Desenvolvimento de Sistema de Conversão de Voz com GPT

1st Higor David Oliveira
Departamento de Computação
Universidade Federal do Espírito Santo
Vitória, Brazil
higor.d.oliveira@edu.ufes.br

2nd Kaique Silva Passos
Departamento de Computação
Universidade Federal do Espírito Santo
Vitória, Brazil
kaique.passos@edu.ufes.br

3rd Victor Freitas Rocha
Departamento de Computação
Universidade Federal do Espírito Santo
Vitória, Brazil
victor.rocha@edu.ufes.br

***Abstract*—- This work presents the development of a system for seamless communication between humans and a GPT, utilizing voice-to-text and text-to-speech conversion. A review of STT, TTS, and virtual assistants is conducted. The methodology involves using OpenAI's Whisper for voice-to-text conversion, a GPT model (DeepSeek via the Groq Cloud API) for natural language processing, and Google's gTTS library for text-to-speech conversion. The development was carried out in Python, using tools such as Google Colab, Python Notebook, and Conda. The results successfully enabled voice communication with a GPT, achieving satisfactory time delay and accuracy on STT. The use of the Groq API to access the DeepSeek model worked as expected, though with an associated cost. Challenges were identified in adapting the GPT model's output for natural text-to-speech conversion, requiring the filtering of formatting characters. A demonstration video of the system is available on GitHub.**

***Index Terms*—- STT, TTS, GPT, Whisper (OpenAI)**

I. Introdução

A comunicação entre humanos e máquinas tem evoluído significativamente com o avanço das tecnologias de reconhecimento de fala e processamento de linguagem natural (*Natural Language Processing - NPL*. Nesse cenário, as interfaces de voz (*Voice User Interfaces* - VUIs) têm se destacado, proporcionando interações mais naturais e intuitivas. Segundo Hoy (2018) [1], a popularização de assistentes virtuais como Siri, Alexa e Google Assistant evidencia a importância dessas interfaces para tornar o acesso à tecnologia mais simples e eficiente.

Nesse contexto, a integração entre VUIs e modelos avançados de inteligência artificial recentes, como o *ChatGPT* [2] (*Generative Pré-trained Transformers* - GPT) e o *DeepSeek* [3], potencializa a comunicação ao combinar a capacidade de converter voz em texto e, posteriormente, de interpretar e gerar respostas em linguagem natural [4]. Porcheron et al. (2018) [4] ressaltam que a fluidez e naturalidade nas interações são fundamentais para a aceitação desses sistemas em diversas aplicações, desde o entretenimento até setores mais críticos como o de saúde e o de educação.

O presente trabalho tem como objetivo desenvolver um sistema que permita uma comunicação fluida entre humanos e uma GPT, utilizando conversão de voz para texto e vice-versa. Com o fim de proporcionar uma experiência de conversação contínua e natural, tornando a interação com a inteligência artificial GPT mais acessível e eficiente. Para isso, o sistema deve ser capaz de converter a voz do usuário em texto, processar essa entrada por meio de um modelo de linguagem baseado em GPT e converter a resposta gerada pela GPT de volta para voz.

O documento a seguir está estruturado da seguinte forma: a seção de II Trabalhos Relacionados trata a respeito de artigos, estudos, projetos e produtos que se assemelham com a proposta do presente trabalho; III Metodologia descreve as tecnologias utilizadas e as etapas executadas para atingir o objetivo deste trabalho; em IV Experimentos, são detalhados os passos executados para implementação da metodologia definida anteriormente; em V Resultados são apresentados os resultados, são citados os desafios enfrentados e, por fim, são realizadas análises críticas a respeito do trabalho, deixando sugestões para aprimoramento futuro.

II. Trabalhos relacionados

Existem diversos trabalhos relacionados à temática que englobam VUIs, Reconhecimento Automático de Fala (*Automatic Speech Recognition* - ASR) ou Conversão Fala para Texto (*Speech-to-text* - STT), modelos de conversação GPT, Conversão de Texto em Voz (*Text-to-speech - TTS*), Assistentes Pessoais de Inteligência Artificial ou suas aplicações. A seguir serão explorados os trabalhos relacionados a alguns desses tópicos.

Primeiramente, a respeito de ASR ou STT, sabe-se que é um desafio que tem sido explorado há décadas, com uma das primeiras máquinas de ASR, o "*Shoebox*", desenvolvido em 1962 pela IMB, capaz de reconhecer 16 palavras diferentes [7]. Recentemente, pesquisam exploram redes neurais para realização da tarefa [8]. Prabhavalkar indica que a introdução de modelagens baseadas em *deep leaning* trouxeram ganhos de 50% relativo sobre a Taxa de Error de Palavras (*Word Error Rate* - WER) em alguns datasets. Atualmente existem diversos modelos baseados em redes neurais profundas, como o *Speech-to-Text* da *Google Chirp* [11], como o *SeamlessM4T* da MetaAI [10], o *Whisper* da *OpenAI* [5], modelos treinados em milhões de horas de áudio, multilíngue, multitarefas e com bilhões de parâmetros.

A respeito do desafio de reproduzir a voz humana, sabe-se que o primeiro sistema a tentar realizar tal tarefa de TTS foi Voder, uma máquina que tentava reproduzir sons humanos eletricamente [12]. Um dos primeiros registros do uso de técnicas de *deep learning* no desafio foi por meio do modelo *WaveNet* [13], que tentava fazer a predição (*forecasting)* de áudio, inspirado na PixelRNN para imagens, ambos de Aaron van den Oord, 2016 [14]. Pesquisas mais recentes incluíram o uso de modelos grandes (*large models)* para o problema, como o StyleTTS 2 [15] e o NaturalSpeech [16], ambos de 2023, com resultados "*human-level*" com capacidade para se adaptar ao falador "*zero-shot*". Atualmente existem modelos como o do *Google*, *gTTS* [6], da *Microsoft*, *Azure* TTS [17] e da *OpenAI*, *Whisper* (capaz de TTS e SST) [5].

Considerando estritamente a ideia de uma comunicação fluída entre o ser humano e uma GPT, há uma ideia de produto que apareceu no mercado nos últimos anos, os dispositivos pessoais assistentes de IA ("*personal AI assistant devices*"). Dispositivos como o Rabbit R1 [19], o *Humane Ai Pin* [20], lançados em 2024, e o Friend Ai [18], com lançamento previsto para 2025, tentam naturalizar a interação com *chatBots* com os modelos de conversação. Tais dispositivos se disfarçam como acessórios de vestimenta e se tornam microfones e câmeras para interagir com os modelos de conversação, facilitando o acesso, trazendo funcionalidades agradáveis ao cotidiano.

Existem, ainda, projetos como o Hertz-dev [21] e o MThreads AI [22] que caminham a pesquisa para um modelo de conversão baseado em áudio, sem a transcrição para texto. Pulando assim etapas de texto, tentando trazer etapas de codificação e decodificação em áudio para uma comunicação de duas vias.

III. Metodologia

A fim de alcançar o objetivo do trabalho, o sistema desenvolvido executa as seguintes etapas: inicialmente, é coletado uma fala do usuário, por meio de um microfone do dispositivo. Logo em seguida, o áudio é dado como entrada para o modelo da OpenAI baseado em GPT de reconhecimento automático de fala (*Automatic Speech Recognition* - ASR), Whisper [5]. O texto de saída do Whisper é então usado como entrada para um modelo de conversão também baseado em GPT, ou o *ChatGPT* ou o *DeepSeek*, gerando uma resposta processada em texto. Posteriormente, a resposta é convertida em voz pela biblioteca de conversão de texto em voz (*Text-to-speech - TTS*), *gTTS* da Google [6]. Tal áudio é a saída para o usuário. O código utilizado para realizar tal processo foi escrito em Python, usando de ferramentas como o Colab e o Python Notebook para auxiliar o desenvolvimento. A figura 1 sumariza a aplicação.

A execução de um modelo de linguagem GPT para processar a solicitação do usuário não é o foco deste trabalho, mas uma etapa necessária. Desta forma, pensando em flexibilizar a escolha do modelo e diminuir a quantidade de recursos para execução do código, a solicitação do usuário será processada usando processamento em nuvem, por meio de uma API e um serviço pago.

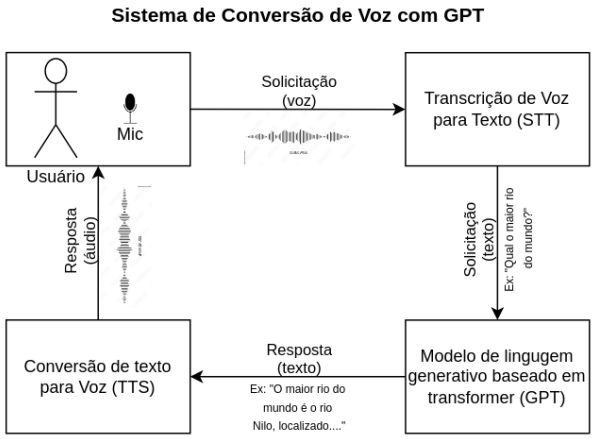

Fig. 1. Diagrama do Sistema Proposto para Conversão de Voz com GPT.

As subseções a seguir discorrem a respeito das tecnologias utilizadas e reforça as principais contribuições para alcançar o objetivo do trabalho.

A. Linguagem de programação: Python

A linguagem de programação *Python* se destaca como uma excelente ferramenta para o desenvolvimento de sistemas de inteligência artificial devido à sua sintaxe clara e intuitiva, que permite a rápida prototipagem [23]. Essa linguagem conta com um vasto ecossistema de bibliotecas e frameworks amplamente utilizados para a implementação de algoritmos de aprendizado de máquina e processamento de linguagem natural como *TensorFlow*, *PyTorch* e *scikit-learn* [24].

Além disso, existe uma forte comunidade de desenvolvedores e pesquisadores que ativamente buscam melhorar e evoluir as ferramentas, promovendo a integração de funcionalidades avançadas por meio de Interfaces de Programação de Aplicações (*Application Programming Interfaces* - APIs), entre as mais diversas ferramentas. Entre essas ferramentas, se encontram as ferramentas de conversão de voz em texto e vice-versa, tornando a linguagem de programação Python uma boa escolha para o sistema proposto.

B. Conversão de voz em texto: Whisper da OpenAI

A ferramenta escolhida para conversão de voz em texto foi a Whisper da OpenAI. Conforme Radford et al., 2022 [25], a Whisper é uma ferramenta avançada, capaz de transcrever áudios com alta precisão mesmo em ambientes desafiadores. O autor ainda destaca que, por ser desenvolvido com tecnologias de deep learning, o modelo é projetado para performar em diversos idiomas e variações de sotaque, o que vem a flexibilizar a utilização e a melhorar a qualidade da aplicação no sistema proposto. A figura 2 sumariza a estrutura do modelo.

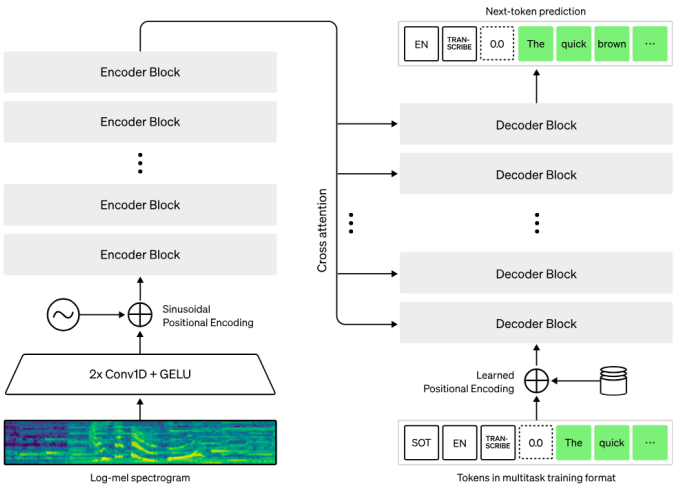

Fig. 2. Sumarização do modelo de reconhecimento automático de fala (ASR) Whisper da OpenAI [5].

A entrada para o modelo Whisper é composta por sequências de áudio. Internamente, o áudio de entrada é reamostrado à 16kHz e representado por espectrograma com 80 canais de magnitude, 25ms de janela e 10ms de deslocamento, chamado espectrograma mel do áudio [28]. O espectograma mel de áudio, trata-se de uma forma de representar visualmente as frequências predominantes do som, que facilita a identificação e interpretação de padrões acústicos para sistemas de inteligência artificial [26].

Os espectogramas são normalizado de -1 a 1 e alimentados a uma rede neural composta por duas camadas de convolução e uma camada de ativação GELU. A saída então é alimentada a um codificador posicional senoidal, gerando *embeddings* de entrada para o modelo baseado em transformers GPT.

Por meio de decodificadores, uma saída de texto é gerada e realimentada no modelo com adição da informação posicional senoidal, gerando a próxima palavra da saída. Essa realimentação continua até o fim do áudio de entrada. Dessa forma, o Whisper combina múltiplos componentes: um codificador que processa a entrada de áudio e um decodificador que gera a saída de texto.

Assim também, o modelo Whisper foi treinado e disponibilizado com diferentes números de parâmetros. A figura 3 mostra a relação Tamanho x Número de parâmetros.

Optou-se por usar o Whisper no presente projeto, pela capacidade de transcrição de áudio em vários idiomas, pelo acesso facilitado ao modelo via API python, pela precisão reportada no artigo base do modelo e no github [27] e por ser um modelo baseado em GPT de código aberto.

C. GPT API python: Groq API

Para processar as solicitações ao GPT vindas do usuário usou-se o *Groq Cloud* API [35]. *Groq* é uma empresa focada em serviços de aceleração de IA em hardware e modelos de linguagem, oferecendo soluções de processamento de alta velocidade para LLMs (Large Language Models) [34]. O uso da ferramenta em um sistema *Python* pode proporcionar menor tempo de espera pela resposta, tornando o sistema mais eficiente. Alé disso, a ferramenta possui compatibilidade para diferentes modelos como *Llama*, *Deepseek*, *Qwen* e outros [34], o que flexibiliza o sistema projetado para este trabalho.

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Fig. 3. Tabela mostrando detalhes do modelo da família Whisper. Tabela retirada de Radford et al. 2022 [25].

D. Conversão de texto em voz: gTTS, Text-To-Speech da Google

Já para a tarefa de conversão de texto para voz (TTS), a biblioteca gTTS [29], oferecida pelo Google, possibilita transformar textos em áudio com suporte a diversas opções de linguagens e a mudança de pronunciação. Essa solução se integra de forma prática ao sistema proposto, permitindo que as respostas geradas em texto sejam convertidas em voz com diferentes tonalidades, contribuindo para uma interação mais dinâmica e acessível para o usuário.

Ademais, o gTTS possui um suporte direto a linguagem de programação Python e uma extensa documentação online, além da existência de códigos exemplo [6].

E. Google Colab Notebook

O *Google Colab* Notebook é uma ferramenta baseada em nuvem que permite a criação, edição e execução de código Python diretamente em um ambiente interativo, usando servidores conectados por rede para execução do ambiente [30]. Ele oferece uma interface semelhante ao Jupyter Notebook, que permite rapidamente configurar um ambiente local para a nuvem e vice-versa. Assim também, o *Colab* tem sua integração com o *Google Drive*, facilitando o armazenamento e compartilhamento de projetos.

Entre os servidores oferecidos pelo Google Colab, é possível obter máquinas com recursos poderosos para computação de modelos neurais, como GPUs e TPUs, permitindo a execução de códigos de forma versátil. Para prototipagem rápida, o Colab permite testar e validar ideias de maneira ágil, sem a necessidade de configurações complexas de ambiente, tornando-o uma ferramenta valiosa e de boa escolha para este projeto.

IV. Experimentos

O sistema do presente projeto foi implementado usando *Python* dentro de ambiente em nuvem usando o *Google Colab* e também dentro de ambiente local usando *Conda*. O código foi versionado usando o *GitHub* [36].

O *Whisper* foi testado em tamanhos diferentes e em versões diferentes (*Whisper* API e o *Faster Whisper* [31], uma reimplementação do *Whisper* pela *SYSTRANSoft* [32] usando uma engine mais rápida).

Para testar o sistema, realizou-se solicitações simples, como "Qual o maior rio do mundo?", "Qual a diferença entre um círculo e uma elipse?" e perguntas similares. Posteriormente, adicionou-se códigos para ajustar o texto de saída gerado pelo GPT para o TTS.

O sistema final foi testado usando o microfone de cada membro do projeto. Um vídeo do funcionamento do sistema foi gravado e posto no mesmo repositório do *Github* com o código [36].

V. Resultados

Para a conversão voz em texto, por meio de frameworks do *Python*, foi possível executar o *Whisper* localmente. O tempo para a conversão STT foi de até 2s, porém, o uso de versões maiores do modelo resultavam em um tempo maior na resposta. Na média, o tempo de resposta do sistema era de aproximadamente 4s. O desempenho da conversão foi considerado satisfatória pelos autores, tendo poucos erros para a conversão em português.

O processamento da solicitação do usuário ao GPT usando o *Groq API* funcionou conforme esperado, o modelo escolhido foi o *DeepSeek*. O modelo respondia às solicitações com desempenho satisfatório. Foi necessário adquirir uma chave de execução da *Groq API*, o que resulta em um custo para solicitação feita, custo que deve ser considerado ao rodar a aplicação.

A utilização do gTTS para conversão da resposta para voz foi feita sem dificuldades. Ademais, foram encontradas algumas dificuldades para a adaptação da resposta gerada pelo modelo de linguagem GPT para voz. A saída gerada pelo *DeepSeek* continha caracteres usados para formatação, como #, /, \, *. Esses caracteres foram pronunciados na conversão TTS, o que soava não-natural, piorando a experiência da comunicação. Por isso, foi necessário adicionar um código para filtrar do texto esses caracteres.

Um vídeo de demonstração de funcionamento do sistema pode ser visto no GitHub deste projeto [36].

Considerando o objetivo de desenvolver um sistema que permita uma comunicação fluida entre humanos e uma GPT, utilizando conversão de voz para texto e vice-versa, o sistema final conseguiu atingir o objetivo.

Referências

[1] Hoy, Matthew. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. Medical Reference Services Quarterly. 37. 81-88. 10.1080/02763869.2018.1404391.

[2] OpenAI, "ChatGPT," OpenAI, 2023. [Online]. Disponível em: https://openai.com/index/chatgpt/. Acesso em: 27-fev-2025.

[3] DeepSeek, "DeepSeek," DeepSeek, 2023. [Online]. Disponível em: https://www.deepseek.com/. Acesso em: 27-fev-2025.

[4] Porcheron, Martin & Fischer, Joel & Reeves, Stuart & Sharples, Sarah. (2017). Voice Interfaces in Everyday Life. 10.1145/3173574.3174214.

[5] OpenAI, "Whisper," OpenAI, 2023. [Online]. Disponível em: https://openai.com/index/whisper/. Acesso em: 03-mar-2025.

[6] P. Durette, "gTTS," GitHub, 2023. [Online]. Disponível em: https://github.com/pndurette/gTTS. Acesso em: 03-mar-2025.

[7] IBM, "Speech recognition," IBM, 2023. [Online]. Disponível em: https://www.ibm.com/br-pt/topics/speech-recognition. Acesso em: 10-mar-2025.

[8] de Sá, João Manuel Alves Mourão. "Reconhecimento de fala em português de Portugal num contexto com poucos recursos" Universidade do Porto - FCUP. [Online]. Disponível em: https://repositorio-aberto.up.pt/bitstream/10216/139258/2/526280.pdf. Acesso em: 10-mar-2025.

[9] Prabhavalkar, Rohit & Hori, Takaaki & Sainath, Tara & Schlüter, Ralf & Watanabe, Shinji. (2023). End-to-End Speech Recognition: A Survey. 10.48550/arXiv.2303.03329.

[10] Communication, Seamless & Barrault, Loïc & Chung, Yu-An & Meglioli, Mariano & Dale, David & Dong, Ning & Duquenne, Paul-Ambroise & Elsahar, Hady & Gong, Hongyu & Heffernan, Kevin & Hoffman, John & Klaiber, Christopher & Li, Pengwei & Licht, Daniel & Maillard, Jean & Rakotoarison, Alice & Sadagopan, Kaushik & Wenzek, Guillaume & Ye, Ethan & Wang, Skyler. (2023). SeamlessM4T-Massively Multilingual & Multimodal Machine Translation. 10.48550/arXiv.2308.11596.

[11] Google Cloud, "Bringing the power of large models to Google Cloud's Speech API," Google Cloud Blog, 2023. [Online]. Disponível em: https://cloud.google.com/blog/products/ai-machine-learning/bringing-power-large-models-google-clouds-speech-api. Acesso em: 10-mar-2025.

[12] What is the Voder?, "What is the Voder?" What is the Voder, 2023. [Online]. Disponível em: https://www.whatisthevoder.com/. Acesso em: 10-mar-2025.

[13] oord, Aaron & Dieleman, Sander & Zen, Heiga & Simonyan, Karen & Vinyals, Oriol & Graves, Alex & Kalchbrenner, Nal & Senior, Andrew & Kavukcuoglu, Koray. (2016). WaveNet: A Generative Model for Raw Audio. 10.48550/arXiv.1609.03499.

[14] Oord, Aaron & Kalchbrenner, Nal & Vinyals, Oriol & Espeholt, Lasse & Graves, Alex & Kavukcuoglu, Koray. (2016). Conditional Image Generation with PixelCNN Decoders. 10.48550/arXiv.1606.05328.

[15] Li, Yinghao & Han, Cong & Raghavan, Vinay & Mischler, Gavin & Mesgarani, Nima. (2023). StyleTTS 2: Towards Human-Level Text-to-Speech through Style Diffusion and Adversarial Training with Large Speech Language Models. 10.48550/arXiv.2306.07691.

[16] Zeqian Ju and Yuancheng Wang and Kai Shen and Xu Tan and Detai Xin and Dongchao Yang and Yanqing Liu and Yichong Leng and Kaitao Song and Siliang Tang and Zhizheng Wu and Tao Qin and Xiang-Yang Li and Wei Ye and Shikun Zhang and Jiang Bian and Lei He and Jinyu Li and Sheng Zhao. (2024). NaturalSpeech 3: Zero-Shot Speech Synthesis with Factorized Codec and Diffusion Models. 10.48550/arXiv:2403.03100

[17] Microsoft, "AI Speech," Microsoft Azure, 2023. [Online]. Disponível em: https://azure.microsoft.com/pt-br/products/ai-services/ai-speech. Acesso em: 10-mar-2025.

[18] Kodjima, "Friend - Open Source AI Wearable Recording Device," Kickstarter, 2023. [Online]. Disponível em: https://www.kickstarter.com/projects/kodjima333/friend-open-source-ai-wearable-recording-device. Acesso em: 10-mar-2025.

[19] Rabbit Tech, "Rabbit R1," Rabbit Tech, 2023. [Online]. Disponível em: https://www.rabbit.tech/rabbit-r1. Acesso em: 10-mar-2025.

[20] Humane AI Inc, "Humane," Humane, 2023. [Online]. Disponível em: https://humane.com/. Acesso em: 10-mar-2025.

[21] Hertz Dev, "Hertz Dev," SI Inc., 2023. [Online]. Disponível em: https://si.inc/hertz-dev/. Acesso em: 10-mar-2025.

[22] Peng Wang and Songshuo Lu and Yaohua Tang and Sijie Yan and Wei Xia and Yuanjun Xiong. (2024). A Full-duplex Speech Dialogue Scheme Based On Large Language Models. arXiv.2405.19487

[23] Oliphant, Travis. (2007). Python for Scientific Computing. Computing in Science & Engineering. 9. 10-20. 10.1109/MCSE.2007.58.

[24] Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 12.

[25] Radford, Alec & Kim, Jong & Xu, Tao & Brockman, Greg & McLeavey, Christine & Sutskever, Ilya. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. 10.48550/arXiv.2212.04356.

[26] Mehrish, Ambuj & Majumder, Navonil & Bharadwaj, Rishabh & Mihalcea, Rada & Poria, Soujanya. (2023). A review of deep learning techniques for speech processing. Information Fusion. 99. 101869. 10.1016/j.inffus.2023.101869.
[27] OpenAI, "Whisper," GitHub, 2023. [Online]. Disponível em: https://github.com/openai/whisper. Acesso em: 03-mar-2025.
[28] Radford, Alec & Kim, Jong & Xu, Tao & Brockman, Greg & McLeavey, Christine & Sutskever, Ilya. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. 10.48550/arXiv.2212.04356.
[29] gTTS Documentation, "gTTS Documentation," 2023. [Online]. Disponível em: https://gtts.readthedocs.io/en/latest/. Acesso em: 05-mar-2025.
[30] Google, "Google Colaboratory," Google, 2023. [Online]. Disponível em: https://colab.google/. Acesso em: 15-mar-2025.
[31] SYSTRAN, "faster-whisper," GitHub, 2023. [Online]. Disponível em: https://github.com/SYSTRAN/faster-whisper. Acesso em: 05-mar-2025.
[32] SYSTRAN, "SYSTRAN GitHub," GitHub, 2023. [Online]. Disponível em: https://github.com/SYSTRAN. Acesso em: 10-mar-2025.
[33] Adobe Stock, "Black soundwave equalizer isolated on white background, abstract music wave, radio signal frequency and digital voice visualisation, audio sound symbol," Adobe Stock, 2023. [Online]. Disponível em: https://stock.adobe.com/br/images/black-soundwave-equalizer-isolated-on-white-background-abstract-music-wave-radio-signal-frequency-and-digital-voice-visualisation-audio-sound-symbol/368444653. Acesso em: 03-mar-2025.
[34] Groq, "Groq," Groq, 2023. [Online]. Disponível em: https://groq.com/. Acesso em: 10-mar-2025.
[35] Groq, "Groq Documentation," Groq, 2023. [Online]. Disponível em: https://console.groq.com/docs/overview. Acesso em: 03-mar-2025.
[36] Oliveira, Higor & Passos, Kaique & Rocha, Victor "Desenvolvimento de Sistema de Conversão de Voz com GPT," GitHub, 2025. [Online]. Disponível em: https://github.com/vfrocha/Sistema-de-Conversao-de-Voz-com-GPT. Acesso em: 23-mar-2025.