



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

MACAIO UPECOR GOMES CACABRO

**IARA: AN ARCHITECTURAL MODEL TO SUPPORT THE DEVELOPMENT OF
PROACTIVE CHATBOTS FOR AUTOMATED MISINFORMATION DETECTION
AND MEDIA LITERACY EDUCATION**

FORTALEZA

2023

MACAIO UPECOR GOMES CACABRO

IARA: AN ARCHITECTURAL MODEL TO SUPPORT THE DEVELOPMENT OF
PROACTIVE CHATBOTS FOR AUTOMATED MISINFORMATION DETECTION AND
MEDIA LITERACY EDUCATION

Dissertação apresentada ao curso de mestrado
do Programa de Pós-Graduação em Ciências
da Computação do Centro de Ciências da
Universidade Federal do Ceará, como requisito
parcial à obtenção do título de mestre em
Ciência da Computação. Área de Concentração:
Sistemas da Informação

Orientador: Prof. Dr. José Maria Da Silva
Monteiro Filho

Coorientador: Prof. Dr. José Wellington
Franco da Silva

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C124i Cacabro, Macaio Upecor Gomes.

IARA: an architectural model to support the development of proactive chatbots for automated misinformation detection and media literacy education / Macaio Upecor Gomes Cacabro. – 2023.
70 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2023.

Orientação: Prof. Dr. José Maria da Silva Monteiro Filho.

Coorientação: Prof. Dr. José Wellington Franco da Silva.

1. Chatbot pró-ativo. 2. Detecção automática de desinformação. 3. Aprendizado de máquina. I. Título.
CDD 005

MACAIO UPECOR GOMES CACABRO

IARA: AN ARCHITECTURAL MODEL TO SUPPORT THE DEVELOPMENT OF
PROACTIVE CHATBOTS FOR AUTOMATED MISINFORMATION DETECTION AND
MEDIA LITERACY EDUCATION

Dissertação apresentada ao curso de mestrado
do Programa de Pós-Graduação em Ciências
da Computação do Centro de Ciências da
Universidade Federal do Ceará, como requisito
parcial à obtenção do título de mestre em
Ciência da Computação. Área de Concentração:
Sistemas da Informação

Aprovada em: 27 de Novembro de 2023

BANCA EXAMINADORA

Prof. Dr. José Maria Da Silva Monteiro Filho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Wellington Franco da Silva (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Angelo Roncali Alencar Brayner
Universidade Federal do Ceará (UFC)

Prof. Dr. José Gilvan Rodrigues Maia
Universidade Federal do Ceará (UFC)

À minha mãe, por sempre me apoiando e acreditar em mim. Quando eu era criança, ela nos diz o seguinte "Eu não tive oportunidade de estudar por causa do meu pai que não me deixou ir para escola, pois ele preferia que eu fosse para interior ajudar no trabalho do campo, mas por vocês eu prefiro não ter o que vestir mas pagarei escola para vocês estudarem".

AGRADECIMENTOS

A Deus por me abençoar e me fortificar a superar varias complicações de saúde perante este percurso de mestrado e aos meus pais Pedro Gomes Cacabro e Domingas Luís Mendes pelo dom da vida que receberam para me iluminar nesta vida e me dando apoio moral.

Ao Professor Doutor José Maria Da Silva Monteiro Filho, por me orientar em minha dissertação de mestrado, por ser alicerce e modelar esta pesquisa, por ter muita paciência comigo, por ter me amparado nos diversos momentos de declínio que tive nesse período e por ser meu professor, ensinando-me a dar passos para não ficar no mesmo lugar.

Ao professor Doutor José Wellington Franco da Silva, por me coorientar nesta dissertação de mestrado, por participar na estruturação da dissertação, pelos conselhos e apoio moral, e por me ter fortificado a seguir em frente.

Ao Doutorando Evandro Claudino de Sá, por estar sempre disponível para esclarecer as minhas dúvidas e conversar sobre melhores métodos para o avanço e credibilizar a pesquisa, pelo apoio estrutural no processo de codificação e implementação do Modelo Arquitetural proposto neste trabalho.

Aos colegas do Laboratório ARIDA: José Caique Militão França e Thiago Gadelha, pela contribuição no avanço e conclusão desta pesquisa.

E ao FUNCAP - Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico, por financiarem os dois anos da minha pesquisa.

"Um passo à frente e você não está mais no
mesmo lugar"

(Prof. Dr. José Maria Da Silva Monteiro Filho)

RESUMO

Durante os últimos anos, a propagação em larga escala de desinformação através das redes sociais tem se tornado uma questão crítica, resultando na diminuição da confiabilidade de informações legítimas, instabilidade social, ameaças à democracia e à saúde pública. Em diversos países em desenvolvimento, como o Brasil, Índia e México, os aplicativos de mensagens instantâneas, incluindo o WhatsApp e o Telegram, têm sido uma das principais fontes de desinformação. Esses aplicativos oferecem um recurso bastante importante: os grupos públicos. Contudo, muitos desses grupos são usados para espalhar desinformação, principalmente como parte de campanhas políticas ou ideológicas bem organizadas. Apesar deste cenário, poucas iniciativas foram desenvolvidas explicitamente para investigar o fenômeno de desinformação nessas plataformas. Neste artigo, propomos um modelo arquitetural para auxiliar o desenvolvimento de *chatbots* especializados na detecção automática de desinformação e na educação midiática. Adicionalmente, seguindo o modelo arquitetural proposto, desenvolvemos um *chatbot* pró-ativo chamado IAraBot, o qual monitora, detecta e alerta, de forma automática e em tempo real, a presença de desinformação. Além disso, o *chatbot* IAraBot fornece orientações e exemplos práticos para que os próprios usuários desenvolvam as capacidades necessárias para, de forma crítica, identificar conteúdos que contenham desinformação. O *chatbot* IAraBot foi desenvolvido para as plataformas *WhatsApp* e *Telegram*.

Palavras-chave: chatbot pró-ativo; detecção automática de desinformação; aprendizado de máquina.

ABSTRACT

In the last few years, the widespread dissemination of misinformation through social platforms has become a critical issue. In several developing countries, such as Brazil, India, and Mexico, instant messaging apps, including WhatsApp and Telegram, have been one of the main sources of misinformation. These apps offer a significant resource: public groups. However, many of these groups are used to spread misinformation, especially as part of well-organized political or ideological campaigns. In this paper, we propose an architectural model, called IAra, to assist in the development of pro-active chatbots for the automatic detection of misinformation and media education. Additionally, following the proposed architectural model, we have developed a proactive chatbot called IAraBot, which detects the presence of misinformation automatically and in real-time. Moreover, the IAraBot provides hints and practical examples for users to develop the necessary skills to critically identify content containing misinformation. The IAraBot was developed for WhatsApp and Telegram platforms.

Keywords: misinformation detection; pro-active chatbots; machine learning.

LISTA DE TABELAS

Tabela 1 – Comparação entre os Trabalhos Relacionados e o <i>Chatbot</i> IAraBot	28
Tabela 2 – Perguntas Utilizadas pelo Módulo <i>Guia de Identificação de Desinformação</i>	35
Tabela 3 – Tecnologias Utilizadas no <i>Chatbot</i> IAraBot.	37

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Modelo Arquitetural	17
2.2	Chatbots	17
2.3	Processamento de Linguagem Natural	19
2.4	Reconhecimento Ótico de Carateres - OCR	21
2.5	Reconhecimento Automática de Fala	23
3	TRABALHOS RELACIONADOS	25
3.1	Chatbots Reativos	25
3.2	Chatbots Pró-ativos	26
3.3	Chatbots para Detecção de Desinformação	27
4	MODELO ARQUITETURAL IARA	29
5	ESTUDO DE CASO	36
5.1	Instanciando o Modelo Arquitetural IARA	36
5.2	Implementação do IAraBot Telegram	39
5.3	Implementação do IAraBot WhatsApp	41
5.4	Exemplos de Utilização do IAraBot Telegram	43
5.4.1	<i>Funcionamento Pró-Ativo:</i>	<i>43</i>
5.4.2	<i>Funcionamento Reativo:</i>	<i>44</i>
6	AVALIAÇÃO DE USABILIDADE	48
6.1	Teste de Usabilidade	48
6.1.1	<i>Configuração da Avaliação de Usabilidade</i>	<i>49</i>
6.1.2	<i>Aplicação de Questionários</i>	<i>49</i>
6.1.2.1	<i>Net Promoter Score(NPS)</i>	<i>49</i>
6.1.2.2	<i>Cálculo do NPS</i>	<i>50</i>
6.1.2.3	<i>System Usability Scale (SUS)</i>	<i>50</i>
6.1.3	<i>Cálculo do SUS</i>	<i>51</i>
6.1.4	<i>População</i>	<i>52</i>
7	RESULTADOS	54
7.1	Resultados de Teste de Usabilidade NPS	54

7.2	Resultados de Teste de Usabilidade SUS	55
7.3	Avaliações das Funcionalidades	56
7.3.1	<i>Avaliação sobre o mini-curso com as 7 DICAS</i>	60
8	CONCLUSÕES E TRABALHOS FUTUROS	64
	REFERÊNCIAS	66

1 INTRODUÇÃO

O conceito de desinformação é definido por (SU *et al.*, 2020) como informação deturpada de forma geral, seja uma informação forjada, enganosa, falsa ou distorcida. Essa definição bastante abrangente engloba diversos conceitos específicos encontrados na literatura e que podem se sobrepor, tais como *Fake News* (LAZER *et al.*, 2018a), rumores (SHU *et al.*, 2017), enganação (MAALEJ, 2001) e outros, conforme ilustra a Figura 1. Uma síntese desses múltiplos conceitos, é apresentada a seguir:

- **rumores e boatos**: não se originaram de eventos de notícias, porém são aceitos publicamente;
- **engano (deception)**: uma mensagem enganosa é aquela transmitida conscientemente por um remetente para fomentar uma crença ou conclusão falsa por parte do receptor;
- **fake news**: artigos de notícias que são criados intencionalmente e comprovadamente falsos com a finalidade de enganar os leitores;
- **troles e embustes (hoaxes)**: são motivados apenas por diversão ou para enganar indivíduos direcionados;
- **spams**: comumente associados a *e-mails* não desejados, os *spams* constituem qualquer campanha publicitária que chega aos leitores por mídia sociais sem que sejam desejadas;
- **caça-cliques (clickbait)**: empregam imagens em miniaturas ou manchetes sensacionalistas, no processo convencimento de usuários a acessarem e compartilharem conteúdos duvidosos, mais se assemelham a um tipo de propaganda falsa;
- **teorias de conspiração**: entendimentos e narrativas que partem de quatro pressupostos básicos: 1) a realidade (o mundo ou um fato) é diferente do que parece; 2) há um ocultamento da verdade por parte de entidades poderosas; 3) esta percepção, ou teoria, é aceita apenas por uma minoria e 4) estas crenças não são passíveis de sustentação em evidências científicas;
- **sátiras e paródias**: pelo conteúdo humorístico embutido, usando sarcasmos e ironias, é factível de ter seu caráter enganoso identificado;
- **desinformação (misinformation)**: conteúdo criado involuntariamente, sem uma origem ou intenção específica de desorientar o leitor;
- **contra-informação (disinformation)**: são informações criadas com intenção específica de confundir o leitor;

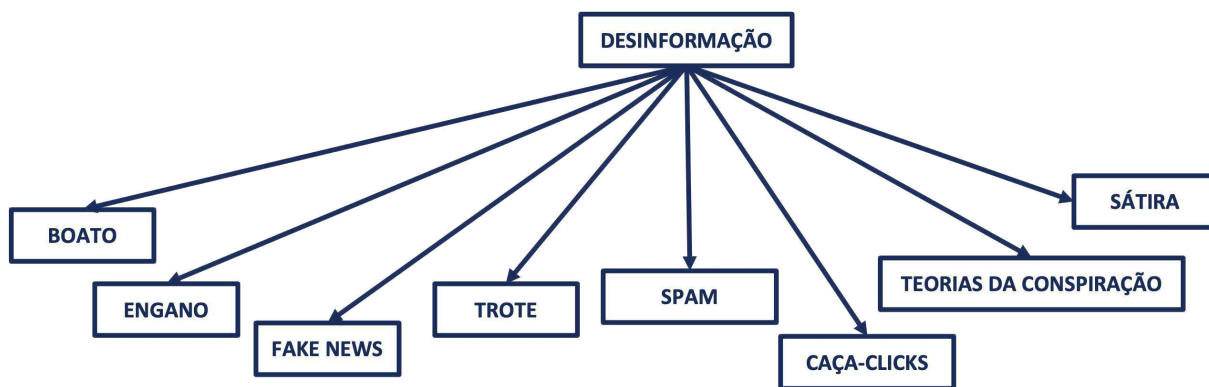


Figura 1 – Conceitos Específicos Relacionados a Desinformação.

Fonte: (SU *et al.*, 2020)

Vale ressaltar que, no Brasil, principalmente, alguns pesquisadores divergem dessa diferenciação entre desinformação (*misinformation*) e contra-informação (*disinformation*). Para estes estudiosos a desinformação é criada com intenções maliciosas para manipular a opinião pública, prejudicar indivíduos, organizações ou grupos sociais e obter ganhos econômicos ou políticos (INTERVOZES, 2019). Assim, a desinformação pode ser entendida como o processo de produção intencional de um ambiente comunicacional pautado por informações falsas, equivocadas ou descontextualizadas para provocar desordem comunicacional (CUNHA *et al.*, 2021). Nesta dissertação utilizaremos o conceito de desinformação (*misinformation*) apresentado em (SU *et al.*, 2020). É importante destacar ainda que, em particular, o termo *Fake News*, embora descreva um tipo específico de desinformação escrita de modo a imitar o estilo de uma notícia jornalística, se tornou muito presente na cultura popular e muitas vezes é utilizado como sinônimo de desinformação de forma indiscriminada.

Nos últimos anos, a ascensão dos aplicativos de troca de mensagens instantâneas alterou significativamente o modo como produzimos, compartilhamos e consumimos informação. O aplicativo WhatsApp é muito popular no Brasil, com mais de 165 milhões de usuários (SÁ *et al.*, 2023b). Além disso, 61% dos usuários de WhatsApp no Brasil abrem o aplicativo pelo menos uma vez ao dia, sendo que, desses, 35% deixam o WhatsApp aberto o dia todo¹. Por outro lado, em apenas um ano, a proporção de *smartphones* com Telegram instalado cresceu no Brasil de 45% para 60% em 2022 (SÁ *et al.*, 2023a)². A popularidade desses aplicativos se deve à sua versatilidade e facilidade de uso. Eles possibilitam o compartilhamento instantâneo de diferentes tipos de mídia, como textos, links, imagens, áudios e vídeos.

¹ <https://blog.opinionbox.com/pesquisa-whatsapp-no-brasil/>

² <https://www.mobiletime.com.br/noticias/31/08/2022/telegram-esta-em-65-dos-smartphones-brasileiros/>

Adicionalmente, as plataformas de mensagens instantâneas fornecem um recurso particularmente importante: grupos de bate-papo públicos. Esses grupos públicos são acessíveis por meio de *links* de convite e, geralmente, possuem temas específicos para discussão, como política, esporte, finanças ou educação, por exemplo. Tanto o WhatsApp quanto o Telegram permitem que um usuário participe de centenas de grupos. Por meio desses grupos, um determinado conteúdo pode chegar a milhares de pessoas em um curto espaço de tempo (SÁ *et al.*, 2023b). A Figura 2 ilustra a evolução da popularidade dos aplicativos de mensagens instantâneas no Brasil.

Todavia, essas plataformas apresentam pouca ou nenhuma moderação, o que tem contribuído para a disseminação em massa de desinformação, principalmente, por meio dos grupos de bate-papo públicos, especialmente como parte de campanhas políticas ou ideológicas articuladas. Neste contexto, a desinformação se espalha de forma mais rápida, profunda e ampla nas plataformas sociais do que a informação legítima (LAZER *et al.*, 2018b). Além disso, devido ao alto volume de informações a que estamos expostos ao usar as plataformas sociais, os humanos têm uma capacidade limitada de distinguir informações verdadeiras de informações falsas (VOSOUGHI S.; ARAL, 2018; QIU *et al.*, 2017; RESENDE *et al.*, 2019). Segundo (GUO *et al.*, 2019), a disseminação da desinformação é um problema social de nível global, causando danos a democracia, justiça, economia, saúde e segurança públicas. Em 2020, as Nações Unidas (ONU) declararam que existe uma “perigosa epidemia de desinformação”³. Estudos recentes indicam que a disseminação de desinformação em larga escala por meio das plataformas sociais tem sido usada para alterar cenários políticos, promover a disseminação de doenças e até mesmo causar mortes (JUNIOR *et al.*, 2021).

Neste cenário, monitorar o conteúdo que circula em grupos públicos e alertar rapidamente os usuários sobre a presença de desinformação é uma tarefa fundamental. Assim, diversas abordagens para a detecção automática de desinformação por meio de métodos computacionais têm sido propostas (CABRAL *et al.*, 2021; MARTINS *et al.*, 2021a; MARTINS *et al.*, 2021b; COUTO *et al.*, 2022; REIS; BENEVENUTO, 2021; REIS *et al.*, 2020; SU *et al.*, 2020). Algoritmos de detecção de desinformação podem ser utilizados para identificar precocemente conteúdo enganoso tão rápido quanto estes surgem nas redes sociais, alertando jornalistas, entidades ou as próprias plataformas para que seja realizada uma checagem de fatos e uma possível mitigação dos danos, seja bloqueando a disseminação do conteúdo enganoso ou alertando usuários da falsidade do mesmo.

³ UN. “Hatred going viral in ‘dangerous epidemic of Misinformation’ during COVID-19 pandemic”. 14 April, 2020. Available in: <https://news.un.org/en/story/2020/04/1061682>. Accessed on: 25 April. 2020.

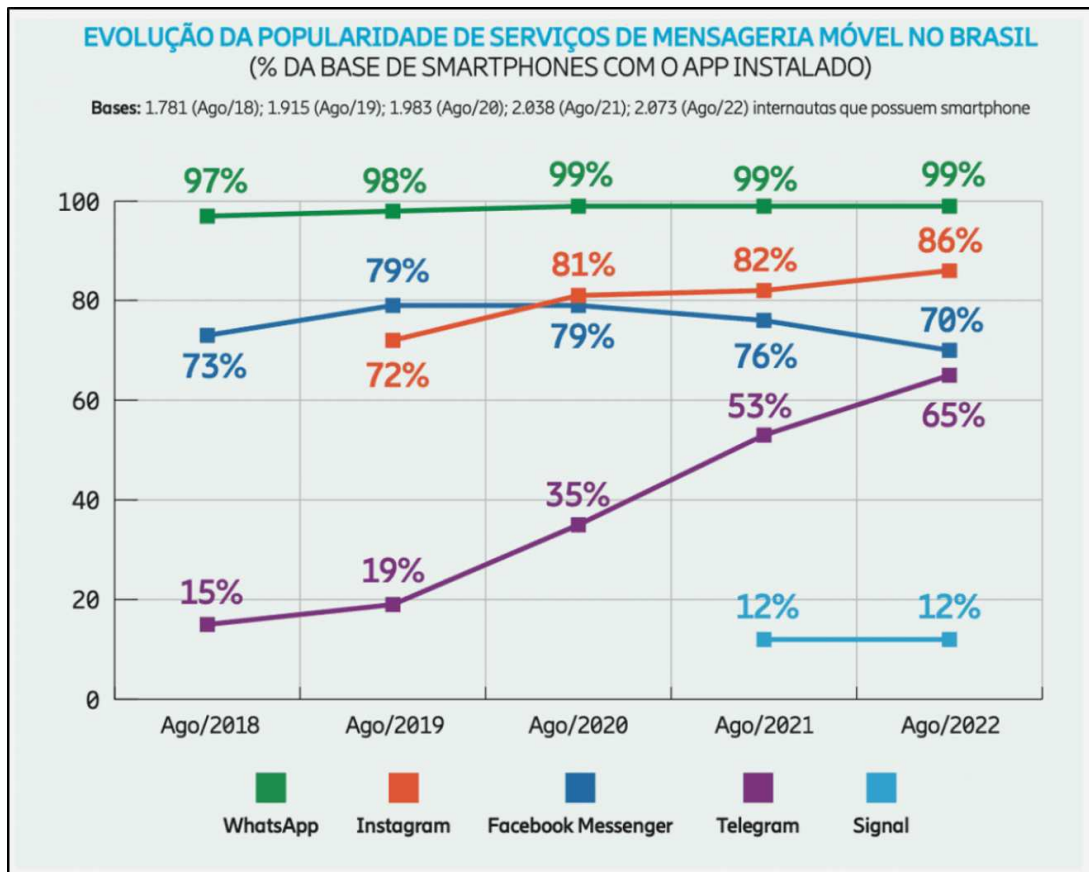


Figura 2 – Evolução da Popularidade dos Serviços de Mensageria no Brasil.
Fonte: Sinch Engage (<https://engage.sinch.com/pt-br/blog/whatsapp-no-brasil/>)

Contudo, é notório que o problema da disseminação de desinformação é extremamente complexo, cuja compreensão e solução é intrinsecamente interdisciplinar, permeando as ciências sociais, a psicologia, o jornalismo, dentre outros campos de estudo (GUPTA *et al.*, 2022). Assim, uma outra tarefa essencial consiste na formação de cidadãos críticos, capazes de lidar com um ambiente digital marcado pela desinformação, o que tem sido chamado de educação midiática (ADJIN-TETTEY, 2022). Pode-se definir educação midiática como o conjunto de conhecimentos, capacidades e competências (incluindo processos para sua aquisição) relativas ao acesso, uso esclarecido, pesquisa e análise crítica da mídia, bem como as capacidades de expressão e de comunicação por meio dessas mesmas mídias (NAGUMO *et al.*, 2022). Portanto, educação midiática é um conceito amplo, que abarca o desenvolvimento de uma série de habilidades voltadas à leitura, análise e produção de informação autêntica e de qualidade (ALMEIDA *et al.*, 2022).

Neste trabalho, propomos um modelo arquitetural, denominado IAra, para auxiliar o desenvolvimento de *chatbots* especializados na detecção automática de desinformação e na educação midiática. Adicionalmente, seguindo o modelo arquitetural proposto, desenvolvemos um *chatbot* chamado IAraBot, o qual pode atuar tanto de forma pró-ativa quanto reativa. Para atuar de forma pró-ativa o IAraBot precisa ser adicionado em um determinado grupo de bate-papo. Em seguida, ele irá monitorar e analisar automaticamente o conteúdo que trafega no grupo. Por fim, caso o IAraBot detecte que um determinado conteúdo tem uma elevada probabilidade de conter desinformação, uma mensagem de alerta é enviada no grupo. Para a detecção automática de desinformação o IAraBot utiliza o método proposto em (CABRAL *et al.*, 2021), o qual obteve para a métrica *F1-Score* um valor de 0,87. Adicionalmente, o IAraBot também pode atuar de forma reativa. Neste caso, um(a) utente inicia um diálogo diretamente com o IAraBot e envia um determinado conteúdo para ser analisado. O IAraBot consegue analisar conteúdos em quatro formatos distintos: texto, imagem, áudio e vídeo. Após esta análise, o IAraBot informa a probabilidade do conteúdo recebido conter desinformação. De forma complementar, o IAraBot fornece orientações e exemplos práticos para que os próprios utentes desenvolvam as capacidades necessárias para, de forma crítica, identificar conteúdos que contenham desinformação. O IAraBot foi desenvolvido para as plataformas *WhatsApp* e *Telegram*.

O restante desse trabalho está organizado da seguinte forma. O Capítulo 2 apresenta os principais conceitos utilizados no decorrer desta dissertação. Os principais trabalhos relacionados são discutidos no Capítulo 3. Já no Capítulo 4 o modelo arquitetural IAra é apresentado em detalhes. No Capítulo 5 um estudo de caso envolvendo a utilização do *chatbot* IAraBot é discutido. Uma avaliação da usabilidade do IAraBot é apresentada no Capítulo 6. Por fim, o Capítulo 7 conclui esta dissertação e aponta caminhos para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos que foram utilizados durante o desenvolvimento dessa pesquisa. Inicialmente iremos discutir os fundamentos teóricos que nortearam a concepção do modelo arquitetural IAra. Em seguida, discutiremos as ferramentas que foram utilizadas na construção do *chatbot* IAraBot.

2.1 Modelo Arquitetural

Um modelo arquitetural (MA) é um projeto ou representação visual que descreve a estrutura e organização de um sistema complexo, como um software, uma rede de computadores ou um edifício (WORTMANN *et al.*, 2020). Ele define os principais componentes do sistema e as relações entre eles, bem como as interfaces e protocolos de comunicação necessários para que esses componentes interajam entre si. Os modelos arquiteturais são usados na engenharia de software e outras disciplinas para ajudar a compreender e visualizar a estrutura e o comportamento de um sistema complexo, bem como para guiar o seu processo de desenvolvimento.

Mais especificamente, um modelo arquitetural deve conter minimamente as seguintes informações:

- **Objetivos e restrições:** descrição de objetivos e restrições (requisitos funcionais e não-funcionais) que impactam significativamente a arquitetura do sistema.
- **Descrição da arquitetura:** descrição da proposta de arquitetura, com diagramas de camadas, módulos e demais abstrações.
- **Decisões e justificativas:** as decisões que devem nortear o desenvolvimento da aplicação, com base na arquitetura definida, além de justificativas de como essa arquitetura atende os objetivos e as restrições identificados previamente.

2.2 Chatbots

Chatbot, ou simplesmente *bot*, é um software que responde à uma entrada em linguagem natural e busca manter uma conversa de forma a imitar um ser humano (RESHMI; BALAKRISHNAN., 2016). Geralmente, eles são projetados para responder perguntas, fornecer informações e executar tarefas simples de forma autônoma.

O termo *chatbot* surgiu da junção das palavras *chatter* (a pessoa que conversa) e da palavra *bot* (abreviatura de *robot*), ou seja, um robô (em forma de software) que conversa com as pessoas. A palavra foi inventada por Michael Mauldin (Criador do primeiro *Verbot*, Julia) em 1994, para descrever estes robôs de conversação na *Twelfth National Conference on Artificial Intelligence*.

O software ELIZA foi o primeiro *chatbot* da história e foi criado por Joseph Weizenbaum no laboratório de Inteligência Artificial do MIT, entre os anos de 1964 e 1966. A ideia básica é simular a conversação entre homem e máquina. O nome ELIZA vem da personagem principal da peça de teatro “Pigmalião”, Eliza Doolittle, escrita por George Bernard Shaw, em 1913. A principal instânciação do ELIZA simula uma conversa entre um paciente e seu psicólogo, na qual o usuário é o paciente e o software o psicólogo.

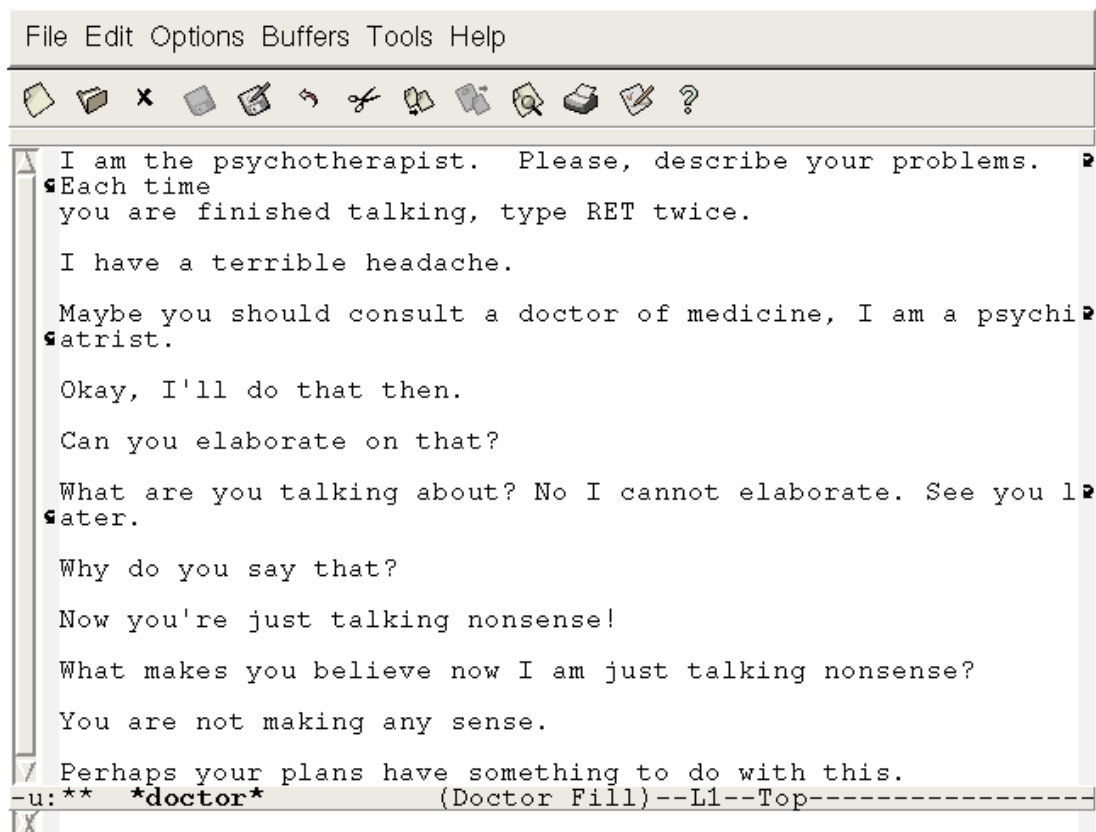


Figura 3 – Chatbot ELIZA Conversando com um Interlocutor Humano em Emacs.

Fonte: Wikipedia

Os *chatbots* podem servir a diferentes propósitos, tais como: atendimento ao cliente, suporte emocional, entretenimento, apoio a processos educacionais, suporte na utilização de equipamentos, dentre outros. Eles podem ser integrados em diferentes plataformas, como sites, aplicativos de mensagens, redes sociais e assistentes virtuais, para oferecer um atendimento rápido e eficiente aos usuários. Os *chatbots* podem ser desenvolvidos com diferentes níveis de complexidade, sendo classificados em dois tipos principais: baseados em regras e baseados em Inteligência Artificial (IA). *Chatbots* baseados em regras possuem fluxos de navegação pré-definidos e palavras-chave parametrizadas (que determinam as respostas automáticas) de acordo com o domínio da aplicação. *Chatbots* baseados em IA utilizam aprendizado de máquina e processamento de linguagem natural para identificar os contextos das conversas e as intenções dos usuários, encontrar ou elaborar as respostas mais adequadas, além de aprender com as interações dos usuários e melhorar suas respostas ao longo do tempo.

Os *chatbots* podem ser classificados usando diferentes aspectos: o domínio do conhecimento, o serviço prestado, os objetivos a serem alcançados, o processamento de entrada e o método de geração de resposta, além do método de construção. A classificação com base nos objetivos considera o objetivo principal que o *chatbot* deve alcançar. Neste sentido, os *chatbots* podem ser classificados em: informativos, conversacionais e baseados em tarefas. Os *chatbots* informativos são projetados para fornecer ao usuário informações armazenadas antecipadamente ou disponíveis em uma fonte de dados fixa. Esta categoria inclui os *chatbots* construídos para responder perguntas frequentes (*Frequently Asked Questions* - FAQ). Um *chatbot* conversacional busca conversar com um ser humano e seu objetivo é responder corretamente à uma frase fornecida por um usuário. Os *chatbots* baseados em tarefas procuram solucionar uma tarefa específica, como encontrar um voo, reservar um restaurante ou até mesmo comprar e vender ações em uma bolsa de valores.

2.3 Processamento de Linguagem Natural

Processamento de Linguagem Natural - PLN (*Natural Language Processing* - NLP) é uma subárea da ciência da computação, inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de línguas humanas naturais. De maneira didática, podemos definir linguagem natural como sendo uma linguagem desenvolvida e evoluída por humanos por meio do uso natural e da comunicação. Assim, a linguagem natural difere daquelas criadas artificialmente, como as linguagens de programação, por exemplo.

Mais precisamente, o Processamento de Linguagem Natural (também denominado Linguística Computacional ou, ainda, Processamento de Línguas Naturais) busca investigar, propor e desenvolver formalismos, modelos, técnicas, métodos e sistemas computacionais para solucionar problemas relacionados à automação da interpretação e da geração da língua humana, como o inglês ou o português. As principais aplicações do PLN envolvem áreas, tais como: Tradução Automática de Textos, Sumarização Automática de Textos, Ferramentas de Auxílio à Escrita, Perguntas e Respostas, Categorização Textual, Recuperação e Extração de Informação, Análise Morfo-sintática, Análise Semântica e Análise de Sentimentos.

A Análise de Sentimentos (AS), também conhecida como Mineração de Opiniões, pode ser definida como um estudo computacional envolvendo opiniões, sentimentos, avaliações, atitudes, afeições, visões, emoções e subjetividade, expressos de forma textual. Esse é um campo em NLP, que visa identificar a polaridade do sentimento (negativo, positivo ou neutro) ou emoção (por exemplo, raiva, antecipação, nojo, medo, alegria, tristeza, surpresa, confiança, etc.) presente em um texto. A análise de sentimento também é usada para detecção de toxicidade, sarcasmo, análise multilíngue de sentimentos, entre outras aplicações complexas. A tarefa de análise de sentimentos pode ser estruturada em três etapas: i) identificar as opiniões expressas sobre determinado assunto ou alvo em um conjunto de documentos; ii) classificar a orientação semântica ou a polaridade dessa opinião em positiva ou negativa; e iii) apresentar os resultados de forma agregada e sumarizada.

Outra área que tem recebido bastante atenção nos últimos anos é a detecção de desinformação e identificação de discurso de ódio. A primeira tem por finalidade identificar conteúdo enganoso divulgado digitalmente, enquanto a segunda busca identificar um tipo de linguagem que não é aceitável e que pode ser prejudicial às pessoas. Outro problema relevante é a modelagem de tópicos, que tem por objetivo identificar estruturas semânticas compartilhadas entre textos para determinar quais eventos, conceitos ou assuntos estão sendo discutidos em um conjunto de documentos. Para isso, parte-se da premissa de que cada documento pode ser representado por uma mistura de tópicos, sendo cada tópico composto por palavras que melhor o define no corpus analisado. Por exemplo, em um corpus de notícias de várias fontes, os tópicos poderiam representar eventos que ocorreram (por exemplo, confronto entre Rússia e Ucrânia, confronto entre Israel e Hamas, etc.) ou delinear grandes temas (e.g., economia, política, educação, etc.). A principal tarefa na modelagem de tópicos, portanto, é descobrir padrões de uso de palavras e relacionar documentos que compartilham padrões semelhantes.

2.4 Reconhecimento Ótico de Carateres - OCR

OCR (*Optical Character Recognition*, em português, Reconhecimento Óptico de Caracteres) é uma tecnologia utilizada para reconhecer e extrair textos de arquivos digitais de diferentes formatos, tais como imagens, fotos, documentos digitalizados, dentre outros (MEMON *et al.*, 2020). Vale destacar que os textos obtidos podem ser editados, pesquisados e armazenados. Assim, OCR pode ser definido como a análise de caracteres em imagens digitais e a posterior tradução do texto encontrado para o formato ASCII, o qual possibilita que o texto digitalizado seja armazenado e manipulado eletronicamente de forma mais eficiente. O Reconhecimento Óptico de Caracteres costuma ser considerado um campo de pesquisa multidisciplinar, envolvendo as áreas de reconhecimento de padrões, inteligência artificial e visão computacional.

O OCR ganhou popularidade em 1950, quando David Shepard e Louis Tordella pesquisavam uma forma de automatizar dados da Agência de Segurança das Forças Armadas. Auxiliados por Harvey Cook, eles criaram o primeiro software de OCR, conhecido como Gismo. Em seguida, Shepard fundou a IMR, empresa que se tornou pioneira no desenvolvimento de softwares comerciais de OCR. Após 3 anos do primeiro software, a IBM, adquiriu uma licença da IMR e desenvolveu seu próprio software que ficou conhecido como *Optical Character Recognition* surgia, dessa forma, a sigla OCR.

As ferramentas de OCR funcionam analisando um arquivo de entrada e comparando seus caracteres com fontes armazenadas em seu banco de dados e/ou reconhecendo características típicas de determinado caractere. As técnicas mais tradicionais de OCR geralmente aplicam filtros ao arquivo de entrada para fazer com que os caracteres sejam evidenciados, e, em seguida, usam algoritmos de classificação para identificar os caracteres. Quando o texto presente no arquivo de entrada está padronizado, em geral, essas atividades são suficientes. Caso contrário, técnicas mais avançadas são necessárias. Algumas técnicas de OCR interessante são: O EAST (*Efficient Accurate Scene Text detector*) e SEE (*Semi-Supervised End-to-End Scene Text Recognition*). Atualmente, diversas APIs (*Application Program Interface*) podem ser utilizadas para acessar serviços de OCR disponibilizados de forma *online*, dentre elas destacam-se: Google Cloud Vision, Microsoft Computer Vision, Sema Media Data, Taggun e Cloudmersive.

Segundo (MAIA *et al.*, 2022), a maioria dos pesquisadores utilizam os seguintes passos para construir sistemas de OCR:

- Varredura óptica: é a captura da imagem via câmera digital ou um *scanner*.
- Pré-Processamento: é a etapa de “limpeza” da imagem, nela são aplicadas técnicas de conversão de escala para cinza, binarização, normalização, distorção, ajuste de iluminação e brilho, com o objetivo de aprimorar a imagem para as etapas seguintes.
- Segmentação: após receber uma imagem limpa e sem ruídos da fase anterior, a segmentação organiza-se em três etapas: primeiro divide-se os caracteres em linha horizontal, em seguida divide-se as palavras de cada linha e, por fim, separa-se os caracteres de cada palavra.
- Extração de características: nesta etapa, espera-se que possam ser extraídos padrões significantes da imagem, que possam distinguir caracteres e dar sentido na próxima fase.
- Reconhecimento e treinamento: aqui são utilizadas técnicas estatísticas, modelos de correspondência e até mesmo redes neurais artificiais. Nesta fase, o problema de reconhecimento de vocabulário incompleto deve ser resolvido.
- Pós-processamento: nesta fase há detecção de erros para serem corrigidos, bem como o agrupamento de símbolos para transformá-los em caracteres.

Para reconhecimento de caracteres, as imagens passam primeiramente por um tratamento como mencionado anteriormente e depois deve-se usar as técnicas para detecção de caracteres. Segundo (MELLO; SANTOS., 2018), as técnicas para detecção de texto podem ser categorizadas em dois grupos principais: abordagem baseada em geometria e a baseada em textura. A abordagem baseada em geometria faz uso de técnicas de processamento de imagens para extrair e modelar o texto analisando a disposição de bordas e buscando por regiões que contenham caracteres. Por outro lado, a abordagem baseada em textura tenta identificar locais onde tenham caracteres levando em consideração que as letras possuem como preenchimento um padrão de textura. Como por exemplo, pode-se assumir que o texto não vai possuir um gradiente muito acentuado em sua coloração.

2.5 Reconhecimento Automática de Fala

O reconhecimento automático de fala (RAF), ou, em inglês, *Automatic Speech Recognition* (ASR), é uma área interdisciplinar originária da linguística computacional que tem por finalidade possibilitar que computadores interpretem a fala humana a partir da síntese de sinais de áudio, ou seja, é uma técnica que possibilita transcrever áudio em texto expresso em linguagem natural (AGUIAR DE LIMA; DA COSTA-ABREU, 2020). Este processo geralmente apresenta enormes desafios, tais como: tamanho do vocabulário, diversidade de idiomas, reconhecimento de fala contínua e a estrutura complexa da voz humana, que depende de diversos fatores, tais como: entonação, velocidade da voz, sotaque e estado emocional.

Técnicas de reconhecimento de fala são construídas, geralmente, a partir de decodificadores, além da modelagem acústica e de linguagem. Os decodificadores desempenham o papel de interpretação do sinal de áudio. Já os modelos acústicos e de linguagem são responsáveis pelo mapeamento de toda a estrutura linguística. O principal desafio envolvendo uma técnica de reconhecimento de fala consiste em obter um modelo acústico e de linguagem que proporcione um desempenho satisfatório. Vale destacar que esses modelos são treinados utilizando uma base de dados pré-processada e rotulada (aprendizado supervisionado), além de técnicas de descrição probabilística para os termos da linguagem. Algumas técnicas populares de RAF são *Time delay neural network* (TDNN) e *Wav2Vec2*. Atualmente, diversas APIs (*Application Program Interface*) podem ser utilizadas para acessar serviços de RAF disponibilizados de forma *online*, dentre elas destacam-se: Google Speech-to-Text, IBM Watson Speech to Text, Amberscript, AssemblyAI, Rev.ai e Scriptix.

A maioria dos sistemas RAF utiliza uma abordagem que converte um sinal de áudio de entrada em uma sequência de *frames* de características com igual duração (SANTOS., 2016). Por trás desses sistemas podemos identificar quatro grandes componentes:

- **Modelo Acústico:** mapeia o sinal original que está sendo processado em palavras e sentenças;
- **Modelo de Linguagem:** responsável por caracterizar a língua e condicionar a combinação de palavras, descartando frases agramaticais;
- **Léxico:** representa as palavras do dicionário utilizado, com suas respectivas transcrições fonéticas e
- **Decodificador:** procura a melhor sequência de palavras num conjunto de hipóteses possíveis (FERREIRA; SOUZA., 2008).

A Figura 4 representa a arquitetura genérica de um sistema de reconhecimento de fala. Neste trabalho utilizamos o *framework Google Speech-to-Text* para reconhecer entrada dos áudios e transcreve-los. Este *framework* permite converter a voz presente num sinal de áudio para uma sequência de palavras em texto, e para isto, o sinal é segmentado em expressões (utterances) que possam ser decodificadas, recorrendo-se normalmente à distinção entre a fala e o silêncio para este efeito (PALMA; OLIVEIRA., 2019). Veja essa ilustração na Figura 5.

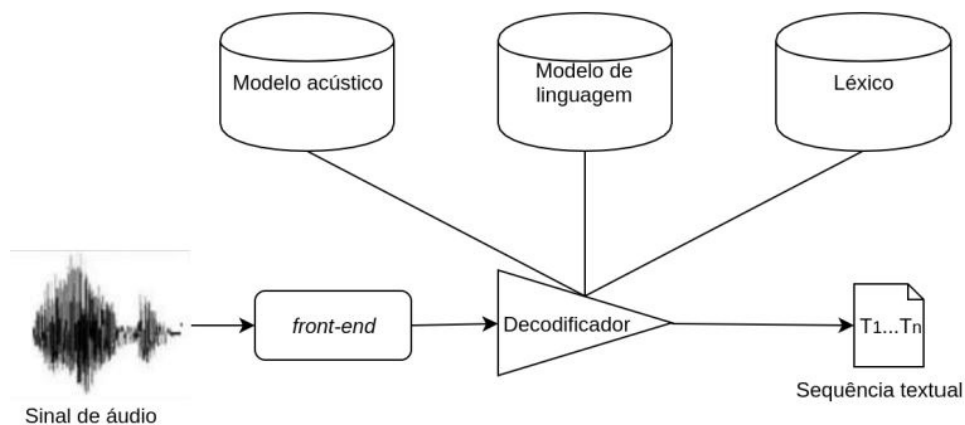


Figura 4 – Arquitetura de Sistemas ASR.
Fonte: (FERREIRA; SOUZA., 2008)

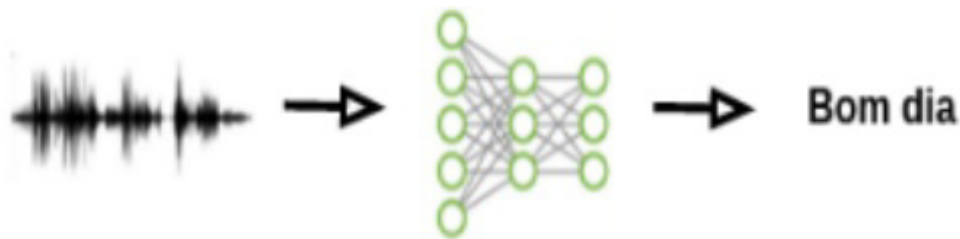


Figura 5 – Reconhecimento de Voz.
Fonte: (DöRR., 2022)

3 TRABALHOS RELACIONADOS

Nesta seção, discutiremos os principais trabalhos relacionados a esta dissertação. Para um melhor entendimento, organizamos tais iniciativas em três categorias: *chatbots* reativos, *chatbot* pró-ativos e *chatbot* para detecção de desinformação. Vale destacar que um *chatbot* pró-ativo possui a capacidade de tomar uma determinada ação sem necessariamente ser requisitado, enquanto um *chatbot* reativo apenas responde a iterações iniciadas pelos utentes.

3.1 Chatbots Reativos

Em (KHAN; ALBATEIN., 2021), os autores propuseram um *chatbot* para WhatsApp capaz de fornecer informações adequadas sobre a COVID-19 com base nos sintomas relatados pelo usuário, além de indicar postos de saúde mais próximos. Adicionalmente, o *bot* orienta o usuário sobre os primeiros socorros ou cuidados imediato que devem ser tomados durante uma situação de contágio.

Um *chatbot* chamado PipaBot foi criado para o Projeto Infância e Poluentes Ambientais (PIPA UFRJ) (PAULA *et al.*, 2019), o qual acompanha mães e seus bebês com o objetivo de avaliar os efeitos de poluentes ambientais na fase de gestação até o início da infância (quatro anos). O PipaBot tem por finalidade ser um canal de comunicação entre os responsáveis do projeto PIPA e as mães, sendo capaz de dialogar com elas a respeito de dúvidas e informações sobre o projeto, consultas, resultados de exames médicos, dentre outras. O PipaBot foi desenvolvido utilizando-se o *framework* Botpress que integra conceitos de IA e possibilita sua integração ao Messenger do Facebook e ao website do projeto PIPA UFRJ. Ele foi avaliado usando o modelo de aceitação da tecnologia (TAM - *Technological Acceptance Model*)

Já em (SILVA *et al.*, 2021), os autores discutiram o desenvolvimento de um *chatbot* baseado em regras para auxiliar a comunidade do Instituto Federal de Brasília (IFB) a encontrar, no portal da instituição, os resultados das solicitações realizadas previamente pelos usuários. O trabalho apresentado em (CRUZ *et al.*, 2020) propõe um Modelo Arquitetural (MA) para construção de Agentes Conversacionais (ACs) especializados na busca e recuperação de informações institucionais a partir de documentos não estruturados (ex.: PDF, HTML, etc.). Adicionalmente, os autores apresentaram uma instância do MA proposto, que consiste em um Agente Conversacional *online* para auxiliar técnicos, professores e alunos da Universidade Federal de São João del-Rei (UFSJ) na consulta e busca por informações relacionados à resoluções e regimentos

internos dessa instituição. Por fim, o agente criado foi avaliado por meio dos métodos SUS (*System Usability Scale*).

Em (ALMEIDA; ARAÚJO, 2021), os autores apresentam um *chatbot* intitulado AnneBot, o qual tem por finalidade auxiliar os alunos no processo de ensino e aprendizagem do Pensamento Computacional (PC). O AnneBot foi desenvolvido na linguagem de programação Python e disponibilizado no aplicativo mensageiro Telegram. Ele possibilita que o estudante esclareça suas dúvidas por meio de um conjunto de perguntas e respostas. Já em (LOPEZ *et al.*, 2021), os autores apresentam o ANTIE *chatbot*, o qual promove a educação BLM (*Black Lives Matter*), ou seja, o ativismo pela igualdade racial, fornecendo informações sobre eleições, notícias relevantes, terminologias, recomendações de mídia, listas de empresas locais de propriedade de negros, dentre outras. Em (ZHI; METOYER, 2020), os autores apresentam o GameBot, um *chatbot* interativo para fãs de esportes, o qual permite explorar dados e estatísticos sobre jogos. O GameBot apresenta (i) respostas diretas às perguntas do usuário e (ii) visualizações de dados como contexto de suporte para que o usuário elabore suas perguntas.

Já em (ATHREYA *et al.*, 2018), os autores apresentam o DBpedia *chatbot*, um *bot* baseado em grafos de conhecimento para fornecer respostas às perguntas mais frequentes dos usuários. A DBpedia é uma base de conhecimento semântica que extrai informações estruturadas a partir da Wikipédia e as disponibiliza em um formato padronizado e acessível à máquina. O desenvolvimento do DBpedia *chatbot* enfrentou quatro desafios principais: (1) entender as consultas do usuário, (2) buscar informações relevantes com base nas consultas, (3) adaptar as respostas com base nos padrões de cada plataforma de saída (ou seja, Web, Slack, Facebook) e (4) desenvolver as interações subsequentes com o usuário.

Em (PUTRI *et al.*, 2020), os autores apresentam um *chatbot* interativo, inteligente e personalizado utilizando AIML e Google Flutter, o qual foi denominado *chatbot-hote*. Este *chatbot* inclui todos os serviços nos hotéis e é capaz de oferecer esses serviços de forma personalizada aos clientes antes que estes cheguem ao hotel. Ele pode acomodar as necessidades do cliente e aumentar as vendas das instalações do hotel.

3.2 Chatbots Pró-ativos

Existem vários trabalhos que buscam desenvolver *chatbots* proativos, em geral, com a finalidade de coletar dados. Um *chatbot* proativo possui a capacidade de tomar uma determinada ação sem necessariamente ser requisitado, como, por exemplo, iniciar uma conversa. Em

(KATTENBECK *et al.*, 2016), os autores projetaram, implementaram e avaliaram o *AIRBOT*, um *chatbot* voltado para *smartphones* que fornece informações aos viajantes durante sua permanência em aeroportos. Em (RESHMI; BALAKRISHNAN, 2016) os autores apresentam um *chatbot* inquisidor, que encontra os dados ausentes na consulta fornecida pelo usuário e realiza uma série de perguntas com a finalidade de coletar os dados necessários para responder corretamente a referida consulta. Por meio desta abordagem, o nível de interatividade entre o usuário e o *chatbot* é aprimorado. Em (YAN; ZHAO, 2018), os autores propõem um novo paradigma generativo de conversação entre humanos e computadores, caracterizado por respostas mais inteligentes com sugestões proativas.

3.3 Chatbots para Detecção de Desinformação

Em (GUPTA *et al.*, 2021), os autores apresentaram um *chatbot* conversacional multi-idioma e multifuncional, denominado *TruthBot*, projetado para buscar a verdade (informações confiáveis e verificadas) sobre tópicos específicos. Ele ajuda os usuários a obter informações acerca de determinados tópicos, a partir de notícias avaliadas por agências de verificação de fatos (*fact-checking*). O *TruthBot* foi inicialmente configurado para buscar informações sobre a COVID-19. Contudo, ele pode ser facilmente personalizado para qualquer outro tópico.

Recente, alguns *chatbots* voltados para o enfrentamento do fenômeno da desinformação foram propostos, dentre eles, destacam-se: *IFCN Chatbot*¹, *Fake News Immunity Chatbot*², *João Fake News*³, *Suve*⁴, *Fátima*⁵ e *FakeCheck*⁶. Todavia, esses *bots* podem ser classificados como reativos, ou seja, precisam ser provocados para, em seguida, enviar uma resposta ou informação.

Na Tabela 1 é ilustrada uma comparação entre os principais *chatbots* desenvolvidos com a finalidade de enfrentar o problema da desinformação e o *chatbot* IAraBot, o qual foi desenvolvido seguindo o modelo arquitetural IAra. A análise comparativa é realizada por meio de diferentes critérios, tais como: i) as plataformas nas quais o *chatbot* está disponível, ii) os idiomas suportados pelo *chatbot*, iii) o desenvolvedor do *chatbot*, iv) a abordagem adotada pelo *chatbot* (pró-ativa ou reativa) e v) se o *chatbot* utilizou ou não um modelo arquitetural.

¹ <https://poy.nu/ifcnchatbotPT>

² <https://fni.arg.tech/>

³ <https://www.showmetech.com.br/unb-whatsapp-joao-fake-news/>

⁴ <https://www.oecd-opsi.org/covid-response/suve-chatbot/>

⁵ <https://www.aosfatos.org/fatima/>

⁶ <http://nilc-fakenews.herokuapp.com/>

Observando Tabela 1, pode-se notar que os trabalhos encontrados na literatura não adotam uma abordagem pró-ativa, sendo esta a principal contribuição do *chatbot* IAraBot. Além disso, a grande maioria dos *chatbots* existentes foi desenvolvida para uma única plataforma. Por outro lado, o *chatbot* IAraBot fornece suporte para duas plataformas diferentes, WhatsApp e Telegram, mantendo exatamente o mesmo comportamento, uma vez que estas instâncias derivam do modelo arquitetural IAra, proposto neste trabalho. Vale destacar ainda que, neste trabalho não estamos propondo nenhuma técnica de aprendizagem de máquina para a detecção automática de desinformação. Para esta finalidade utilizamos o melhor modelo obtido em (CABRAL *et al.*, 2021), mais precisamente uma LSVM (*Linear Kernel Support Vector Machine*), treinada com vetores TF-IDF, usando unigramas, bigramas e trigramas como *tokens*, após a remoção de *stopwords* e aplicação de lematização, o qual obteve para a métrica *F1-Score* um valor de 0,87.

Nome	Plataformas	Língua	Desenvolvedor	Abordagem	M. Arquitetural
TruthBot	WhatsApp, Messenger e Telegram	108 Languages (Google translate)	IIT Kanpur	Reativo	Não
IFCN	WhatsApp	Português	International Fact-Checking Network	Reativo	Não
Fake News Immunity	Web	Inglês	University of Liverpool	Reativo	Não
João Fake News	WhatsApp	Português	Universidade de Brasília	Reativo	Não
Suve	Web	English, Estonian, Russian	OPSI	Reativo	Não
Fátima	WhatsApp	Português	AosFatos	Reativo	Não
FakeCheck	WhatsApp	Português	USP	Reativo	Não
IAraBot	WhatsApp e Telegram	Português	UFC	Reativo e Pró-ativo	Sim

Tabela 1 – Comparação entre os Trabalhos Relacionados e o *Chatbot* IAraBot

4 MODELO ARQUITETURAL IARA

Neste capítulo, iremos apresentar em detalhes o modelo arquitetural IARA, o qual tem por finalidade auxiliar o desenvolvimento de *chatbots* especializados na detecção automática de desinformação e na educação midiática. A Figura ?? ilustra o modelo arquitetural IARA. A seguir, iremos discutir detalhadamente cada componente do IARA.

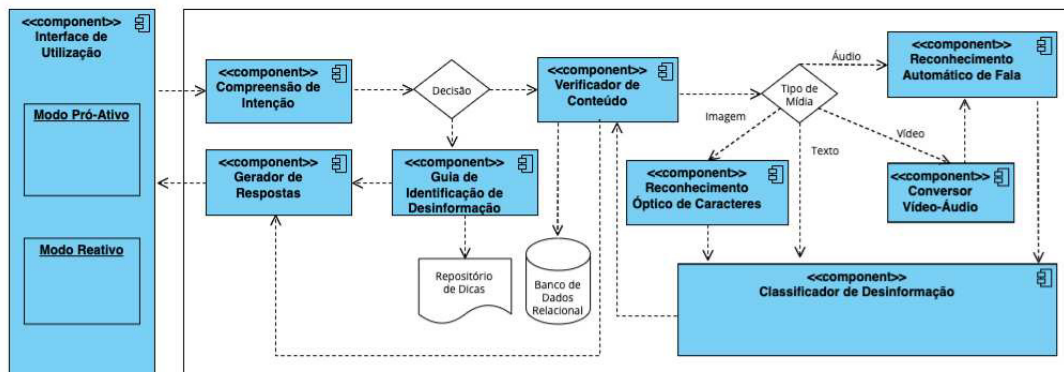


Figura 6 – O Modelo Arquitetural IARA.

O módulo *Interface de Utilização* é responsável por interagir com o usuário, recebendo as ações de entrada e exibindo as respostas produzidas pelo *chatbot*. Logicamente, sua implementação está vinculada ao ambiente utilizado pelo usuário, o qual pode ser, por exemplo, um navegador *web*, um aplicativo como o WhatsApp ou Telegram, dentre outros.

O modelo arquitetural IARA prevê que seja possível habilitar e desabilitar o comportamento pró-ativo do *chatbot*, o que pode ser implementado por meio de um arquivo de configurações, por exemplo. Caso o modo pró-ativo esteja habilitado, o *chatbot* necessita, inicialmente, ser adicionado a um grupo de bate papo. Em seguida, ele irá monitorar e analisar automaticamente o conteúdo que trafega no grupo, incluindo texto, imagem, áudio e vídeo. Por fim, caso o *chatbot* detecte que um determinado conteúdo tem uma probabilidade de conter desinformação acima de um determinado limiar, uma mensagem de alerta é enviada no grupo. O modelo arquitetural proposto indica que este limiar seja definido pelo usuário, por meio de um arquivo de configurações, por exemplo. Caso o modo pró-ativo esteja desabilitado, o *chatbot* funcionará apenas de modo reativo, respondendo a interações iniciadas pelos usuários. Adicionalmente, o *chatbot* deve permitir que o usuário dialogue diretamente com ele, enviando conteúdos para serem avaliados ou ainda desenvolvendo sua capacidade de identificação de desinformação, por meio de um tutorial composto por um conjunto de dicas e exemplos. Desta forma, o funcionamento do módulo *Interface de Utilização* é condicionado pelas configurações

previamente definidas.

O módulo *Compreensão de Intenção* recebe a ação de entrada fornecida pelo usuário e busca compreender a sua intenção. Este componente pode ser implementado com o uso de *frameworks*, tais como o DialogFlow¹, Twilio² e o Watson Assistant da IBM³. A comunicação primordial deve ser textual, ou seja, o usuário comunica-se com o *chatbot* por meio de texto.

Neste contexto, o módulo *Compreensão de Intenção* pode ser desenvolvido seguindo a abordagem por correspondência de padrões. Desta forma, todas as respostas às entradas do usuário podem ser pré-programadas. O módulo *Compreensão de Intenção* classifica a ação solicitada pelo usuário em dois grandes contextos: i) verificação de conteúdo para detecção automática de desinformação e ii) educação midiática, ou seja, o desenvolvimento das habilidades necessárias para a identificação de desinformação.

Se a ação de entrada do usuário tiver uma intenção relacionada à verificação de conteúdo para detecção automática de desinformação, esta é repassada para o módulo *Verificação de Conteúdo*. Neste cenário, é necessário perceber qual o tipo de conteúdo o usuário deseja verificar: texto, imagem ou áudio. Esta tarefa é realizada pelo componente *Tipo de Mídia*.

Caso o conteúdo a ser analisado seja um texto, este deverá ser enviado para o módulo *Classificador de Desinformação*. Este módulo possui um classificador binário (ou seja, que possui duas classes, mais precisamente: “Desinformação” X “Não Desinformação”), isto é, um modelo de aprendizado automático, previamente treinado, que recebe um texto como entrada e retorna a probabilidade deste conter algum tipo de desinformação.

Ressaltamos novamente que, neste trabalho não estamos propondo nenhuma técnica de aprendizagem de máquina para a detecção automática de desinformação. Para esta finalidade pode-se utilizar abordagens já existentes, tais como: (??) (REIS *et al.*, 2020), (??), (CABRAL *et al.*, 2021), (MARTINS *et al.*, 2021a), (MARTINS *et al.*, 2021b), (REIS; BENEVENUTO, 2021), (??) ou ainda (COUTO *et al.*, 2022).

Vale destacar ainda que tanto o texto recebido pelo módulo *Classificador de Desinformação* quanto a probabilidade dele conter desinformação devem ser armazenados de forma persistente. Para esta finalidade, o modelo arquitetural IAra indica a utilização de um sistema de bancos de dados relacional (que pode ser o PostgreSQL, Oracle, SQLServer, MySQL, SQLite, por exemplo).

¹ <https://cloud.google.com/dialogflow>

² <https://www.twilio.com/pt-br/>

³ <https://www.ibm.com/br-pt/products/watson-assistant>

Caso o conteúdo fornecido pelo usuário seja uma imagem, esta é, inicialmente, enviada para o módulo *Reconhecimento Óptico de Caracteres*, o qual tem por finalidade reconhecer e extrair o texto presente no arquivo recebido. Este módulo analisa a imagem recebida como entrada e identifica os caracteres nela presentes, transformando esses dados em texto, o qual pode ser lido e pesquisado por softwares específicos (tais como Word, Google Docs, Notepad, OpenOffice, LibreOffice, dentre outros), ou ainda armazenado de forma persistente.

Atualmente, diversas APIs (*Application Program Interface*) disponibilizam serviços de Reconhecimento Óptico de Caracteres (ou *Optical Character Recognition* - OCR) de forma *online*, dentre elas destacam-se: Google Cloud Vision, Microsoft Computer Vision, Sema Media Data, Taggun, Cloudmersive e Tesseract. Logicamente, o módulo *Reconhecimento Óptico de Caracteres* pode ser implementado por meio da utilização de uma dessas APIs.

Neste cenário, o módulo *Reconhecimento Óptico de Caracteres* recebe uma imagem como entrada, envia este arquivo para um serviço de OCR disponibilizado de forma *online* e recebe o texto extraído. Em seguida, o texto obtido é enviado para o módulo *Classificador de Desinformação*, seguindo o fluxo anteriormente descrito.

Caso o conteúdo fornecido pelo usuário seja um áudio, este é enviado para o módulo *Reconhecimento Automático de Fala*, cuja finalidade consiste em transcrever o discurso contido no arquivo recebido como entrada em texto. Desta forma, este módulo busca reconhecer a fala humana presente em um arquivo de áudio e transcrevê-la em texto.

Atualmente, diversas APIs (*Application Program Interface*) podem ser utilizadas para acessar serviços de reconhecimento automático de fala (RAF) disponibilizados de forma *online*, dentre elas destacam-se: Google Speech-to-Text, IBM Watson Speech to Text, Amberscript, AssemblyAI, Rev.ai e Scriptix. Naturalmente, o módulo *Reconhecimento Automático de Fala* pode ser implementado por meio da utilização de uma dessas APIs.

Neste contexto, o módulo *Reconhecimento Automático de Fala* recebe um áudio como entrada, envia este arquivo para um serviço de RAF disponibilizado de forma *online* e recebe o texto extraído. Em seguida, o texto obtido é enviado para o módulo *Classificador de Desinformação*, seguindo o fluxo anteriormente descrito.

Por fim, caso o conteúdo a ser analisado seja um vídeo, este é enviado para o módulo *Conversor Vídeo-Áudio*, o qual tem por finalidade extrair o áudio contido no arquivo recebido como entrada. A conversão de um arquivo de mídia digital, como um arquivo de áudio ou vídeo, de um formato para outro, é denominada transcodificação. Isso costuma ser realizado por meio

da decodificação e recodificação do arquivo. Por exemplo, você pode converter um arquivo MP4 em MP3. Atualmente existem diversas bibliotecas e serviços *online* que realizam a conversão de arquivos de vídeo em áudio, como, por exemplo: a biblioteca Python *MoviePy Editor*, *Azure Media Service API*, *AWS Elemental MediaConvert*, *Telestream Cloud*, *Google Transcoder API*, dentre outras. Logicamente, o módulo *Conversor Vídeo-Áudio* pode ser implementado por meio da utilização de uma dessas ferramentas.

Neste cenário, o módulo *Conversor Vídeo-Áudio* recebe um vídeo como entrada, envia este arquivo para uma biblioteca ou para um serviço de transcodificação e recebe o áudio extraído. Em seguida, o áudio obtido é enviado para o módulo *Reconhecimento Automático de Fala*, o qual irá transcrever o áudio recebido em texto. Depois disso, o texto obtido é enviado para o módulo *Classificador de Desinformação*, seguindo o fluxo anteriormente descrito.

Se a ação de entrada do usuário tiver uma intenção relacionada à educação midiática, esta será enviada para o módulo *Guia de Identificação de Desinformação*. A educação midiática pode ser definida como o processo de desenvolvimento de competências que buscam formar pessoas com pensamento crítico e aptas a consumir, analisar e produzir conteúdos na era digital, contribuindo para que estas possam participar do ambiente informacional e midiático. Ela ganha importância em meio à preocupação com as manipulações políticas derivadas dos ambientes digitais. Desta forma, a educação midiática contribui para que as pessoas possam participar de maneira crítica do ambiente informacional e midiático em todos os seus formatos — dos impressos aos digitais.

Neste contexto, o módulo *Guia de Identificação de Desinformação* tem por finalidade guiar o usuário por meio de perguntas que o instiguem a refletir acerca das ações que deveriam ser realizadas durante o consumo de um conteúdo digital. O Modelo Arquitetural IAra propõe a utilização de sete perguntas básicas, cobrindo diferentes aspectos, conforme sugere o LAPIN – Laboratório de Políticas Públicas e Internet, em seu guia sobre como identificar *fake news* na Internet. O LAPIN é um centro independente de pesquisa e ação voltado para os desafios sociais, éticos, e jurídicos que as tecnologias digitais trazem a uma sociedade global conectada. Desde 2016, o Laboratório desenvolve pesquisas científicas, notas técnicas, cursos, campanhas, e ações direcionadas a temas como privacidade, proteção de dados pessoais, liberdade de expressão e inovação⁴.

A Tabela 2 ilustra cada uma dessas perguntas e os textos explicativos a elas asso-

⁴ <https://lapin.org.br/>

ciados. Vale destacar que as perguntas seguem uma ordem pré-determinada. O componente *Repositório de Dicas* consiste em um repositório utilizado pelo módulo *Guia de Identificação de Desinformação* para armazenar as perguntas e explicações que serão exibidas para o usuário. Desta forma, a inclusão, remoção e alteração de perguntas e textos explicativos podem ser realizadas de maneira prática.

Por fim, o módulo *Gerador de Respostas* tem por finalidade gerar as respostas que serão enviadas para o módulo *Interface de Utilização*. Este irá adaptar o resultado a ser exibido ao utente de acordo com o ambiente por ele utilizado.

Uma forma possível de representar um fluxo de diálogos completo consiste em utilizar uma máquina de estados. A Figura 7 ilustra o fluxo sugerido pelo Modelo Arquitetural IAra, o qual é formado pelos seguintes estados:

1. Boas-vindas: apresentação do *chatbot*;
2. Escolha do grupo de tarefas: verificação de conteúdo ou educação midiática;
3. Escolha do modo de verificação de conteúdo;
4. Verificação por texto;
5. Verificação por áudio;
6. Verificação por imagem;
7. Verificação por vídeo;
8. Educação midiática.

Número	Pergunta	Texto Associado
1	Você verificou fonte e autor?	Textos de <i>fake news</i> não costumam ter fontes e autores assinando o texto, muitas vezes são criadas em perfis de redes sociais que desejam espalhar desinformação. Ou são compartilhadas a partir de sites criados apenas com essa finalidade. Sem renome ou respaldo na mídia tradicional.
2	O título é muito chamativo?	Títulos chamativos que evoquem sentimento de revolta ou descontentamento fazem parte de uma tática dos criadores de <i>fake news</i> para chamar atenção e criar revolta no leitor. Não somente o título como o corpo do texto muitas vezes é repleto de palavras e frases que preveem grandes catástrofes ou conquistas, além de seu conteúdo incitar emoções fortes como raiva ou tristeza.
3	Você leu o texto completo?	É importante ler o texto e não apenas o título. Muitas vezes os autores de notícias falsas criam um texto com informações desconexas ou apenas cópias de notícias comuns confiando que a maioria dos leitores irão ler apenas o título.
4	Você verificou a data da notícia	Algumas vezes autores de <i>fake news</i> tentam enganar os leitores utilizando notícias reais porém em outro contexto temporal, utilizando notícias antigas.
5	Você verificou os nomes dos especialistas citados em algum buscador?	É comum em conteúdo desinformativo utilizarem nomes de supostos especialistas para tentar dar respaldo a notícia, porém muitas vezes esses especialistas não existem ou não possuem renome na comunidade científica.
6	Você notou o incentivo ao compartilhamento desta notícia?	Os autores de <i>fake news</i> desejam que seu conteúdo atinja o máximo de pessoas possíveis, então pedem aos leitores que compartilhem de forma veemente utilizando termos fortes.
7	Você verificou se a notícia saiu nas grandes mídias?	Notícias importantes saem em mais de um canal comunicativo da imprensa e grande mídia. Ao pesquisar em buscadores se a notícia não aparece em sites de renome é provável que exista uma desinformação.

Tabela 2 – Perguntas Utilizadas pelo Módulo *Guia de Identificação de Desinformação*

5 ESTUDO DE CASO

Com a finalidade de avaliar o Modelo Arquitetural IAra, proposto neste trabalho, realizamos a sua instanciação em dois ambientes distintos, WhatsApp e Telegram, dando origem aos *chatbots* IAraBot WhatsApp e IAraBot Telegram, respectivamente. Os *chatbots* foram desenvolvidos com base na metodologia *Lean Inception*, cujo objetivo é desenvolver versões incrementais de um Mínimo Produto Viável (*Most Value Product* - MVP, em inglês). Esta metodologia utiliza o conceito de *Lean Startup*, no qual o MVP é construído seguindo-se o ciclo construir - medir - aprender.

5.1 Instanciando o Modelo Arquitetural IAra

Os principais módulos que compõem o modelo arquitetural IAra podem ser instanciados de forma independente da plataforma de mensageria a ser utilizada. A Figura 8 ilustra o modelo arquitetural IAra destacando os módulos que foram instanciados de forma independente da plataforma de troca de mensagens utilizada.

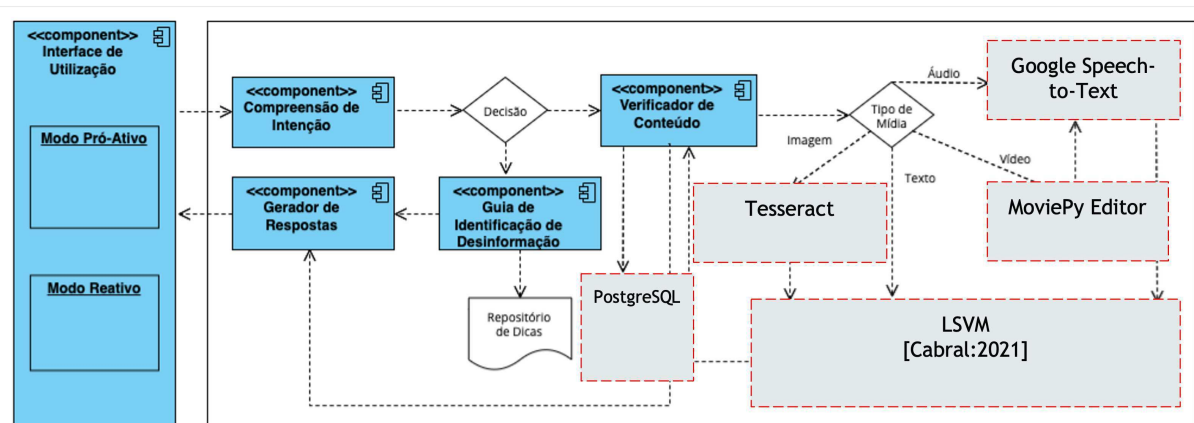


Figura 8 – Instanciação do Modelo Arquitetural IAra de Maneira Independente de Plataforma.

A Tabela 3 ilustra as tecnologias utilizadas no IAraBot. A escolha dessas tecnologias foi guiada pelos seguintes critérios: i) facilidade de uso, ii) disponibilidade *online* e iii) custo de utilização. Todos os códigos utilizados na criação dos *chatbots* IAraBot WhatsApp e IAraBot Telegram estão disponíveis no *Github* ¹.

¹ <https://github.com/jmmfilho/paper-iara>

Módulo	Tecnologia Utilizada
Reconhecimento Óptico de Caracteres	Tesseract
Reconhecimento Automático de Fala	Google Speech-to-Text
Conversor Vídeo-Áudio	MoviePy Editor
Classificador de Desinformação	LSVM
Banco de Dados Relacional	PostgreSQL

Tabela 3 – Tecnologias Utilizadas no *Chatbot* IAraBot.

O módulo “Classificador de Desinformação” foi implementado utilizando-se o melhor modelo obtido em (CABRAL *et al.*, 2021), mais precisamente uma LSVM (*Linear Kernel Support Vector Machine*), treinada com vetores TF-IDF, usando unigramas, bigramas e trigramas como *tokens*, após a remoção de *stopwords* e aplicação de lematização, o qual obteve para a métrica *F1-Score* um valor de 0,87.

Adicionalmente, um arquivo de configurações foi utilizado para habilitar/desabilitar o modo pró-ativo, bem como para definir o limiar a partir do qual o IAraBot iria considerar um conteúdo como desinformação. O limiar utilizado foi de 94%, ou seja, o IAraBot somente alerta, de forma autônoma, a possibilidade da presença de uma desinformação no grupo de bate papo se a probabilidade retornada pelo módulo *Classificador de Desinformação* for maior ou igual a 94%. Esse valor foi obtido de maneira empírica, por meio de experimentos que buscavam encontrar um limiar para o qual nenhum falso positivo fosse gerado. Para isso, inicialmente, avaliou-se o valor de 99% e, em seguida, fomos reduzindo esse valor em 1% a cada teste.

O módulo “Reconhecimento Óptico de Caracteres” foi implementado utilizando-se o Tesseract OCR, que é um software de reconhecimento ótico de caracteres de código aberto (Licença Apache 2.0) disponível no GitHub². O Tesseract originou-se de um projeto de pesquisa de doutorado na Hewlett-Packard (HP) em 1984, tendo sido disponibilizado em código aberto no final de 2005. O Tesseract trabalha em conjunto com a biblioteca Leptonica, a qual é responsável por realizar manipulações sobre as imagens, realizando transformações, rotações e outras operações. O Tesseract funciona nas plataformas Linux, Windows e MacOS.

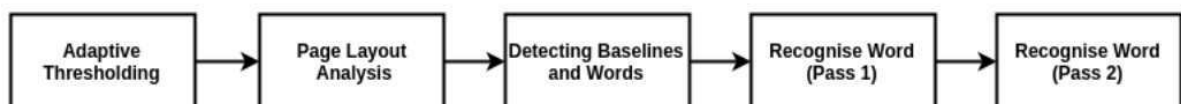


Figura 9 – Pipeline Utilizado pelo Tesseract OCR.

² <https://github.com/tesseract-ocr/tesseract>

O pipeline utilizado pelo Tesseract OCR é ilustrado na Figura 9. O primeiro passo é o *Adaptive Thresholding*, que converte a imagem em uma versão binária. A segunda etapa é a análise do *layout* da página, que é usada para extrair os blocos de texto dentro do documento. Na terceira etapa as linhas de base de cada linha são detectadas e o texto é dividido em palavras usando espaços definidos e espaços difusos. Na quarta etapa, os contornos dos caracteres são extraídos das palavras. O reconhecimento do texto é então iniciado como um processo de duas etapas. Na primeira passagem, o reconhecimento é feito usando um classificador estático. Cada palavra identificada satisfatoriamente é enviada para um classificador adaptativo. Uma segunda passagem é executada ao longo da página, usando o classificador adaptativo recém-aprendido em que palavras que não foram reconhecidas suficientemente bem são reconhecidas novamente.

O módulo “Reconhecimento Automático de Fala” foi implementado utilizando-se o *Google Speech-to-Text*, o qual foi publicamente lançado pela Google em 2008 na forma de um aplicativo de pesquisa por voz para iPhone. Utilizando as grandes quantidades de dados armazenados em seus servidores e avançados algoritmos de aprendizagem de máquina, o Google criou o primeiro sistema RAF em larga escala, a *API Cloud Speech-to-Text*, lançada em abril de 2017, incluindo o suporte para 90 idiomas. Atualmente, a *API Cloud Speech-to-Text* pode ser usada para reconhecer mais de 120 idiomas e dialetos. Esta API oferece recursos adicionais, como dicas de frases, *streaming*, detecção automática de idioma, filtragem de conteúdo inadequado, pontuação automática, reconhecimento multicanal, dentre outros. A API oferece suporte para as seguintes linguagens de programação: C#, Java, Python, Ruby, Go, Node.JS e PHP.

A *API Cloud Speech-to-Text* permite ao usuário escolher entre quatro modelos de reconhecimento de voz, cada um apropriado para um objetivo:

- *command_and_search*, para comandos curtos ou pesquisa por voz;
- *phone_call*, para chamadas telefônicas, o áudio neste modelo são gravados com uma taxa de amostragem de 8.000 Hz e está disponível apenas para o idioma inglês;
- *video*, indicado para vídeo e situações em que existe mais de um interlocutor, sugere-se que o áudio fornecido tenha uma taxa de amostragem de 16.000 Hz, está disponível apenas para o idioma inglês;
- *default*, o modelo padrão para áudios de longas durações e que não corresponde a nenhum modelo anterior.

O módulo “Conversor Vídeo-Áudio” foi implementado utilizando-se o MoviePy Editor, o qual é uma biblioteca Python para edição de vídeo que possibilita diversas operações básicas, tais como: corte, concatenações, inserções de títulos, composição de vídeo (também conhecida como edição não linear), processamento de vídeo e criação de efeitos personalizados. O MoviePy Editor pode ler e gravar todos os formatos de áudio e vídeo mais comuns, além disso funciona em Windows, MacOS e Linux. Ele é um software de código aberto originalmente escrito por Zulko e lançado sob licença do MIT, estando atualmente hospedado no GitHub ³.

5.2 Implementação do IAraBot Telegram

O IAraBot Telegram foi desenvolvido utilizando-se o *framework* DialogFlow e a *Bot API*, a qual permite criar programas que usam as mensagens do *Telegram*. A criação de *bots* por meio da *Bot API* é bastante simples, o que motivou a sua escolha. O Dialogflow é uma plataforma de processamento de linguagem natural que facilita o projeto e a integração de uma interface do usuário conversacional com aplicativos para dispositivos móveis, aplicativos Web, *chatbots*, sistemas interativos de resposta de voz e etc. Usando o Dialogflow, é possível fornecer maneiras novas e atraentes para os usuários interagirem com um produto de *software*. Esse *framework* permite analisar diferentes tipos de entrada de seus clientes, incluindo entradas de texto ou áudio. Ele também pode responder aos seus clientes de várias maneiras, seja por meio de texto ou com fala sintética.

Telegram Bots são contas especiais que não requerem um número de telefone para que sejam criadas. Essas contas servem como uma interface para um determinado código, o qual estará em execução em algum servidor. O *Telegram* oferece aos desenvolvedores dois tipos de API's diferentes: A *Bot API* permite criar programas que usam as mensagens do *Telegram*. Já a *Telegram API* (ou *Telegram Database Library - TDLib*) possibilita construir clientes *Telegram* personalizados. A criação de *bots* por meio da *Bot API* é bastante simples. Basta estar logado em um aplicativo *Telegram*, entrar em contato com *@BotFather* em seguida utilizar o comando */newbot*, escolher um nome para o *bot* e ele será criado automaticamente e de forma gratuita. *Tokens* de acesso necessários para o desenvolvimento do *bot* são liberados no momento da criação. Todas as consultas à *Bot API* devem ser atendidas por HTTPS, com suporte a métodos GET e POST.

Para conversar com o IAraBot Telegram basta enviar uma mensagem qualquer para

³ <https://zulko.github.io/moviepy/>

@Iara_UFC_bot ou utilizar o *QRCode* exibido na Figura 10. Já para avaliar o funcionamento do IAraBot Telegram no modo pró-ativo, o usuário pode entrar no grupo público denominado “Grupo IAraBot Telegram”. Isso pode ser realizado por meio de um *link* convite⁴ ou utilizando-se o *QRCode* exibido na Figura 11.

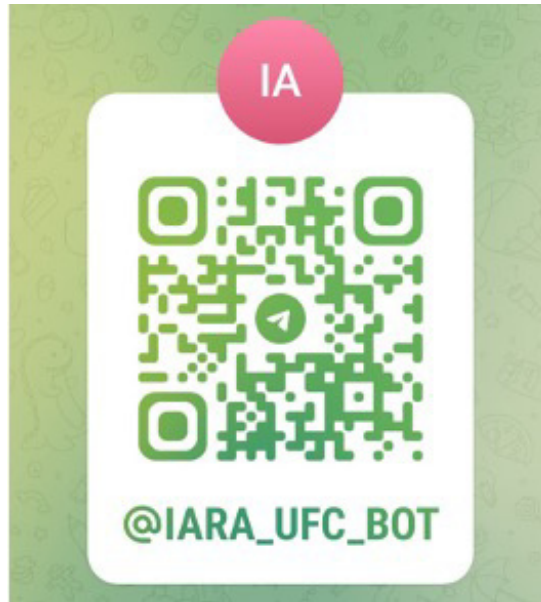


Figura 10 – QRCode para Conversar com o IAraBot Telegram.



Figura 11 – QRCode para Avaliar o IAraBot Telegram no Modo Pró-ativo.

⁴ <https://t.me/+HMTbqshm8VVmN2Rh>

5.3 Implementação do IAraBot WhatsApp

O IAraBot WhatsApp foi desenvolvido utilizando-se o *framework* Wago, que é um *fork* da biblioteca Go WhatsMeow⁵. Essa biblioteca implementa um cliente para interagir com a API Web multi-dispositivo do WhatsApp. Dentre os principais recursos disponibilizados pela biblioteca WhatsMeow, destacam-se:

⁵ <https://github.com/tulir/whatsmeow>

- Envio de mensagens para grupos de bate-papos públicos e privados (incluindo texto e mídia);
- Recebimento de mensagens (incluindo texto e mídia);
- Gerenciamento de grupos, incluindo o recebimento de informações acerca de diferentes eventos, tais como: mudança de nome do grupo, entrada e saída de membros, etc;
- Entrar em grupos por meio de *links* convite;
- Criar e enviar *links* convite;
- Envio e recebimento de recibos de entrega e leitura de mensagens;
- Envio e tratamento de recibos de nova tentativa se a descriptografia da mensagem falhar;
- Ler e escrever o estado do aplicativo (por exemplo, grupos silenciados, etc);

Para conversar com o IAraBot WhatsApp basta enviar uma mensagem qualquer para o número 55 85 98977-2830 ou utilizar o *QRCode* exibido na Figura 12. Já para avaliar o funcionamento do IAraBot WhatsApp no modo pró-ativo, o usuário pode entrar no grupo público denominado “Grupo IAraBot WhatsApp”. Isso pode ser realizado por meio de um *link* convite⁶ ou utilizando-se o *QRCode* exibido na Figura 13.



Figura 12 – QRCode para Conversar com o IAraBot WhatsApp.

Vale destacar ainda que os códigos dos *chatbots* IAraBot WhatsApp e IAraBot Telegram estão disponíveis no GitHub⁷.

⁶ <https://chat.whatsapp.com/Hw5VBHRm97tJx36Cb9s2qs>

⁷ <https://github.com/jmmfilho/paper-iara>



Figura 13 – QRCode para Avaliar o IARA Bot WhatsApp no Modo Pró-ativo.

5.4 Exemplos de Utilização do IARA Bot Telegram

Nesta seção, iremos ilustrar o funcionamento do *chatbot* IARA Bot Telegram. Para efeitos didáticos, será apresentado, inicialmente, a execução pró-ativa do IARA Bot Telegram e, em seguida, seu funcionamento reativo, destacando alguns fluxos de conversação e o conteúdo das mensagens geradas.

5.4.1 Funcionamento Pró-Ativo:

Para funcionar de forma pró-ativa, o IARA Bot Telegram necessita, inicialmente, ser adicionado a um grupo de bate papo. Em seguida, ele irá enviar uma mensagem de apresentação, conforme ilustra a Figura 14. Posteriormente, o IARA Bot Telegram irá monitorar e analisar automaticamente o conteúdo que trafega no grupo, incluindo texto, imagem, áudio e vídeo. Por fim, caso ele detecte que um determinado conteúdo tem uma probabilidade de conter desinformação acima do limiar especificado no arquivo de configurações, uma mensagem de alerta é enviada no grupo.

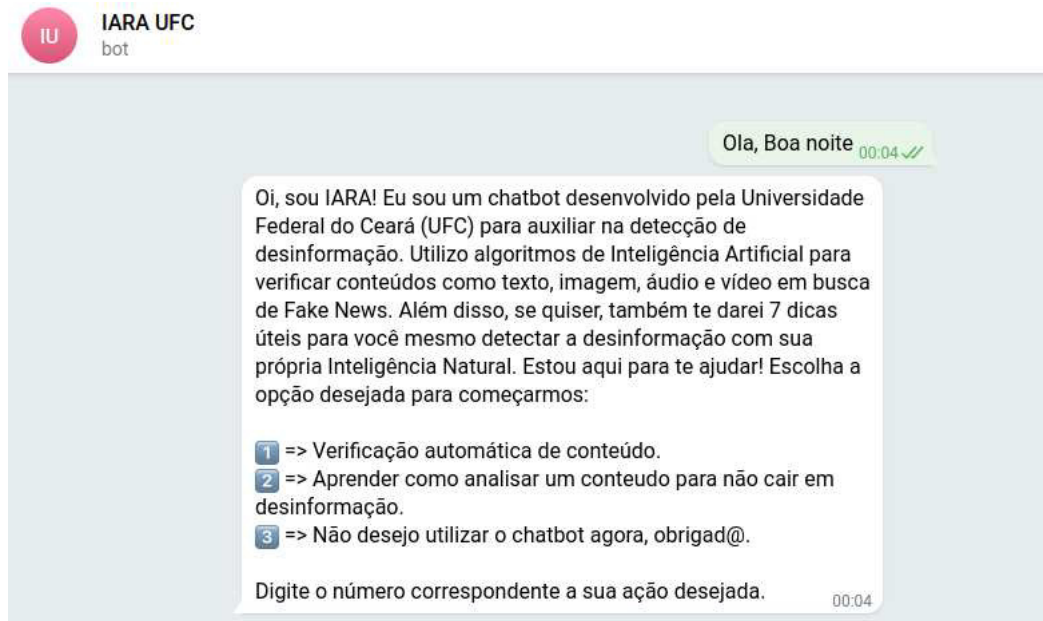


Figura 14 – Mensagem de Apresentação do IAraBot

Um exemplo do comportamento pró-ativo do IAraBot Telegram é ilustrado na Figura 15. Observe que uma mensagem de texto (contendo mais de 20 caracteres) foi postada por um usuário chamado “Macaio Cacabro” em um grupo de bate-papo. Essa mensagem foi capturada automaticamente pelo IAraBot Telegram e enviada para o módulo “Classificador de Desinformação”, o qual indicou que o texto possui 94% de probabilidade de conter desinformação. Como esse valor é maior ou igual ao limiar definido no arquivo de configurações, uma resposta foi automaticamente gerada e enviada ao grupo de bate-papo informando o risco da postagem conter uma desinformação. Vale destacar que um procedimento similar é adotado pelo IAraBot Telegram quando mídias digitais (imagem, áudio ou vídeo) são compartilhadas pelos usuários no grupo de bate-papo.

5.4.2 *Funcionamento Reativo:*

Para iniciar uma conversa o usuário deve enviar uma mensagem diretamente ao IAraBot Telegram e esperar a resposta. Em seguida, o IAraBot Telegram pergunta se o usuário deseja (1) realizar uma verificação automática de conteúdo, (2) aprender a analisar conteúdos ou (3) parar de usar o *chatbot*, conforme ilustrado na Figura 14.

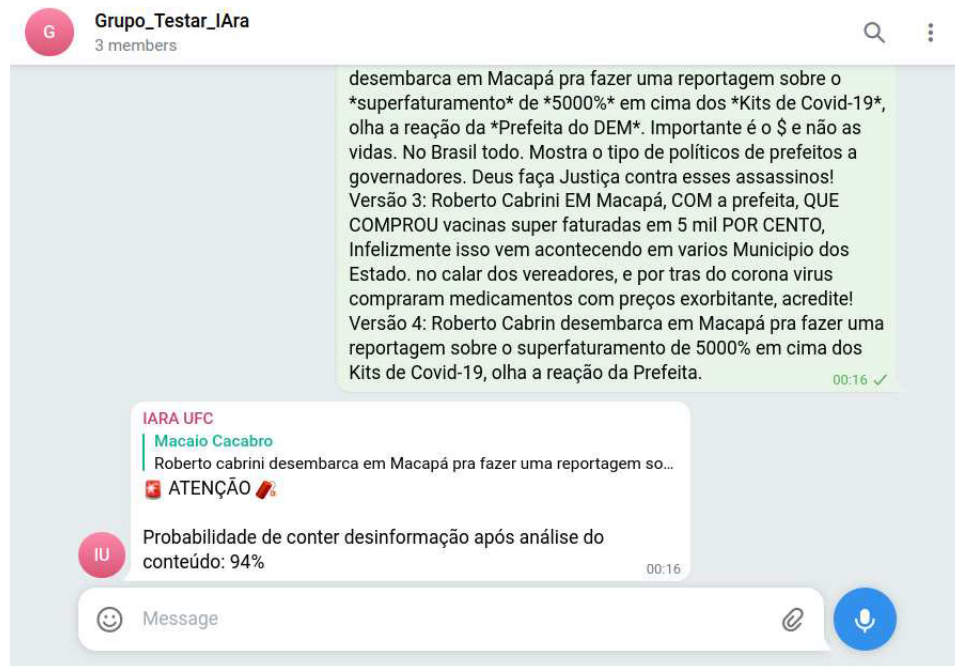


Figura 15 – Exemplo de Análise Pró-Ativa

Caso o usuário selecione a primeira opção, “Verificação automática de conteúdo”, o IAraBot Telegram apresenta uma mensagem explicando como prosseguir com a verificação e indicando os quatro tipos de conteúdos suportados: texto, imagem, áudio e vídeo. Nesta etapa, o usuário deve informar qual o tipo de conteúdo que deseja verificar, clicando em um dos *links* disponibilizados, conforme ilustra a Figura 16.

Se o usuário selecionar a opção “texto”, o IAraBot irá informar acerca de uma restrição que deve ser seguida: a mensagem de texto fornecida pelo usuário deve conter pelo menos 20 palavras. Essa limitação se deve à forma com que o classificador proposto em (CABRAL *et al.*, 2021) foi construído. Após o usuário enviar a mensagem, ela será analisada. A resposta que o *chatbot* retorna ao usuário contém a probabilidade do texto fornecido conter alguma desinformação, conforme ilustra a Figura 17.

Caso o usuário selecione a segunda opção, “Aprender como analisar um conteúdo”, o IAraBot Telegram inicia um tutorial sobre educação midiática composto por sete dicas acerca de como identificar desinformação. A Figura 18 ilustra a primeira dica do tutorial de educação midiática, enquanto a Figura 19 mostra um exemplo de desinformação que poderia ser identificada a partir desta dica.

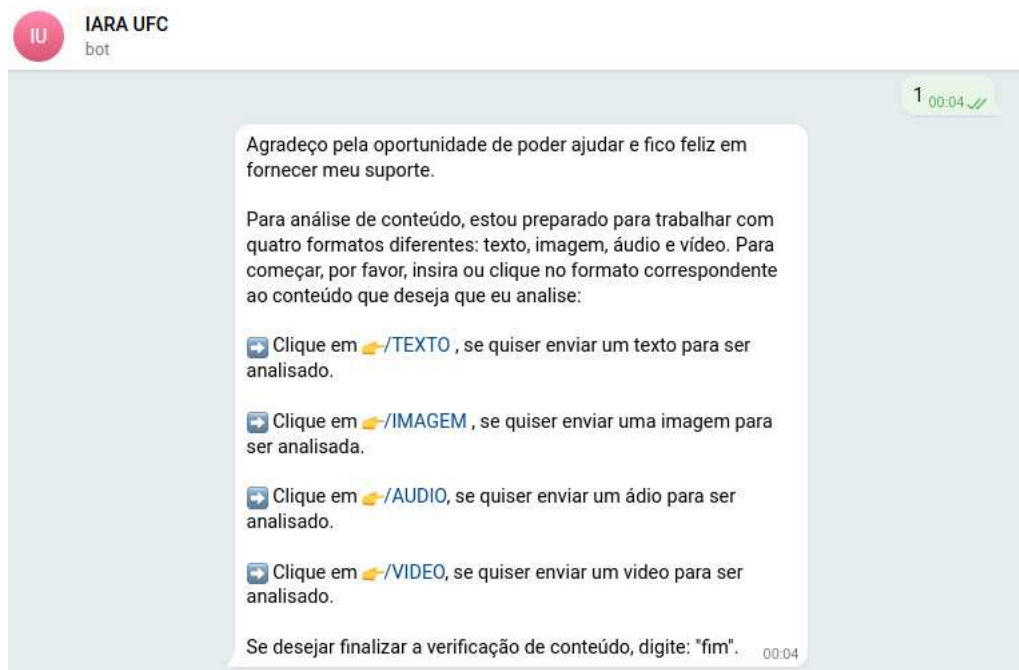


Figura 16 – Opções para Verificação de Desinformação

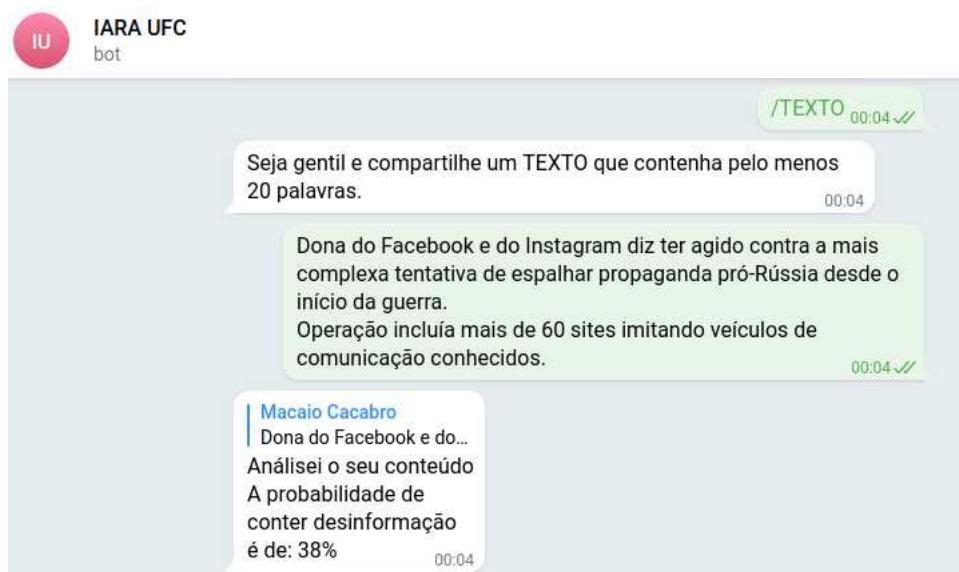


Figura 17 – Verificando Desinformação em Texto

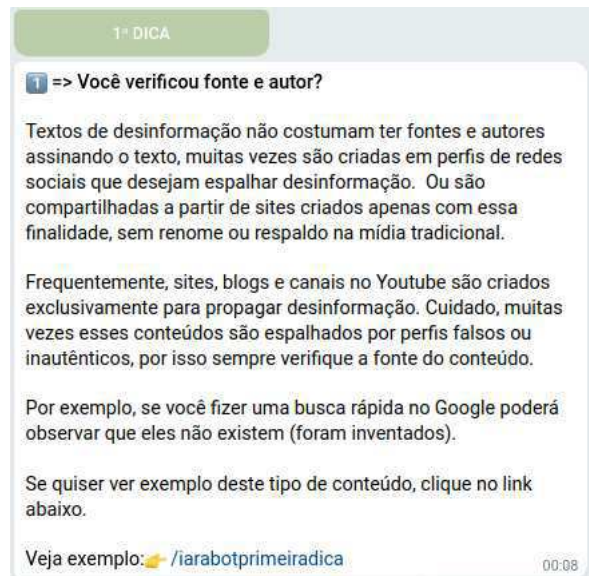


Figura 18 – Educação Midiática - Primeira Dica do Tutorial de Identificação de Desinformação



Figura 19 – Exemplo de Desinformação Referente à Primeira Dica

6 AVALIAÇÃO DE USABILIDADE

Neste capítulo vamos analisar as abordagens e métodos utilizados no teste de usabilidade, destacando as dificuldades, facilidades e erros identificados pelos usuários durante a avaliação da eficácia do *chatbot IAra por meio de dois diferentes testes: Net Promoter Score (NPS) e System Usability Scale (SUS)*. Posteriormente, apresentaremos de forma detalhada os resultados obtidos.

6.1 Teste de Usabilidade

De acordo com (MARTINS., 2005), a usabilidade representa uma das áreas mais significativas da ergonomia quando se trata de avaliar a interação entre o produto e o usuário. Esse conceito foi inicialmente desenvolvido para atender às exigências e demandas no campo da tecnologia da informação. Em (MARTINS *et al.*, 11/06/2013), demonstra-se que ao longo do tempo, o conceito de usabilidade foi constantemente revisado e aprimorado, evoluindo para se tornar mais abrangente e desafiador.

De acordo com (ARAÚJO *et al.*, 2017), a usabilidade pode ser dividida em cinco critérios básicos:

- **Intuitividade** – O sistema deve permitir a facilidade de uso de forma que, mesmo sem experiência, o usuário seja capaz de produzir algum trabalho satisfatoriamente;
- **Eficiência** – O sistema deve ser eficiente em seu desempenho apresentando um alto nível de produtividade;
- **Memorização** – Suas telas devem apresentar facilidade de memorização permitindo que usuários ocasionais consigam utilizá-lo mesmo depois de um longo intervalo de tempo;
- **Erro** – A quantidade de erros apresentados pelo sistema deve ser o mais reduzido possível, além de apresentarem soluções simples e rápidas mesmo para usuários iniciantes. Erros graves ou sem solução não podem ocorrer;
- **Satisfação** – O sistema deve agradar ao usuário, sejam eles iniciantes ou avançados, permitindo uma interação agradável.

Neste estudo, utilizamos o método de perspectiva, que envolveu usuários reais nas interações. Isso nos permitiu avaliar o desempenho por meio de questionários, entrevistas e avaliação da satisfação subjetiva, resultando em uma análise direcionada ao público-alvo.

6.1.1 Configuração da Avaliação de Usabilidade

A avaliação da usabilidade do *chatbot* IAra neste estudo foi conduzida especificamente com usuários do Telegram e do Whatsapp, independentemente de seus conhecimentos técnicos na área de inteligência artificial. A proposta consiste em, inicialmente, avaliar o *chatbot* com usuários do Telegram, sem exigir que tenham utilizado previamente um *chatbot* voltado para a detecção automática de desinformação ou possuam conhecimento prévio em sistemas de informação. Posteriormente, será realizada uma avaliação com usuários do Whatsapp, levando em consideração que estes tenham experiência prévia com pelo menos um *chatbot* especializado na detecção automática de desinformação. O IAra é um *chatbot* de monitoramento remoto. Os participantes receberam um link ou *QR code* para acessar o *chatbot* e foram convidados a explorar e testar todas as suas funcionalidades.

6.1.2 Aplicação de Questionários

Ao término da avaliação, os usuários responderam a um questionário que abordou aspectos tanto qualitativos quanto quantitativos. Além disso, empregamos o Questionário SUS (*System Usability Scale*) e *Net Promoter Scores* (NPS) para avaliar a facilidade de aprendizado, eficiência, facilidade de memorização, minimização de erros e satisfação dos participantes.

6.1.2.1 Net Promoter Score(NPS)

O *Net Promoter Score* (NPS) ainda é um método popular da medição de fidelidade do cliente, apesar de estudos recentes argumentarem que a fidelidade do cliente é multidimensional (ZAKI *et al.*, 2016). O NPS consiste em um questionário de duas partes. Em primeiro lugar, os entrevistados são apresentados a uma pergunta de classificação: “*Qual a probabilidade de você recomendar nosso produto/serviço a um amigo ou colega?*” em uma escala de **0** (pouco provável) a **10** (muito provável), que permite aos entrevistados de fornecer o principal razão para sua pontuação. (BNURS *et al.*, 2022). Com base nas respostas dadas pelos utentes, os participantes são divididos em três categorias:

1. **Promotores (9 - 10):** São clientes que deram notas altas e são considerados entusiastas do produto ou serviço. Eles estão mais propensos a recomendá-lo;
2. **Passivos (7 - 8):** São clientes que estão satisfeitos, mas não entusiásticos. Eles podem não recomendar o produto ou serviço ativamente;

3. **Detratores (0 - 6):** São clientes que deram notas baixas e estão insatisfeitos. Eles podem até mesmo desencorajar outras pessoas a usar o produto ou serviço.

6.1.2.2 Cálculo do NPS

Net Promoter Score é calculado subtraindo a porcentagem de Detratores da porcentagem de Promotores, portanto o NPS pode variar de **-100** (ou seja, todos os Detratores) a **+100** (ou seja, todos os Promotores) (BNURS *et al.*, 2022).

$$\text{NPS} = \text{Porcentagem de Promotores} - \text{Porcentagem de Detratores}$$

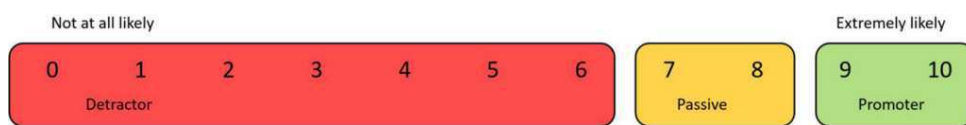


Figura 20 – Divisão dos Utentes em Três Categorias.

Fonte: (BNURS *et al.*, 2022)

6.1.2.3 System Usability Scale (SUS)

O SUS é uma ferramenta amplamente reconhecida para medir a usabilidade subjetiva, e sua aplicação na nossa avaliação é respaldada por (PADRINI-ANDRADEA *et al.*, 12/11/2018). Conforme (PADRINI-ANDRADEA *et al.*, 12/11/2018), o SUS é tecnologicamente agnóstico, o que significa que pode ser utilizado para avaliar diversos tipos de produtos e serviços, incluindo *websites*, *hardware*, sistemas multimodais, sistemas de comando por voz, aplicativos móveis e sistemas clínicos.

Segundo (BOUCINHA; TAROUÇO., 2013), a Escala SUS é um questionário composto por 10 itens, cada um oferecendo 5 opções de respostas. Essa escala foi desenvolvida por *John Brooke* em 1986, no laboratório da *Digital Equipment Corporation*, no Reino Unido. De acordo com (OLIVEIRA., 2021), sendo um questionário pós-teste, a recomendação é aplicá-lo imediatamente após o usuário concluir as tarefas no sistema em avaliação, antes de iniciar qualquer discussão. Além disso, é importante ressaltar que as respostas devem ser fornecidas de forma imediata, sem que o entrevistado tenha um longo período de reflexão. A Escala SUS consiste em 10 perguntas, como ilustrado na Figura 21, e os usuários respondem usando uma escala *Likert*, conforme descrito por (JÚNIOR; COSTA, 2014). Essa escala varia de 1 a 5, onde 1 representa “*Discordo completamente*” e 5 representa “*Concordo completamente*”.

System Usability Scale Questionnaire	Strongly Disagree					Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5	
2. I found the product unnecessarily complex.	1	2	3	4	5	
3. I thought the product was easy to use.	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5	
5. I found the various functions in the product were well integrated.	1	2	3	4	5	
6. I thought there was too much inconsistency in this product.	1	2	3	4	5	
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5	
8. I found the product very awkward to use.	1	2	3	4	5	
9. I felt very confident using the product.	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5	

Figura 21 – Questionário Padrão do SUS.

Fonte: (FILHO., 2022)

6.1.3 Cálculo do SUS

Após a conclusão do questionário, é essencial realizar uma quantificação para consolidar o resultado, que varia de 0 a 100. Essa quantificação segue a abordagem de (BARBOZA., Jun 18, 2019), que envolve a seguinte metodologia: para as respostas ímpares (1, 3, 5, 7, 9), subtrai-se 1 da pontuação atribuída pelo usuário a essa resposta; enquanto para as respostas pares (2, 4, 6, 8, 10), subtrai-se a pontuação atribuída de $5(5 - x)$. Posteriormente, somam-se todas as pontuações das dez perguntas e multiplica-se o resultado por 2.5. Esse processo está exemplificado na Figura 23, que detalha o cálculo da Escala SUS.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
P1	4 (4 -1= 3)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	3 (5-3 = 2)	2 (5-2 = 3)	5 (5-1 = 4)	2 (5-2 = 3)	5 (5-1 = 4)	1 (5-1 = 4)
P2	4 (4 -1= 3)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	3 (5-3 = 2)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)
P3	4 (4 -1= 3)	1 (5-1 = 4)	4 (4 -1= 3)	3 (5-3 = 2)	4 (4 -1= 3)	1 (5-1 = 4)	4 (4 -1= 3)	1 (5-1 = 4)	4 (4 -1= 3)	3 (5-3 = 2)
P4	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	4 (4 -1= 3)	4 (4 -1= 3)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)
P5	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)
P6	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	4 (4 -1= 3)	1 (5-1 = 4)	4 (4 -1= 3)	1 (5-1 = 4)
P7	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	5 (5-1 = 4)
P8	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	5 (5-1 = 4)
P9	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)	5 (5-1 = 4)	1 (5-1 = 4)
P10	2 (5-2 = 3)	3 (5-3 = 2)	5 (5-1 = 4)	1 (5-1 = 4)	4 (4 -1= 3)	4 (4 -1= 3)	5 (5-1 = 4)	4 (4 -1= 3)	5 (5-1 = 4)	1 (5-1 = 4)

Figura 22 – Cálculo da Escala SUS, Utentes do Telegram.
fonte: Próprio autor

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
P1	3 (5 -3) = 2	1 (5 -1) = 4	3 (5 -3) = 2	1 (5 -1) = 4	4 (4 -1) = 3	3 (5 -3) = 2	1 (5 -1) = 4	1 (5 -1) = 4	2 (5 -2) = 3	1 (5 -1) = 4
P2	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4	5 (5 -1) = 4
P3	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	2 (5 -2) = 3	4 (4 -1) = 3	2 (5 -2) = 3	4 (4 -1) = 3	2 (5 -2) = 3	4 (4 -1) = 3	1 (5 -1) = 4
P4	4 (4 -1) = 3	2 (5 -2) = 3	4 (4 -1) = 3	1 (5 -1) = 4	2 (5 -2) = 3	4 (4 -1) = 3	3 (5 -3) = 2	4 (4 -1) = 3	3 (5 -3) = 2	1 (5 -1) = 4
P5	4 (4 -1) = 3	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	4 (4 -1) = 3	5 (5 -1) = 4	1 (5 -1) = 4	4 (4 -1) = 3	4 (4 -1) = 3
P6	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	2 (5 -2) = 3	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4
P7	3 (5 -3) = 2	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	4 (4 -1) = 3	3 (5 -3) = 2	3 (5 -3) = 2	1 (5 -1) = 4	4 (4 -1) = 3	2 (5 -2) = 3
P8	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4
P9	4 (4 -1) = 3	1 (5 -1) = 4	4 (4 -1) = 3	3 (5 -3) = 2	3 (5 -3) = 2	3 (5 -3) = 2	4 (4 -1) = 3	1 (5 -1) = 4	4 (4 -1) = 3	3 (5 -3) = 2
P10	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4
P11	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4
P12	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	3 (5 -3) = 2	1 (5 -1) = 4	2 (5 -2) = 3	1 (5 -1) = 4
P13	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4	5 (5 -1) = 4	1 (5 -1) = 4
P14	4 (4 -1) = 3	1 (5 -1) = 4	5 (5 -1) = 4	3 (5 -3) = 2	5 (5 -1) = 4	3 (5 -3) = 2	5 (5 -1) = 4	1 (5 -1) = 4	3 (5 -3) = 2	3 (5 -3) = 2
P15	5 (5 -1) = 4	2 (5 -2) = 3	4 (4 -1) = 3	1 (5 -1) = 4	3 (5 -3) = 2	3 (5 -3) = 2	5 (5 -1) = 4	2 (5 -2) = 3	3 (5 -3) = 2	1 (5 -1) = 4

Figura 23 – Cálculo da Escala SUS, Utentes do Whatsapp.
fonte: Próprio autor

6.1.4 População

Ao todo, são 25 participantes na teste de usabilidade, pessoas de diversas áreas profissionais (professores universitários(as) administrador, assistentes sociais, psicologa, pedagoga, cientista social e biblioteconomia, educador popular, assessor de comunicação, vendedor/músico, dentista, sociólogo e estudantes) com nacionalidades diferentes (brasileiros, guineense, espanhol e boliviana) e possuem os seguintes dados demográficos: idades variando entre 22 e 64 anos.

Para avaliação de usabilidade, foi feita uma entrevista individual com cada participante no qual receberam convites para participar na avaliação e uma explicação sobre objetivo do nosso projeto de pesquisa, depois foram convidados por um encontro remoto, nessa ocasião foi explicada a importância de teste de usabilidade e o que eles devem levar em consideração, deixando bem claro que é preciso avaliar todas as funcionalidades oferecidas pelo *IARA*Bot. Em seguida, suas duvidas foram esclarecidas, e para dar prosseguimento a teste, o qrcode e *link* de acesso ao *IARA*Bot foi disponibilizado para os participantes. Destacamos que entre os participantes, 10 testaram a versão para Telegram e 15 participantes tentaram a versão para Whatsapp.

Após avaliação de usabilidade do *IARA*Bot, os participantes foram convidados a

responder um questionário qualitativo e quantitativo via *Google Plataform*. Por fim o questionário SUS e NPS foram aplicados, para avaliar sua facilidade de aprendizagem, eficiência, facilidade de memorização, minimização dos erros, e satisfação dos participantes.

7 RESULTADOS

Neste capítulo, discutiremos e apresentaremos os resultados obtidos na avaliação de usabilidade, tanto de maneira quantitativa quanto qualitativa. Isso inclui a aplicação do método SUS, NPS e a avaliação da satisfação dos usuários.

7.1 Resultados de Teste de Usabilidade NPS

Como mencionado no capítulo anterior, calculamos NPS baseando na sua pergunta principal cotada de 0 a 10. Como pode se notar na figura 24 representando pontuação obtida neste teste de usabilidade, dos quinze participantes, 12 atribuíram a nota entre 9 e 10 (classificados como promotores), dois atribuíram nota 7 e 8 (classificados como passivos ou neutros) e um atribuiu nota 1 (classificado como detrator), veja a figura 24.

Participante	Pontuação	Classificação
P1	5	Detrator
P2	10	Promotor
P3	9	Promotor
P4	7	Passivo
P5	8	Passivo
P6	10	Promotor
P7	10	Promotor
P8	10	Promotor
P9	10	Promotor
P10	10	Promotor
P11	10	Promotor
P12	10	Promotor
P13	10	Promotor
P14	10	Promotor
P15	9	Promotor

Figura 24 – Categorização da resposta NPS.
fonte: Próprio autor

O NPS é dada pela subtração da porcentagem dos Promotores menos a porcentagem dos Detratores. Valores de NPS acima de 70 são considerados "excelentes". Já os valores entre 50 e 70 são considerados "muito bons". Por fim, valores entre 0 e 50 são considerados "bons". O teste do NPS feito nesta pesquisa, como bem detalhado na figura 25, teve um resultado com o questionário NPS, o valor de 73,33%. Esse resultado é classificado no nível excelente.

Detratores	1	6,67%
Neutros	2	13,33%
Promotores	12	80,00%
Total	15	100,00%

NPS	73,33%
------------	---------------

Figura 25 – Resultado NPS.
fonte: Próprio autor

7.2 Resultados de Teste de Usabilidade SUS

Após a coleta dos questionários preenchidos, procedemos à conversão das pontuações brutas dos 10 itens em pontuações ajustadas. Esse processo é realizado da seguinte forma: para as respostas ímpares, subtrai-se 1 da pontuação atribuída pelo usuário; para as respostas pares, subtrai-se a pontuação atribuída de $5(5 - x)$. Em seguida, somamos todas as pontuações das dez perguntas e multiplicamos o resultado por 2.5.

As Figuras 26 e 27 apresentam os resultados obtidos com o questionário SUS para versão IAraBot Telegram e resultou em uma média de 93.25 e a versão IAraBot WhatsApp resultou em uma média de 88.17. Estes valores, de acordo com (BANGOR *et al.*, 2009), são considerados a cima da expetativa pois o resultado sus pode ser classificados de seguinte forma:

- Valores menores que 25 são considerados “*Muito ruins*”;
- Valores entre 25 e 38 são considerados “*Pobres*”;
- Valores entre 38 e 52 são considerados “*OK*”;
- Valores entre 52 e 72 são considerados “*Bons*”;
- Valores entre 72 e 85 são considerados “*Excelentes*”;
- Valores acima de 85 são considerados “*Acima das expectativas*”;

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Raw Score	SUS Score Final
P1	4	1	5	1	3	2	5	2	5	1	35	87,5
P2	4	1	5	1	5	3	5	1	5	1	37	92,5
P3	4	1	4	3	4	1	4	1	4	3	31	77,5
P4	5	1	5	1	4	4	5	1	5	1	38	95
P5	5	1	5	1	5	1	5	1	5	1	40	100
P6	5	1	5	1	5	1	4	1	4	1	38	95
P7	5	1	5	1	5	1	5	1	5	5	40	100
P8	5	1	5	1	5	1	5	1	5	5	40	100
P9	5	1	5	1	5	1	5	1	5	1	40	100
P10	2	3	5	1	4	4	5	4	2	1	34	85
Média												93,25

Figura 26 – Resultado do Cálculo Aplicado aos Usuários do Telegram.
fonte: Próprio autor

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Raw Score	SUS Score Final
P1	2	4	2	4	3	2	4	4	3	4	32	80
P2	4	4	4	4	4	4	4	4	4	4	40	100
P3	4	4	4	3	3	3	3	3	3	4	34	85
P4	3	3	3	4	3	3	2	3	2	4	30	75
P5	3	4	4	4	4	3	4	4	3	3	36	90
P6	4	4	4	3	4	4	4	4	4	4	39	97,5
P7	2	4	4	4	3	2	2	4	3	3	31	77,5
P8	4	4	4	4	4	4	4	4	4	4	40	100
P9	3	4	3	2	2	2	3	4	3	2	28	70
P10	4	4	4	4	4	4	4	4	4	4	40	100
P11	4	4	4	4	4	4	4	4	4	4	40	100
P12	4	4	4	4	4	4	2	4	3	4	37	92,5
P13	4	4	4	4	4	4	4	4	4	4	40	100
P14	3	4	4	2	4	2	4	4	2	2	31	77,5
P15	4	3	3	4	2	2	4	3	2	4	31	77,5
Média												88,17

Figura 27 – Resultado do Cálculo Aplicado aos Usuários do Whatsapp.
fonte: Próprio autor

7.3 Avaliações das Funcionalidades

Nesta seção será apresentado os resultados estatístico sobres as funcionalidades testadas pelos usuários. Como mencionados nos capítulos anteriores, os participantes deram notas de avaliação de 1 a 5 em função de sua satisfação com o funcionamento de *chatbot* considerando:

- 1 - Muito insatisfatória;
- 2 - Insatisfatória;
- 3 - Indiferente;
- 4 - Satisfatória;

– 5 - Muito satisfatória.

Os usuários avaliaram a detecção automática de desinformação por texto, imagem, áudio e vídeo e um mini-curso que foi feito por eles com 7 dicas valiosas para identificação de desinformação. A seguir vamos detalhar avaliação a partir de cada item avaliado.

As Figuras 28, 29 exibem os resultados da avaliação realizada pelos usuários em relação à detecção de desinformação por meio do recurso de análise de texto. Nessa funcionalidade, o usuário escolhe enviar um texto ao *bot* para que seja analisada a probabilidade de conter desinformação.

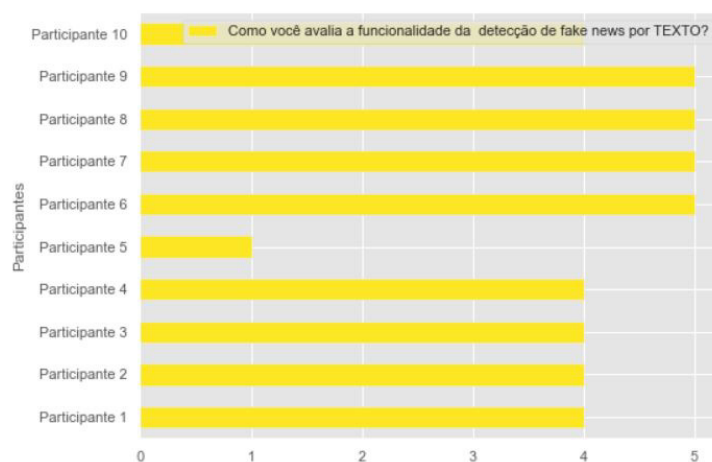


Figura 28 – Detecção de *Fake News* por Texto.

Fonte: Próprio autor. (Versão Telegram)



Figura 29 – Detecção de *Fake News* por Texto.

Fonte: Próprio autor. (Versão Whatsapp)

As Figuras 30, 31 exibem os resultados da avaliação realizada pelos usuários em relação à detecção de desinformação por meio do recurso de análise de imagem. Nessa funciona-

lidade, o usuário escolhe enviar uma imagem ao *bot* para que seja analisada a probabilidade de conter desinformação.

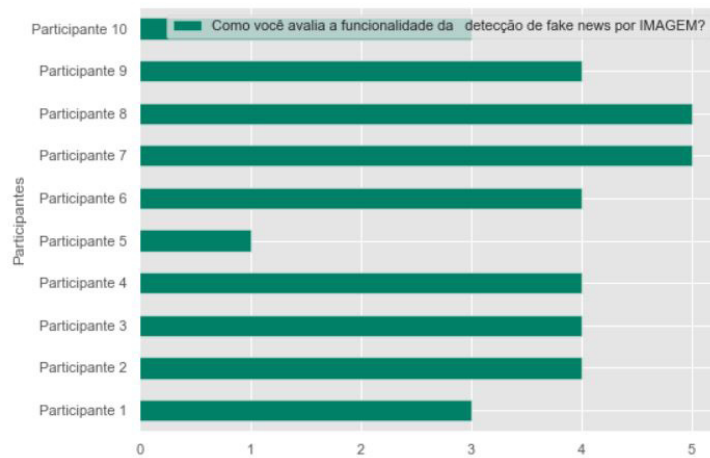


Figura 30 – Detecção de *Fake News* por Imagem.

Fonte: Próprio autor. (Versão Telegram)



Figura 31 – Detecção de *Fake News* por Imagem.

Fonte: Próprio autor. (Versão Whatsapp)

As Figuras 32, 33 exibe os resultados da avaliação realizada pelos usuários em relação à detecção de desinformação por meio do recurso de análise de áudio. Nessa funcionalidade, o usuário escolhe enviar um áudio ao *bot* para que seja analisada a probabilidade de conter desinformação.

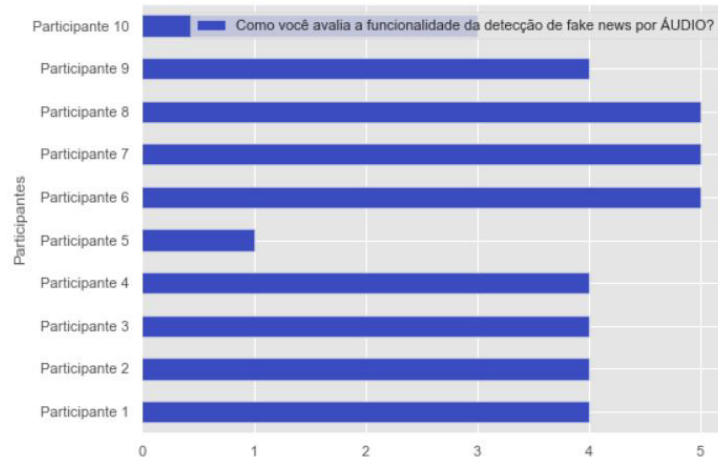


Figura 32 – Detecção de *Fake News* por Áudio.

Fonte: Próprio autor. (Versão Telegram)

Como você avalia a funcionalidade da detecção de fake news por AUDIO

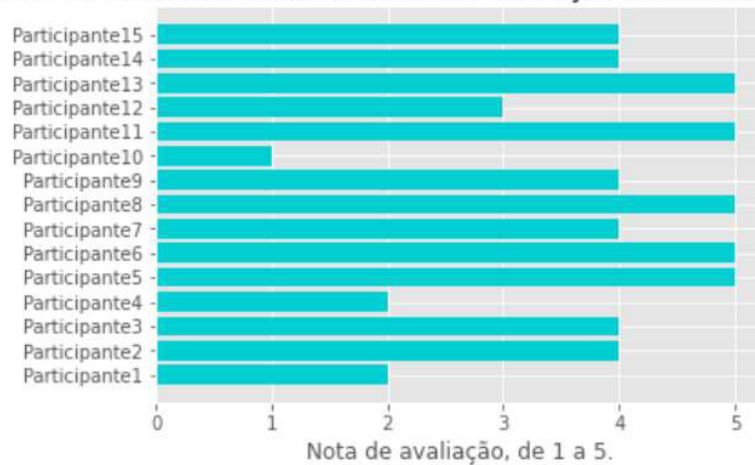


Figura 33 – Detecção de *Fake News* por Áudio.

Fonte: Próprio autor. (Versão Whatsapp)

As Figuras 34 exibe os resultados da avaliação realizada pelos usuários em relação à detecção de desinformação por meio do recurso de análise de vídeo. Nessa funcionalidade, o usuário escolhe enviar um vídeo ao *bot* para que seja analisada a probabilidade de conter desinformação.



Figura 34 – Detecção de *Fake News* por Vídeo.
 fonte: Próprio autor. (Versão Telegram)

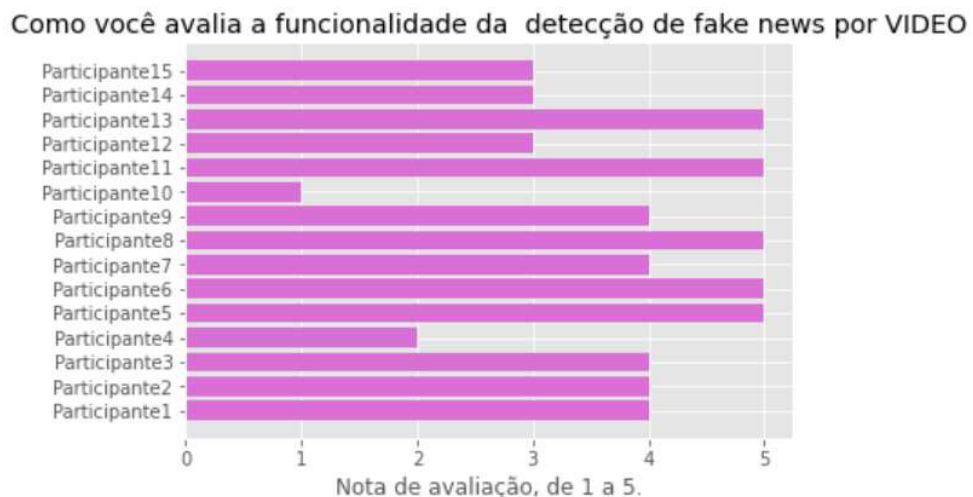


Figura 35 – Detecção de *Fake News* por Vídeo.
 fonte: Próprio autor. (Versão Whatsapp)

7.3.1 Avaliação sobre o mini-curso com as 7 DICAS

Foi avaliado também a função Educação Midiática em formato de um mini-curso com as 7 DICAS para combater a desinformação, pontuando a nota de 1 muito insatisfeito e a nota 5 muito satisfeito. A Figura 36 exibe a relação de usuários com suas notas como podemos constatar, 100% dos participantes atribuíram notas 5.

O que você acha do mini-curso com as 7 DICAS?

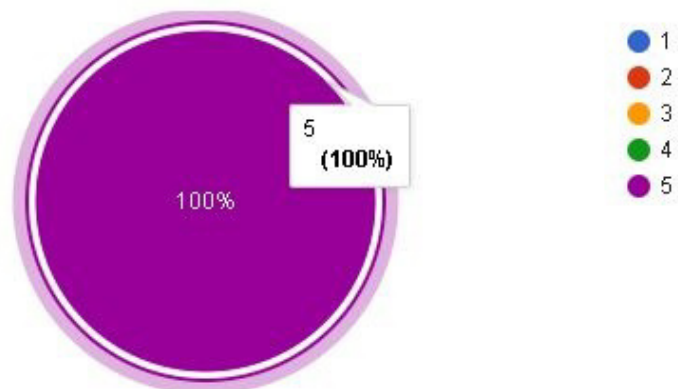


Figura 36 – Educação midiática no formato de mini-curso.
Fonte: Próprio autor

E por fim, foi perguntado aos usuários se “*eles recomendaria chatbot IAra para alguém usar?*”, 100% dos participantes responderam que sim, na Figura 37 temos o resultado dessa pergunta, e em seguida pontuaram uma nota de 1 a 10, para indicar o quanto recomendaria chatbot IAra, onde a nota 1, nem um pouco provável e a nota 10 extremamente provável. Veja figura 38.

Você recomendaria chatbot IAra para alguém?

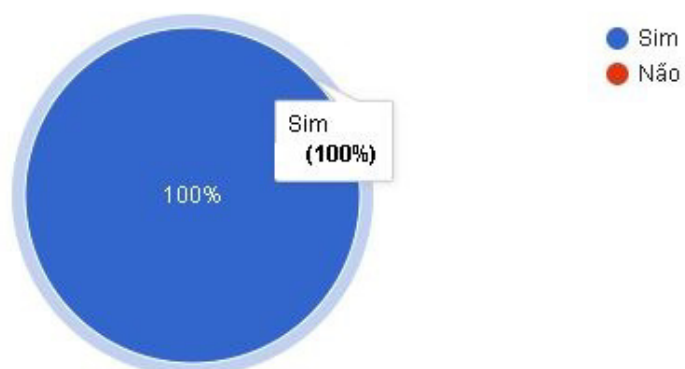


Figura 37 – Aprovaçã do Chatbot IAra.
fonte: Próprio autor



Figura 38 – Recomendar IAra para Alguém.
fonte: Próprio autor

Todos os tipos de conteúdo mencionados, como texto, imagem, áudio e vídeo, que são enviados para serem analisados pelo *chatbot*, passam por um processo de extração e cálculo em porcentagem para determinar a veracidade do conteúdo. Em seguida, o *chatbot* envia uma resposta ao usuário. Além disso, os dados resultantes desse processo são armazenados em um banco de dados, que serve para alimentar a base de conhecimento. Este banco de dados pode ser disponibilizado para outros pesquisadores utilizarem. O banco de dados utilizado para armazenar esses conteúdos é o PostgreSQL. Na Figura 39, temos alguns exemplos dos dados coletados durante o diálogo.

message_chat_id	message_chat_type	message_date	message_text	number_of_words	number_of_characteres	probability
5988496557	private	1670882904	Boa tarde irmã Idalina! Sou o Pr. Mardes Silva...	101.0	541.0	94%
5805765242	private	1670977786	Roberto cabrini desembarca em Macapá pra fazer...	205.0	1029.0	94%
392950427	private	1670881264	O bilionário Elon Musk (foto) foi vaiado ao su...	56.0	269.0	9%
5805765242	private	1670977804	Urander Fernandes de Oliveira, CEO da holding ...	458.0	2520.0	86%
5938639520	private	1670881449	Ponta de Langa @pontalancahdl IAaki Willia...	154.0	680.0	8%
...
5988496557	private	1670882248	Y, Caravana do DIREITOS HUMANOS (band VIOLAGA...	91.0	407.0	1%

Figura 39 – Dados Coletados Durante Dialogo do IAraBot Telegram com os Usuários
Fonte: Próprio autor

8 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho tem como principal contribuição um modelo arquitetural, chamado IAra, específico para auxiliar o desenvolvimento de *chatbots* especializados na detecção automática de desinformação e na educação midiática. Outra contribuição relevante consistiu no desenvolvimento de um *chatbot* pró-ativo chamado IAraBot, para duas plataformas diferentes, WhatsApp e Telegram, instanciando assim o modelo arquitetural IAra. Vale destacar que os trabalhos encontrados na literatura adotam uma abordagem reativa e, em geral, foram desenvolvidos para uma plataforma específica. O IAraBot, em seu modo pró-ativo, monitora, detecta e alerta, de forma automática e em tempo real, a presença de conteúdos contendo desinformação. Adicionalmente, o IAraBot fornece orientações e exemplos práticos para que os próprios usuários desenvolvam as habilidades necessárias para identificar conteúdos que contenham desinformação. A fim de avaliar a usabilidade do IAraBot Telegram e a satisfação do usuário, dois diferentes testes foram utilizados: *System Usability Scale* (SUS) e *Net Promoter Score* (NPS), respectivamente.

Para funcionar em modo pró-ativo, o IAraBot Telegram necessita executar uma espécie de “escuta ativa”, comumente utilizada pelos assistentes virtuais, tais como Alexa, Siri ou Cortana. Logicamente, essa funcionalidade levanta questionamentos éticos, bem como pode limitar a adoção da ferramenta.

No módulo “Classificador de Desinformação”, o modelo arquitetural IAra sugere a utilização do melhor modelo obtido em (CABRAL *et al.*, 2021), o qual apresentou para a métrica *F1-Score* um valor de 0,87. Este modelo foi concebido para textos que contenham pelo menos 20 palavras. Contudo, sabe-se que boa parte dos textos compartilhados em aplicativos de mensagens instantâneas são curtos. Por outro lado, textos contendo desinformação tendem a ser mais longos, uma vez que buscam simular a linguagem jornalística, além de criar pânico por meio de teorias da conspiração, por exemplo. De qualquer forma, é possível substituir o classificador utilizado de maneira bastante simples, uma vez que este é atualmente disponibilizado por meio de um serviço *web*.

Por fim, vale destacar que o modelo arquitetural IAra foi publicado como artigo completo no 29^o. Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2023).

Como trabalhos futuros pretendemos:

- Investigar como avaliar a presença de desinformação a partir de *links*, os quais podem referenciar páginas *Web* ou repositórios de conteúdo, como, por exemplo, o YouTube ou o Facebook;
- Avaliar a percepção dos usuários acerca do funcionamento pró-ativo do IAraBot;
- Avaliar a utilização de classificadores de desinformação voltados para textos curtos;
- Instanciar o IAraBot para outras plataformas, tais como o Reddit e o Discord.

REFERÊNCIAS

- ADJIN-TETTEY, T. D. Combating fake news, disinformation, and misinformation: Experimental evidence for media literacy education. **Cogent Arts & Humanities**, Cogent OA, v. 9, n. 1, p. 2037229, 2022. Disponível em: <https://doi.org/10.1080/23311983.2022.2037229>. Acesso em: 16 de Janeiro de 2023.
- AGUIAR DE LIMA, T.; DA COSTA-ABREU, M. **A survey on automatic speech recognition systems for Portuguese language and its variations**. 2020. 101055 p. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0885230819302992>. Acesso em: 04 de Abril de 2022.
- ALMEIDA, A.; ARAÚJO, F. O. **AnneBot: Um Chatbot para Auxiliar no Processo de Ensino e Aprendizagem do Pensamento Computacional**. Porto Alegre, RS, Brasil: SBC, 2021. 12–13 p. Disponível em: https://sol.sbc.org.br/index.php/educomp_estendido/article/view/14851. Acesso em: 14 de junho de 2023.
- ALMEIDA, G. L. M.; LIMA, M. d. O.; OLIVEIRA, A. S. S. d.; CHAGAS, A. M. **A EDUCAÇÃO MIDIÁTICA E O COMBATE AS FAKE NEWS**. 2022. 1470–1480 p. Disponível em: <https://periodicorease.pro.br/rease/article/view/5564>. Acesso em: 08 de março de 2023.
- ARAÚJO, J.; PEREIRA, M.; SOUSA, V. G.; CAMPOS., L. F. O design centrado no usuário e a usabilidade: Uma proposta de aplicativo para celular. 16º Ergodesign – Congresso Internacional de Ergonomia e Usabilidade de Interfaces Humano Tecnológica, 2017.
- ATHREYA, R. G.; NGOMO, A.-C. N.; USBECK, R. **Enhancing Community Interactions with Data-Driven Chatbots–The DBpedia Chatbot**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018. 143–146 p. Disponível em: <https://doi.org/10.1145/3184558.3186964>. Acesso em: 21 de setembro de 2022.
- BANGOR, A.; KORTUM, P.; MILLER., J. Determining what individual sus scores mean: Adding an adjective rating scale. *journal of usability studies* - Vol. 4, Issue 3, May 2009, pp. 114-123, 2009.
- BARBOZA., A. **Medindo a usabilidade do seu produto com System Usability Scale (SUS)**. Jun 18, 2019. Disponível em: <https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>. Acesso em: 14 de dezembro de 2022.
- BNURS, C. A.; BPHARM, R. W.; BSW(HONS, A. M. S. A.; HARRISON., R. The ultimate question? evaluating the use of net promoterscore in healthcare: A systematic review. *health expectations - Health Expectations an international journal of public Participation in Health Care and Health Polivy* ., 2022.
- BOUCINHA, R. M.; TAROUÇO., L. M. R. Avaliação de ambiente virtual de aprendizagem com o uso do sus - system usability scale. CINTED-UFRGS - Novas Tecnologias na Educação, 2013.
- CABRAL, L.; MONTEIRO, J. M. S.; SILVA, J. W. F. da; MATTOS, C. L. C.; MOURÃO, P. J. C. *Fakewhatsapp.br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages*. In: **International Conference on Enterprise Information Systems**. [S. l.: s. n.], 2021.

COUTO, J. M. M.; SALLES, I.; PIMENTA, B. C.; ASSIS, S.; SILVA, L. A.; REIS, J. C. S.; BENEVENUTO, F. **Detecção Automática de Desinformação Relacionada à Covid-19 no Brasil**. [S. l.]: SBC, 2022. 164–176 p. Disponível em: <https://doi.org/10.5753/sbbd.2022.224326>. Acesso em: 07 de janeiro de 2023.

CRUZ, J.; AVELAR, M.; VIANA, M.; ALBERGARIA, E.; CARVALHO, D.; ROCHA, L. **Creating an Academic Conversational Agent for Dynamic Information Retrieval**. Porto Alegre, RS, Brasil: SBC, 2020. Disponível em: <https://sol.sbc.org.br/index.php/sbsi/article/view/13781>. Acesso em: 23 de Novembro de 2021.

CUNHA, L. C. C. da; MONTEIRO, J.; SILVA, W. da; MATTOS, C.; MOURAO., P. **Fakewhastapp.br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages**. Science and Technology Publications, Lda., 2021.

DÖRR., J. B. Um estudo sobre técnicas utilizadas para o reconhecimento de sons com o uso de inteligência artificial e python. 19º Congresso Latino-americano de Software Livre e Tecnologias Abertas, p. 6 – 10, 2022.

FERREIRA, M. V. G.; SOUZA., J. F. de. Use of automatic speech recognition systems for multimedia applications. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Minicursos, 2008.

FILHO., J. A. C. Um guia prático para apoiar tarefas preditivas em ciência de dados. Aprovado pela Banca Examinador de Universidade Federal do Ceará (UFC), 2022.

GUO, B.; DING, Y.; YAO, L.; LIANG, Y.; YU, Z. **The Future of Misinformation Detection: New Perspectives and Trends**. 2019.

GUPTA, A.; KUMAR, N.; PRABHAT, P.; GUPTA, R.; TANWAR, S.; SHARMA, G.; BOKORO, P. N.; SHARMA, R. Combating fake news: Stakeholder interventions and potential solutions. **IEEE Access**, v. 10, p. 78268–78289, 2022.

GUPTA, A.; VARUN, Y.; DAS, P.; MUTTINENI, N.; SRIVASTAVA, P.; ZAFAR, H.; CHAKRABORTY, T.; NATH., S. **TruthBot: An Automated Conversational Tool for Intent Learning, Curated Information Presenting, and Fake News Alerting**. 2021. Disponível em: <https://arxiv.org/abs/2102.00509>. Acesso em: 30 de Agosto de 2022.

INTERVOZES. **Desinformação: ameaça ao direito à comunicação muito além das fake news. Disponível**. 2019. Disponível em: <http://intervozes.org.br/publicacoes/desinformacao-ameaca-ao-direito-a-comunicacao-muito-alem-das-fake-news/>. Acesso em: 29 de junho de 2022.

JUNIOR, M.; MELO, P. F.; SILVA, A. P. C. da; BENEVENUTO, F.; ALMEIDA, J. M. **Towards Understanding the Use of Telegram by Political Groups in Brazil**. ACM, 2021. 237–244 p. Disponível em: <https://doi.org/10.1145/3470482.3479640>. Acesso em: 09 de Fevereiro de 2022.

JÚNIOR, S. D. d. S.; COSTA, F. J. Mensuração e escalas de verificação: uma análise comparativa das escalas de likert e phrase completion. **PMKT–Revista Brasileira de Pesquisas de Marketing, Opinião e Mídia**, v. 15, n. 1-16, p. 61, 2014.

KATTENBECK, M. K. M.; FERSTL, M.; ALT, F.; LUDWIG., B. Airbot: Using a work flow model for proactive assistance in public spaces. In: **MOBILEHCI'18 (Ed.)**. [S. l.]: Proceedings

of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, 2016. p. 213–220.

KHAN, N. A.; ALBATEIN., J. Covibot- an intelligent whatsapp based advising bot for covid-19. 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) March 17–18, 2021, Amity University Dubai, UAE, 2021.

LAZER, D. M. J.; BAUM, M. A.; BENKLER, Y.; BERINSKY, A. J.; GREENHILL, K. M.; MENCZER, F.; METZGER, M. J.; NYHAN, B.; PENNYCOOK, G.; ROTHCHILD, D.; SCHUDSON, M.; SLOMAN, S. A.; SUNSTEIN, C. R.; THORSON, E. A.; WATTS, D. J.; ZITTRAIN, J. L. The science of fake news. **Science**, American Association for the Advancement of Science, v. 359, n. 6380, p. 1094–1096, 2018. ISSN 0036-8075. Disponível em: <https://science.sciencemag.org/content/359/6380/1094>. Acesso em: 11 de Julho de 2022.

LAZER, D. M. J.; BAUM, M. A.; BENKLER, Y.; BERINSKY, A. J.; GREENHILL, K. M.; MENCZER, F.; METZGER, M. J.; NYHAN, B.; PENNYCOOK, G.; ROTHCHILD, D.; SCHUDSON, M.; SLOMAN, S. A.; SUNSTEIN, C. R.; THORSON, E. A.; WATTS, D. J.; ZITTRAIN, J. L. The science of fake news. **Science**, v. 359, n. 6380, p. 1094–1096, 2018. Disponível em: <https://www.science.org/doi/abs/10.1126/science.aao2998>. Acesso em: 11 de Julho de 2022.

LOPEZ, C.; RICHARDSON, A.; MARSH, K.; ALVAREZ, L. Antie: The activism chatbot. In: **Proceedings of the 52nd ACM Technical Symposium on Computer Science Education**. New York, NY, USA: Association for Computing Machinery, 2021. (SIGCSE '21), p. 1383. ISBN 9781450380621. Disponível em: <https://doi.org/10.1145/3408877.3439693>. Acesso em: 20 de Abril de 2023.

MAALEJ, Z. **Discourse Studies**, Sage Publications, Ltd., v. 3, n. 3, p. 376–378, 2001. ISSN 14614456, 14617080. Disponível em: <http://www.jstor.org/stable/24047513>. Acesso em: 22 de Setembro de 2023.

MAIA, R. F. M.; FILHO, J. M. da S. M.; MILLER., J. Iara: Um chatbot para detecção de desinformação no whatsapp. 2022.

MARTINS, A. D. F.; CABRAL, L.; MOURÃO, P. J. C.; MONTEIRO, J. M.; MACHADO, J. C. Detection of misinformation about COVID-19 in brazilian portuguese whatsapp messages. In: MÉTAIS, E.; MEZIANE, F.; HORACEK, H.; KAPETANIOS, E. (Ed.). **Natural Language Processing and Information Systems - 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021, Saarbrücken, Germany, June 23-25, 2021, Proceedings**. Springer, 2021. (Lecture Notes in Computer Science, v. 12801), p. 199–206. Disponível em: https://doi.org/10.1007/978-3-030-80599-9_18. Acesso em: 27 de Dezembro de 2022.

MARTINS, A. D. F.; CABRAL, L.; MOURÃO, P. J. C.; MONTEIRO, J. M.; MACHADO, J. C. Detection of misinformation about COVID-19 in brazilian portuguese whatsapp messages using deep learning. In: **2021: Proceedings of the 36th Brazilian Symposium on Databases, SBBD 2021, Rio de Janeiro, Brazil (Online), October 4-8, 2021**. SBC, 2021. p. 85–96. Disponível em: <https://doi.org/10.5753/sbbd.2021.17868>. Acesso em: 29 de Dezembro de 2022.

MARTINS, A. I.; QUEIRÓS, A.; ROCHA, N. P.; SANTOS., B. S. Avaliação de usabilidade: Uma revisão sistemática da literatura. *Revista Ibérica de Sistemas e Tecnologias de Informação - RISTI*, 11/06/2013.

MARTINS., S. B. O conforto no vestuário: Uma interpretação da ergonomia metodologia de avaliação de usabilidade e conforto no vestuário. 2005.

MELLO, N. H. de; SANTOS., M. M. D. Estudo de técnicas para reconhecimento Ótico de caracteres e seu uso para detecção de placas de identificação automotivas brasileiras. *Revista de Engenharia e Tecnologia* ISSN, - 2176-7270, 2018.

MEMON, J.; SAMI, M.; KHAN, R. A.; UDDIN, M. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). **IEEE Access**, v. 8, p. 142642–142668, 2020.

NAGUMO, E.; TELES, L. A. F. A.; SILVA, L. A. d. A. EDUCAÇÃO E DESIGN: LETRAMENTO DIGITAL, CIÊNCIA E DIÁLOGO. **ETD EducaTica Digital**, scielo, v. 24, p. 220 – 237, 01 2022. ISSN 1676-2592. Disponível em: http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S1676-25922022000100220&nrm=iso. Acesso em: 21 de Fevereiro 2023.

OLIVEIRA., L. M. de. Adaptação transcultural, desenvolvimento e validação psicométrica da versão em libras da escala de usabilidade do sistema. Termo de Ciência e Autorização (TECA) para Disponibilizar Novas Versões Eletrônica de Teses, 2021.

PADRINI-ANDRADEA, L.; BALDAA, R. de C. X.; ARECOA, K. C. N.; BANDIERA-PAIVAA, P.; NUNESB, M. do V.; MARBAC, S. T. M.; CARVALHOD, W. B. de; RUGOLOE, L. M. S. de S.; ALMEIDAF, a. S. P. João Henrique Carvalho de; DUARTEH, J. L. M. B.; REGOI, M. A. S.; FERREIRAJ, D. M. de L. M.; FILHOK, N. A.; GUINSBURGA, R.; DINIZD, E. M. de A.; SANTOSL, J. P. F. dos; TESTONIM, D.; SILVAN, N. M. de Mello e; GONZALESO, M. R. C.; SILVAP, R. V. C. da; MENESESQ, J.; GONÇALVES-FERRIR, W. A.; SILVAR, R. P. e; BOMFIMS., O. Avaliação da usabilidade de um sistema de informação em saúde neonatal segundo a percepção do usuário. © 2018 Sociedade de Pediatria de São Paulo. Publicado por Zeppelini Publishers, 12/11/2018.

PALMA, J. D. G.; OLIVEIRA., R. A. Solução mobile – voice assisted chatbots. p. 6 – 88, 2019. Disponível em: https://repositorio.ul.pt/bitstream/10451/40279/1/ulfc125129_tm_Jo%C3%A3o_Palma.pdf. Acesso em: 09 de janeiro de 2023.

PAULA, L. S. de; GONÇALVES, T. G.; FERNANDES, T. V. R. B.; TRAVASSOS, G. H. Pipabot: um canal de comunicação para o pipa ufrj. In: **Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web**. Porto Alegre, RS, Brasil: SBC, 2019. p. 103–107. ISSN 2596-1683. Disponível em: https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/8146. Acesso em: 17 de Agosto de 2022.

PUTRI, F. P.; MEIDIA, H.; GUNAWAN, D. Designing intelligent personalized chatbot for hotel services. In: **Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence**. New York, NY, USA: Association for Computing Machinery, 2020. (ACAI '19), p. 468–472. ISBN 9781450372619. Disponível em: <https://doi.org/10.1145/3377713.3377791>. Acesso em: 02 de janeiro de 2022.

QIU, X.; OLIVEIRA, D. F.; SHIRAZI, A. S.; FLAMMINI, A.; MENCZER, F. Limited individual attention and online virality of low-quality information. **Nature Human Behaviour**, Nature Publishing Group, v. 1, n. 7, p. 0132, 2017.

REIS, J. C. S.; BENEVENUTO, F. Supervised learning for misinformation detection in whatsapp. In: PEREIRA, A. C. M.; ROCHA, L. C. D. da (Ed.). **WebMedia '21: Brazilian Symposium on Multimedia and the Web, Belo Horizonte, Minas Gerais, Brazil, November 5-12, 2021**. ACM, 2021. p. 245–252. Disponível em: <https://doi.org/10.1145/3470482.3479641>. Acesso em: 03 de março de 2022.

REIS, J. C. S.; MELO, P. de F.; GARIMELLA, K.; BENEVENUTO, F. Detecting misinformation on whatsapp without breaking encryption. **CoRR**, abs/2006.02471, 2020. Disponível em: <https://arxiv.org/abs/2006.02471>. Acesso em: 14 de maio de 2022.

RESENDE, G.; MELO, P. F.; SOUSA, H.; MESSIAS, J.; VASCONCELOS, M.; ALMEIDA, J. M.; BENEVENUTO, F. (mis)information dissemination in whatsapp: Gathering, analyzing and countermeasures. In: LIU, L.; WHITE, R. W.; MANTRACH, A.; SILVESTRI, F.; MCAULEY, J. J.; BAEZA-YATES, R.; ZIA, L. (Ed.). **The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019**. ACM, 2019. p. 818–828. Disponível em: <https://doi.org/10.1145/3308558.3313688>.

RESHMI, S.; BALAKRISHNAN. Implementation of an inquisitive chatbot for database supported knowledge bases. In: SA[⁠] DHANA (Ed.). [S. l.]: Kochi 682 022, India, 2016. p. 1174–1178.

RESHMI, S.; BALAKRISHNAN, K. Implementation of an inquisitive chatbot for database supported knowledge bases. **Sādhana**, v. 41, p. 1173–1178, 2016.

SÁ, I. C. de; GADELHA, T.; VINUTO, T.; SILVA, J. W. F. da; MONTEIRO, J. M.; MACHADO, J. C. A real-time platform to monitoring misinformation on telegram. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 25th International Conference on Enterprise Information Systems, ICEIS 2023, Volume 1, Prague, Czech Republic, April 24-26, 2023**. SCITEPRESS, 2023. p. 271–278. Disponível em: <https://doi.org/10.5220/0012039100003467>. Acesso em: 16 de Agosto de 2023.

SÁ, I. C. de; GALIC, L.; FRANCO, W.; GADELHA, T.; MONTEIRO, J. M.; MACHADO, J. C. BATMAN: A big data platform for misinformation monitoring. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). **Proceedings of the 25th International Conference on Enterprise Information Systems, ICEIS 2023, Volume 1, Prague, Czech Republic, April 24-26, 2023**. SCITEPRESS, 2023. p. 237–246. Disponível em: <https://doi.org/10.5220/0011995500003467>. Acesso em: 17 de Agosto de 2023.

SANTOS., R. M. Uma abordagem híbrida cnn-hmm para reconhecimento de fala tolerante a ruídos de ambiente. Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para obtenção do grau de Mestre em Ciência da Computação, 2016.

SHU, K.; SLIVA, A.; WANG, S.; TANG, J.; LIU, H. **Fake News Detection on Social Media: A Data Mining Perspective**. 2017.

SILVA, K.; TIerno, R.; BRANCHINE, S.; VILAÇA, D.; OLIVEIRA, F. H. Desenvolvimento de ferramenta de chatbot como solução para a comunicação do ifb. In: **Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 185–188. ISSN 0000-0000. Disponível em: https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/15375. Acesso em: 28 de setembro de 2022.

SU, Q.; WAN, M.; LIU, X.; HUANG, C.-R. Motivations, methods and metrics of misinformation detection: An nlp perspective. **Natural Language Processing Research**, v. 1, p. 1–13, 2020. ISSN 2666-0512. Disponível em: <https://doi.org/10.2991/nlpr.d.200522.001>.

VOSOUGHI S., R. D.; ARAL, S. The spread of true and false news online. In: SCITEPRESS (Ed.). [S. l.]: Science, 359:1146–1151, 2018.

WORTMANN, A.; BARAIS, O.; COMBEMALE, B.; WIMMER, M. Modeling languages in industry 4.0: An extended systematic mapping study. **Softw. Syst. Model.**, Springer-Verlag, Berlin, Heidelberg, v. 19, n. 1, p. 67–94, jan 2020. ISSN 1619-1366. Disponível em: <https://doi.org/10.1007/s10270-019-00757-6>. Acesso em: 08 de Abril de 2022.

YAN, R.; ZHAO, D. Smarter response with proactive suggestion: A new generative neural conversation paradigm. In: . [S. l.]: AAAI Press, 2018. (IJCAI'18), p. 4525–4531. ISBN 9780999241127.

ZAKI, M.; KANDEIL, D.; NEELY, A.; MCCOLL-KENNEDY, J. R. The fallacy of the net promoter score: Customer loyalty predictive model. Find out more about the Cambridge Service Alliance: Linkedin Group: Cambridge Service Alliance www.cambridgeservicealliance.org., 2016.

ZHI, Q.; METOYER, R. Gamebot: A visualization-augmented chatbot for sports game. In: **Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2020. (CHI EA '20), p. 1–7. ISBN 9781450368193. Disponível em: <https://doi.org/10.1145/3334480.3382794>. Acesso em: Acesso em: 19 de Janeiro de 2023.