de aceitação pelos usuários. No Gráfico 3, podemos visualizar a disposição dos dados levantados.

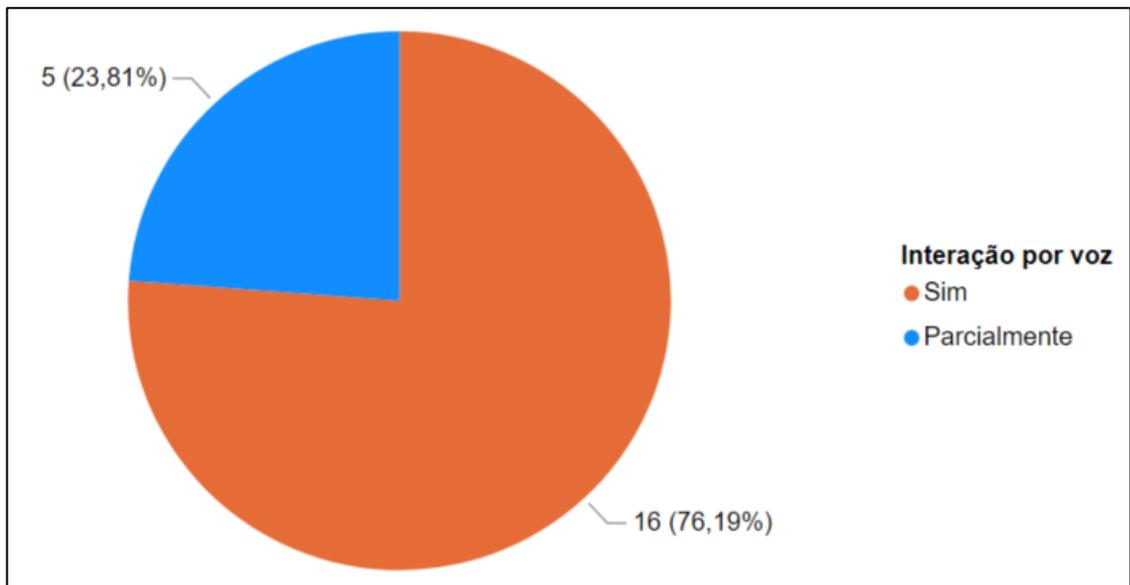
No gráfico 4, dispõe os dados sobre a pergunta “A interação por voz com a ferramenta facilitou o cadastro do medicamento ou consulta?”, 76,19% dos usuários responderam “Sim”, ou seja, a interação por voz diminuiu o esforço que o usuário teria, caso fosse digitar as informações solicitadas. Em contra partida, 23,81% dos participantes afirmaram que a interação por voz ajudou, parcialmente, na atividade de cadastro de consulta ou medicamento. Podemos concluir nesta análise, que de modo geral, a interação por voz facilita o uso da ferramenta no quesito de inserção de informação.

Gráfico 3: Satisfação do usuário em relação ao cadastro de medicamento ou consulta.



Fonte: Autor (2021).

Gráfico 4: Satisfação em relação a interação por voz como ferramenta no cadastro de medicamento ou consulta .

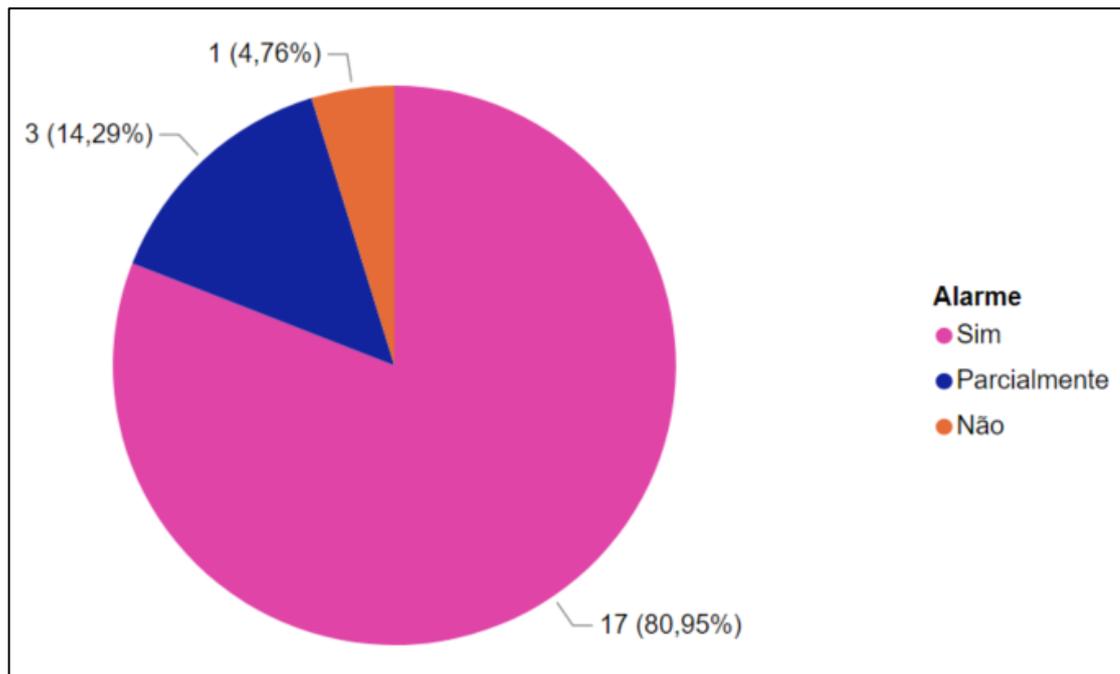


Fonte: Autor (2021).

No gráfico 5 podemos verificar as respostas dos participantes das pesquisas em relação a pergunta “O alarme configurado no momento do

cadastro do medicamento ou consulta funcionou da maneira esperada?”, 80,95% dos usuários responderam “Sim”, ou seja, o SAGMID configurou o alarme de acordo com os dados informados, 14,29% dos participantes responderam “parcialmente”, devido algum erro do sistema no cadastro de medicamento e somente 4,76% responderam que “Não”, evidenciando que o SAGMID não conseguiu configurar o alarme de modo esperado.

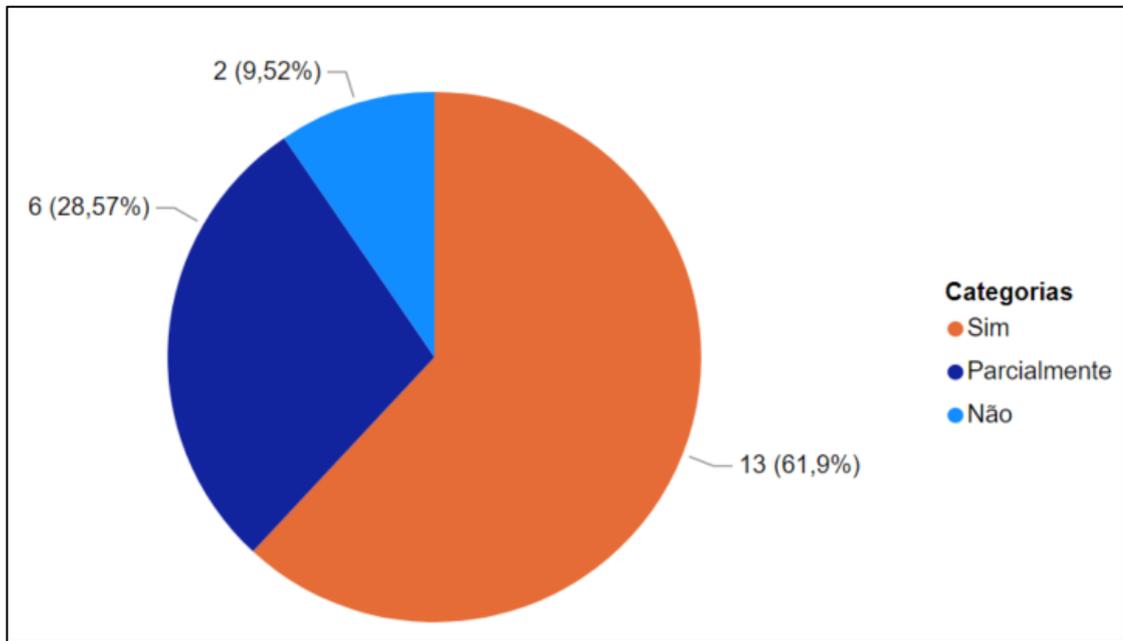
Gráfico 5: Configuração do alarme.



Fonte: Autor (2021).

No gráfico 6, analisamos as respostas dos usuários que participaram dos testes do SAGMID para pergunta: “O aplicativo conseguiu responder suas perguntas de forma satisfatória?”, 61,9% dos participantes responderam “Sim”, ou seja, o SAGMID conseguiu responder as perguntas do usuário dentro do contexto para o qual foi desenvolvido e 28,57% dos usuários respondeu “Parcialmente”, pois o aplicativo pode não ter compreendido o que o usuário falou, ainda temos 9,52% dos participantes respondendo “Não”, afirmando que o sistema não foi capaz de responder as perguntas feitas.

Gráfico 6: Satisfação sobre as repostas dadas pelo SAGMID.



Fonte: Autor (2021)

Conclui-se, com a análise dos dados levantados sobre diferentes aspectos de funcionamento do SAGMID, que ele pode ser utilizado como ferramenta para auxiliar na gestão de medicamentos em idosos, tendo em vista que o sistema teve um desempenho de 70% nos pontos analisados.

Nesta seção foram apresentados detalhes sobre implementação do SAGMID e as bibliotecas utilizadas da linguagem Python e o framework para o desenvolvimento do banco de dados. Foi abortado o fluxo de funcionamento do SAGMID e sua interface e modo de interação será por voz. Além disso, foram analisados dados sobre diferentes aspectos de funcionamento do sistema, estes dados validam o aplicativo como uma ferramenta que pode ser utilizada para o auxílio da gestão de medicamentos em idosos.

7. CONCLUSÃO

Apresentou-se neste trabalho um *Chatbot* para os usuários idosos, desenvolvido para cadastrar consultas e medicamentos através da interação por voz. Neste sentido, esperava-se uma aceitação de que superasse os 70% dos usuários.

Para atingir os objetivos gerais, a pesquisa atendeu aos objetivos específicos que foram propostos inicialmente. Neste contexto, foi proposto investigar o uso de chatbot como ferramenta de auxílio na gestão de medicamentos para idosos, nas pesquisas constatou-se que este tipo de sistema é empregado para auxiliar o idoso com seus medicamentos, mas os estudos levantados a interface propostas se mostravam difícil para o idoso interagir, pois a maior parte das ferramentas tem os menus em formato pequeno, letras pequenas e sem possibilidade de outros meios de interação a não ser via texto.

Com base nos estudos feitos para o desenvolvimento deste trabalho foi proposta os alguns diagramas UML, tais como: diagrama de caso de uso, diagrama de classe e diagrama de sequência. Estes diagramas foram desenvolvidos para analisamos os requisitos para desenvolvimento do SAGMID, com base nos sistemas que foram estudados, nesta fase de projeto analisamos que a interação por voz com o SAGMID facilitaria a utilização pelo idoso.

Além disso, avaliamos a arquitetura proposta para o SAGMID, comparando a funcionalidade levantadas por outros pesquisadores, e assim foi possível chegar à conclusão de que o uso de chatbot que pudesse sintetizar e reconhecer voz seria a melhor solução para o público alvo, os idosos.

Desta forma foi possível desenvolver o SAGMID a partir de todos os requisitos levantados. O SAGMID foi desenvolvido para a plataforma Android e utilizar arquitetura cliente servidor. O servidor foi desenvolvido na linguagem Python, pois nesta linguagem encontramos um interpretador para os arquivos AIML que foram utilizados para desenvolver a conversão do chatbot SAGMID.

Para analisar se o SAGMID atende os requisitos propostos no seu desenvolvimento foi disponibilizado aos usuários, de forma gratuita, o SAGMID

para a realização de teste de funcionalidade, analisando, assim, o desempenho e qualificação da ferramenta.

Constatou-se que os resultados obtidos foram satisfatórios, pois de acordo com as análises feita, o SAGMID foi aprovado com 80%. Enquanto, se tem por base para aprovação de algum aplicativo 75% de satisfação. Portanto, os resultados obtidos foram satisfatórios.

As perspectivas de trabalhos futuros, nesta pesquisa, apontam adicionar novas funcionalidades ao SAGMID, para isso, segue, alguns desafios:

- ✓ Aumentar a base de conhecimento do SAGMID, a fim de melhorar a conversão;
- ✓ Melhorar a função de alarme;
- ✓ Implementar uma funcionalidade para verificação de pressão;
- ✓ Estabelecer parâmetros para iniciar uma conversa com o idoso, caso este passe muito tempo sem acessar o SAGMID.
- ✓ Testa com usuários idosos, pois não foi possível alcançarmos este público devido a pandemia da Covid-19.

Portanto, após análise, pode-se concluir que o SAGMID atende ao objetivo, responde com êxito ao desempenho da aplicação em ambiente real, computa o desempenho em diferentes idades e afere o desempenho do sistema entre os gêneros. Desse modo, acredita-se que o SAGMID teve êxito de acordo com os resultados obtidos.

Referências

- [1] A. P. d. L. Dias, “Desenvolvimento de um Chatbot para apoio clínico,” 2018.
- [2] I. Y. Igarashi, “MedAmigo: UM APLICATIVO DE AUXÍLIO A IDOSOS NA INGESTÃO DE MEDICAMENTOS,” 2019.
- [3] M. F. L. B. d. M. A. S. A. B. Matheus Costa Stutzel, “SMAI - Sistema Móvel de Assistência ao Idoso,” *16º Workshop de Informática Médica*, 2015.
- [4] A. Piau, “Information and Communication Technologies, a Promising Way to Support Pharmacotherapy for the Behavioral and Psychological Symptoms of Dementia,” *Frontiers Pharmacology*, 2019.
- [5] E. A. Bayan Abu Shawar, “Chatbots: Are they Really Useful?,” *LDV-Forum*, 2007.
- [6] A. R. C. Fernando da Silva CARES, “O USO DE CHATBOTS PARA AUTOMATIZAÇÃO DE PROCESSOS NA ERA COGNITIVA,” 2018.
- [7] L. M. A. C. Rafaela Lunardi Comarella, “CHATTERBOT: conceito, características, tipologia e construção,” *Inf. & Soc.:Est.*, vol. 18, 2008.
- [8] F. G. K. Fenfei Guo, “Topic-based Evaluation for Conversational Bots,” *Conference on Neural Information Processing Systems*, 2018.
- [9] J.-P. Sansonnet, “Architecture of a Framework for Generic Assisting Conversational Agents,” *LIMSI-CNRS BP 133 F-91403 Orsay Cedex*, 2006.
- [10] A. C. D. A. C. P. Natália Oliveira, “HelpCare: Um Protótipo de ChatBot para o Auxílio do Tratamento de Doenças Crônicas.”. *Universidade Federal do Pará (UFPA)*, 2018.
- [11] S. d. L. Pereira, “Processamento de Linguagem Natural,” 2018.

- [12] T. S. Spanhol, "UM ESTUDO SOBRE A INTERAÇÃO ENTRE USUÁRIOS E CHATTERBOTS," Araranguá, 2017.
- [13] A. M. M. d. Neves, "IAML: Um Mecanismo para o tratamento de intenção em Chatterbots," Recife, 2005.
- [14] B. J. Kříž, "Chatbot for Laundry and Dry Cleaning Service," 2017.
- [15] R. V. Alvares, "Investigação do Processo de Stemming na Língua Portuguesa.," 2005.
- [16] V. G. Roncero, "CLASSIFICAÇÃO SEMI-SUPERVISIONADA DE TEXTOS EM AMBIENTES DISTRIBUÍDOS," 2010.
- [17] A. A. M. Ticom, "APLICAÇÃO DAS TÉCNICAS DE MINERAÇÃO DE TEXTOS E SISTEMAS ESPECIALISTAS NA LIQUIDAÇÃO DE PROCESSOS TRABALHISTAS," 2007.
- [18] S. C. S. Pinto, "Processamento de Linguagem Natural e Extração de Conhecimento," 2015.
- [19] C. M. d. Santos, "Classificação de Documentos com Processamento de Linguagem Natural," 2015.
- [20] J. Cahn, "CHATBOT: Architecture, Design, & Development," 2017.
- [21] G. T. A. Guedes, UML 2: Uma abordagem prática, Sao Paulo: Novatec, 2011.
- [22] R. Maitino Neto, Engenharia de software, Londrina: Editora e Distribuidora Educacional, 2016.
- [23] S. S. d. F. D. A. E. A. P. P. C. D. C. N. T. Santos, "DESENVOLVIMENTO DO CHATBOT ELLEN COMO," 2018.
- [24] T. Rochaa, "Aplicação Web para o auxílio da toma de medicamentos e de gestão de informação de consultas," Portugal, 2018.

- [25] A. Fadhil, "A Conversational Interface to Improve Medication Adherence: Towards AI Support in Patient's Treatment," Italy, 2018.
- [26] P. C. M. Inácio, "PROTÓTIPO DE APLICAÇÃO ANDROID PARA CONTROLE DA ROTINA DE MEDICAMENTOS PARA IDOSOS," 2014.
- [27] E. K. e. E. L. Steven Bird, Natural Language Processing with Python, vol. First Edition, 2009.
- [28] R. S. Wallace, The Anatomy of ALICE, 2014
- [29] S. Russell e P. Norvig., Inteligencia Artificial, Rio Janeiro: Elsevier, 2010.
- [30] J. C. d. Carvalho e C. F. d. A. Sena, "Problema Relacionado à manutenção do tratamento medicamentoso em pacientes idosos e as contribuições da atenção farmacêutica," *Revista Brasileira de Ciência da Vida*, nº 5, p. 23.
- [31] M. R. d. O. Guilherme Andrioli Cruzato, "O DESENVOLVIMENTO E A NECESSIDADE DOS CHATBOTS NA ATUALIDADE," 2018.
- [32] R. A. P. R. F. V. S. Elizandra Cristina Pedrazzi, "MORBIDADE REFERIDA E CAPACIDADE FUNCIONAL DE IDOSOS," *Cienc Cuid Saude*, 2007.
- [33] C. M. L. B. B. Lara Carvalho Vilela de Lima, "ENVELHECIMENTO E GÊNERO: A VULNERABILIDADE DE IDOSAS NO BRASIL," *Revista Saúde e Pesquisa*, 2009.
- [34] T. A. M. d. Lima, E. R. Fazan, L. L. V. Pereira e M. F. d. Godoy, "Acompanhamento farmacoterapêutico em idosos," 2016.
- [35] B. A. Shawar, "Using dialogue corpora to train a chatbot," 2014.
- [36] O. A. d. A. Junior, "Beck: Um Chatbot Baseado na Terapia Cognitivo-Comportamental para Apoiar Adolescentes com Depressão," p. 165, 2017.
- [37] J. Cahn, "CHATBOT: Architecture, Design, & Development," 2017.

- [38] A. D. M. GALVÃO, “Uma arquitetura para Desenvolvimento Chatterbos com Personalidade,” 2003.
- [39] E. F. Gehrke, “CHATTEREDU 2.0: FERRAMENTA WEB DE AUXÍLIO ACADÊMICO UTILIZANDO CHATTERBOT,” Blumenau, SC., 2017.
- [40] J. Q. U. Leandro Padilha Ferreira, “Desenvolvimento de um chatbot para auxiliar o ensino de Espanhol como Língua Estrangeira,” *Bazar*, p. 5, 2006.
- [41] “Deep Learning for Chatbots, Part 1 – Introduction – WildML.” [Online]. Available: <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/>. [Acessado: 05-Março-2020].
- [42] “Ultimate Guide to Leveraging NLP & Machine Learning for your Chatbot.” [Online]. Available: <https://chatbotlife.com/ultimate-guide-to-leveraging-nlp-machine-learning-for-you-chatbot-531ff2dd870c>. [Accessed: 05-Mar-2018].
- [43] “Deep Learning chatbot - analysis and implementation - Sigmoidal.” [Online]. Available: <https://sigmoidal.io/chatbots-for-b2c-and-deep-learning/>. [Acessado: 07-Mar-2018].
- [44] “Rule based bots vs AI bots – #WeCoCreate – Medium.” [Online]. Available: <https://medium.com/botsupply/rule-based-bots-vs-ai-bots-b60cdb786ffa>. [Acessado: 08-Outubro-2020].
- [45] <https://pypi.org/project/pyttsx3/>. [Acessado: 10-Fevereiro-2021]
- [46] <https://pypi.org/project/SpeechRecognition/>. [Acessado: 15-Fevereiro-2021]
- [47] <https://pypi.org/project/python-aiml/>. [Acessado: 20-Fevereiro-2021]

APÊNDICE

APÊNDICE A- MODEL PARA CADASTRO DE DADOS DO USUÁRIO

```
from django.db import models
from django.utils import timezone

class Base(models.Model):
    class Meta:
        abstract = True
    created_at_stamp = models.DateTimeField(default=timezone.now)
    def save(self):
        if not self.created_at_stamp:
            self.created_at_stamp = timezone.now()
        super().save()

class User(Base):
    uuid = models.CharField(max_length=255, blank=True, null=True)
    name = models.CharField(max_length=255, blank=True, null=True)
    age = models.IntegerField(null=True)

class Appointment(Base):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    doctor = models.CharField(max_length=255, blank=True, null=True)
    date_time = models.DateTimeField(null=True)

class Medicament(Base):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    name = models.CharField(max_length=255)
    time = models.TimeField(null=True, auto_now=False, auto_now_add=False)
    hour_interval = models.IntegerField(null=True)
```

APENDICE B – AIML

B.1 CADASTRO DE CONSULTA

```

<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
  <category>
    <pattern>CADASTRO DE CONSULTA</pattern>
    <template>
      QUAL O NOME DO MÉDICO?
    <think>
      <set name="topic">CONSULTA</set>
    </think>
    </template>
  </category>
  <!-- LINKS -->
  <category>
    <pattern>CADASTRAR CONSULTA</pattern>
    <template>
      <srai>CADASTRO DE CONSULTA</srai>
    </template>
  </category>
  <category>
    <pattern>CADASTRAR DE CONSULTA</pattern>
    <template>
      <srai>CADASTRO DE CONSULTA</srai>
    </template>
  </category>
  <category>
    <pattern>CADASTRAR CONSULTAS</pattern>

```

```
<template>
  <srai>CADASTRO DE CONSULTA</srai>
</template>
</category>
<category>
<pattern>CADASTRO DE CONSULTAS</pattern>
<template>
  <srai>CADASTRO DE CONSULTA</srai>
</template>
</category>
<category>
<pattern>LISTAR CONSULTA</pattern>
<template>
  {{ APPOINTMENTS }}
</template>
</category>
<category>
<pattern>LISTAR CONSULTAS</pattern>
<template>
  <srai>LISTAR CONSULTA</srai>
</template>
</category>
<category>
<pattern>CONSULTAS</pattern>
<template>
  <srai>LISTAR CONSULTA</srai>
</template>
</category>
<!-- LINKS -->
<topic name="CONSULTA">
```

```

<category>
<pattern>*/</pattern>
  <that>* NOME */</that>
  <template>
    QUAL A DATA DA CONSULTA
  <think>
    <set name="nome"><star/></set>
  </think>
  ? EXEMPLO: 11 de março de 2021
</template>
</category>
<category>
<pattern>*/</pattern>
  <that>* DATA DA CONSULTA */</that>
  <template>
    QUAL O HORÁRIO DA CONSULTA
  <think>
    <set name="data_consulta"><star/></set>
  </think>
  ? EXEMPLO: 14:30
</template>
</category>
<category>
<pattern>*/</pattern>
  <that>* HORÁRIO DA CONSULTA */</that>
  <template>
  <think
  <set name="horario_consulta"><star/></set>
  </think>
  #addappointment#

```

```

    </template>
  </category>
</topic>
</aiml>

```

B.2 CADASTRO DE MEDICAMENTO

```

<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
  <category>
    <pattern>CADASTRO DE MEDICAMENTO</pattern>
    <template>
      QUAL O NOME DO MEDICAMENTO?
    <think>
      <set name="topic">MEDICAMENTO</set>
    </think>
    </template>
  </category>
  <category>
    <pattern>LISTAR MEDICAMENTO</pattern>
    <template>
      {{ MEDICAMENTS }}
    </template>
  </category>

  <!-- LINKS -->
  <category>
    <pattern>CADASTRO DE MEDICAMENTOS</pattern>
    <template>
      <srai>CADASTRO DE MEDICAMENTO</srai>
    </template>

```

```
</category>
```

```
<category>
```

```
  <pattern>MEDICAMENTOS</pattern>
```

```
  <template>
```

```
    <srai>LISTAR MEDICAMENTO</srai>
```

```
  </template>
```

```
</category>
```

```
<category>
```

```
  <pattern>LISTAR MEDICAMENTOS</pattern>
```

```
  <template>
```

```
    <srai>LISTAR MEDICAMENTO</srai>
```

```
  </template>
```

```
</category>
```

```
<!-- LINKS -->
```

```
<topic name="MEDICAMENTO">
```

```
  <category>
```

```
    <pattern>*</pattern>
```

```
    <that>* NOME *</that>
```

```
  <template>
```

```
    <think>
```

```
      <set name="nome">
```

```
        <star/>
```

```
      </set>
```

```
    </think>
```

```
      QUAL O HORÁRIO INICIAL PARA O ÍNICIO DO TRATAMENTO?
```

```
      EXEMPLO: 14:30
```

```
    </template>
```

```
  </category>
```

```
<category>
```

```

<pattern>* </pattern>
<that>* HORÁRIO INICIAL PARA O ÍNICIO * </that>
<template>
  <think>
    <set name="horario">
      <star/>
    </set>
  </think>
  DE QUANTAS EM QUANTAS HORAS O MEDICAMENTO DEVE SER
TOMADO?
  EXEMPLO: 12
</template>
</category>
<category>
  <pattern>* </pattern>
  <that>* EM QUANTAS HORAS O MEDICAMENTO * </that>
  <template>
    <think>
      <set name="intervalo_horas"><star/></set>
    </think>
    #addmedicament#
  </template>
</category>
</topic>
</aiml>

```

B.3 CADASTRO DE USUÁRIO

```

<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
  <category>
    <pattern>CADASTRO DE USUARIO</pattern>
    <template>
      QUAL O SEU NOME E SOBRENOME?
    <think>
      <set name="topic">USUARIO</set>
    </think>
    </template>
  </category>
  <!-- LINKS -->
  <category>
    <pattern>CADASTRO DE USUÁRIO</pattern>
    <template>
      <srai>CADASTRO DE USUARIO</srai>
    </template>
  </category>
  <category>
    <pattern>CADASTRO DE USUÁRIOS</pattern>
    <template>
      <srai>CADASTRO DE USUARIO</srai>
    </template>
  </category>
  <category>
    <pattern>CADASTRO DE USUARIOS</pattern>
    <template>

```

```
<srai>CADASTRO DE USUARIO</srai>
</template>
</category>
<!-- LINKS -->
<topic name="USUARIO">
  <category>
    <pattern>* </pattern>
    <that>* SEU NOME * </that>
    <template>
      <think>
        <set name="nome"><star/></set>
      </think>
      QUAL A SUA IDADE?
    </template>
  </category>
  <category>
    <pattern>* </pattern>
    <that>* SUA IDADE</that>
    <template>
      <think>
        <set name="idade"><star/></set>
      </think>
      #setuser#
    </template>
  </category>
</topic>
</aiml>
```

C. Formulário para opinião do usuário sobre SAGMID

1 Qual a sua idade ?

2 Qual o sexo ?

Masculino

Feminino

3 Qual a sua opinião sobre o desempenho do aplicativo ?

Muito bom

Bom

Regular

Ruim

Muito Ruim

4 Como foi sua satisfação em cadastrar um medicamento ou consulta ?

Completamente Satisfeito

Satisfeito

Insatisfeito

Complemente Insatisfeito

5 A interação por voz com a ferramenta facilitou o cadastro do medicamento ou consulta ?

Sim

Não

Parcialmente

6 O alarme configurado no momento do cadastro do medicamento ou consulta funcionou da maneira esperada ?

Sim

Não

Parcialmente

7 O aplicativo conseguir responder suas perguntas de forma satisfatória ?

Sim

Não

Parcialmente

8 Faça sua crítica e sugestão sobre a ferramenta.
