Voice Analysis: uma proposta de aplicativo de análise de oratória

Eduardo Dalcin[1], Nicolas Lima Morche[2]

[1]Professor Orientador – Instituto Federal Farroupilha Campus Panambi (IFFAR) – Panambi – RS - Brasil

[2]Aluno do Curso Superior de Tecnologia em Sistemas para Internet – Instituto Federal Farroupilha Campus Panambi (IFFAR) – Panambi – RS - Brasil

eduardo.dalcin@iffarroupilha.edu.br,
nicolas.20200018409@aluno.iffar.edu.br

Abstract. *Developing communication skills to the public is important for all students, teachers and market professionals. Based on this, the present project, based on applied research, aims to create an application to identify repeated words and extended words during speech. Throughout the research, the following technologies were used: Visual Studio Code, JavaScript, React Native, Expo Go, Python, Django and Assembly.AI. The system demonstrated accuracy in identifying extended and repeated words, proving the possibility of using this application for public speaking improvement projects.*

Resumo. *Desenvolver a habilidade de comunicação ao público é importante para todos os alunos, professores e profissionais do mercado. Com base nisto, o presente projeto baseado em uma pesquisa aplicada, visa criar um aplicativo para identificar as palavras repetidas e palavras estendidas durante o discurso. Ao longo da pesquisa foram utilizadas as tecnologias: Visual Studio Code, JavaScript, React Native, Expo Go, Python, Django e Assembly.AI. O sistema demonstrou precisão na identificação de palavras estendidas e repetidas, comprovando a possibilidade do uso desse aplicativo para projetos de aperfeiçoamento de oratória.*

1. Introdução

No ano de 1963, perto do *Lincoln Memorial*, uma das mais influentes apresentações do mundo foi realizada por Marthin Luther King Jr., conhecida como *'I have a dream'*. Onde o orador, com muita prática e treinamento da habilidade de comunicação, influenciou o mundo inteiro a seguir seus ideais de construir um país livre da segregação e livre de racismos (Carson; Lewis, 2023).

Diante deste contexto histórico, podemos confirmar o quanto a oratória é uma habilidade importante a ser desenvolvida no mundo para potencializar processos de interação. Porém, atualmente, após o período pandêmico, com a evolução tecnológica acelerada, é observável uma queda na habilidade de comunicação e interação das pessoas como a de falar em público. Como consequência, as pessoas muitas vezes se tornam mais tímidas, ansiosas, bloqueadas pelo medo e a falta de prática dessa atividade. De acordo com um estudo realizado no Programa de Pós-Graduação em Ciências Fonoaudiológicas da Universidade Federal de Minas Gerais (UFMG), 59,7%

dos participantes demonstravam sintomas de ansiedade como: respiração ofegante e taquicardia e, também, tinham medo de falar em público (Redação Tribuna Online, 2022).

De acordo com Nurmalasari, Tahir e Korompot (2023), em um estudo científico sobre apresentações com estudantes, após a prática de falar em público houve um aumento de autoconfiança nos mesmos. De acordo com Arora (2021), apresentações em público ajudam as pessoas a pensar de forma criativa e crítica, aumenta nossa confiança, aperfeiçoa nossas habilidades de comunicação e, como consequência, aumenta nossa influência sobre as pessoas. Outro estudo identificou que, em três empresas de consultoria com 138 funcionários, confirmou-se que a comunicação eficaz tem um efeito positivo no desempenho organizacional (Musheke; Phiri, 2021).

Portanto, com a expansão do uso da tecnologia da informação e comunicação (TIC) evoluindo exponencialmente com o uso da inteligência artificial, *machine learning*, *big data,* etc… Desse modo, validar pesquisas e projetos com o intuito de auxiliar pessoas a melhorar suas habilidades comunicacionais e autoconfiança frente ao público é algo relevante.

Com base nas informações anteriores, é esperado termos informações suficientes para explorar a possibilidade da criação de um aplicativo e validar sua viabilidade para auxílio na melhoria da habilidade da oratória de alunos, educadores e profissionais. Assim, objetivo geral dessa pesquisa é criar um aplicativo de identificação de palavras repetidas e palavras estendidas no discurso. Para o desenvolvimento do aplicativo, foram organizadas as seguintes etapas:

1. Identificação das tecnologias mais adequadas para criação do aplicativo e do algoritmo de análise de discurso;

2. Desenvolvimento do algoritmo de análise de discurso;

3. Desenvolvimento da interface;

4. Implementação do aplicativo e comunicação com servidor;

5. Validação do aplicativo.

2. Fundamentação Teórica

Neste trabalho, diante dos artigos analisados, programas e bibliotecas utilizadas e desenvolvidos, foram, teoricamente, baseadas em um tipo de rede neural artificial que, atualmente, é uma das tecnologias mais utilizadas em pesquisas científicas, relacionadas ao Reconhecimento Automático de Fala (ASR), conhecido também como: Rede Neural Profunda Discriminativa em conjunto com o HMM (Modelo Oculto de Markov Generativo) (Ravanelli, 2018).

De acordo com Ravanelli e Brakel(2018, p. 1):

> O DNN (*Deep Neural Network*) é normalmente empregado para fins de modelagem acústica para prever alvos telefônicos dependentes do contexto. As previsões de nível acústico são posteriormente incorporadas em uma estrutura baseada em HMM, que também integra transições telefônicas, léxico e informações de modelo de linguagem para recuperar a sequência final de palavras.

Durante a pesquisa, não foram encontrados estudos relacionados à criação de aplicativos focados na comunicação, mas sim, artigos científicos que tiveram a mesma proposta de, através da tecnologia, obter informações sobre a comunicação e, em alguns casos, gerar *feedback* para o usuário. Um deles foi o *RAP System*, sistema desenvolvido objetivando retornar uma avaliação sobre erros básicos de apresentação ao público para alunos sobre sua oratória, analisando postura e voz. Comprovou-se que o sistema retornou feedbacks com uma qualidade positiva para o público-alvo, indicando que sistemas podem auxiliar pessoas a aperfeiçoarem sua oratória (Ochoa; Domínguez; Guamán; Maya; Falcones; Castels, 2018).

Outro trabalho a ser mencionado foi o de um desenvolvimento de modelo de rede neural audiovisual para avaliação da oratória considerando postura corporal, atributos faciais e características acústicas. Para criação do modelo, analisaram separadamente os elementos da oratória: postura corporal, expressão facial e áudio da fala dos autores das palestras do *Ted Talks*, totalizando mais de 290 horas de discurso para análise. Com isto, mesmo com um conjunto de dados desafiador, foi possível estabelecer um desempenho moderadamente bom para as classificações do modelo (Michelson, 2021).

Dos trabalhos citados anteriormente, enquanto o RAP System é concentrado na avaliação de erros básicos de apresentação (Ochoa; Domínguez; Guamán; Maya; Falcones; Castels, 2018), e o modelo de rede neural realiza avaliação dos elementos da oratória (Michelson, 2021), o aplicativo proposto será concentrado em elementos básicos da fala dos oradores como repetição de palavras e fluidez na articulação.

3. Metodologia

Esta seção aborda as metodologias estabelecidas, tecnologias e algoritmos utilizados e, também, os testes de validação para atingir o objetivo geral.

3.1. Tipo de Pesquisa

O tipo de pesquisa utilizada no presente estudo é a pesquisa aplicada. De acordo com Frossard (2022), “A pesquisa aplicada é conhecida, no campo científico, por aqueles processos que buscam converter o conhecimento puro, ou seja, as teorias, em conhecimento prático e útil para a sociedade.”. Logo, foi realizada a seleção dos conhecimentos relacionados a codificação de programas e oratória para criar uma solução prática.

3.2. Ferramentas utilizadas

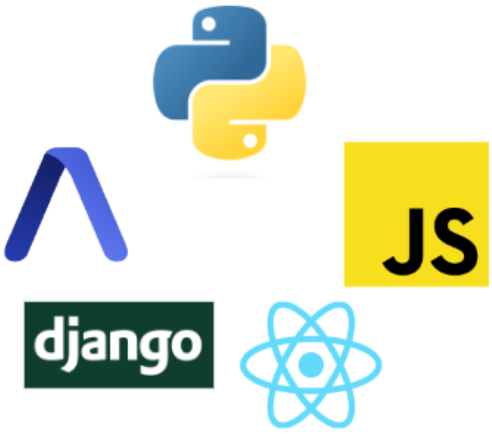
Antes da execução das atividades de desenvolvimento, decorreu a validação das ferramentas no desenvolvimento. *Visual Studio Code* foi utilizado como Ambiente de Desenvolvimento Integrado (IDE) na codificação em *script Python* e *React Native* que usa *JavaScript*. Para simulações do *app* deu-se o uso do *Expo Go*. Já para criação do design do aplicativo, foi utilizado o Figma, editor gráfico de vetor e de prototipagem.

3.3. Tecnologias utilizadas

Para o andamento das tarefas e implementações de código, utilizou-se as seguintes tecnologias (conforme Figura 1): *React Native*, *framework cross-platform* que permite a criação de interfaces de alta performance para as plataformas Android e iOS. Para

backend, optou-se pelo Django[1], devido a sua modularidade, eficiência e base sólida para gerenciar a lógica de negócios e interações com o banco de dados.

Figura 1. Principais tecnologias utilizadas na pesquisa



Fonte: Elaborado pelo autor (2023).

Vale a pena ressaltar algumas informações acerca do uso dessas tecnologias enumeradas, quanto ao contexto técnico de desenvolvimento do aplicativo. Assim, Roveda (2021), descreve sobre essas tecnologias:

> [...] baseia-se no modelo MVT -- ou *Model-View-Template*, os três aspectos de uma aplicação web trabalhados pelo Django. Mais detalhadamente, são eles:
>
> *Model*: o modelo é a estrutura que representa os dados dessa aplicação, ou seja, possui ligação direta com um banco de dados. Esta é a camada de abstração feita para manipular, incluir ou excluir dados. Sempre que um *model* é criado, o Django fornece uma API para esta manipulação.
>
> *View*: o *view* é uma função Python feita para receber uma requisição e enviar uma resposta em retorno. É aqui onde os dados são extraídos e produzem uma resposta.
>
> *Template*: esta é a camada onde o *template* da aplicação é produzido através de artefatos compreendidos pelo navegador *web*. Esta é a parte que diz respeito a tudo que o usuário final é capaz de visualizar em seu dispositivo.

De acordo com o processo de desenvolvimento do aplicativo, é relevante destacar alguns apontamentos técnicos: para cada um desses pilares descritos anteriormente é utilizado uma pasta ou arquivo dentro do projeto do Django para configurá-lo. O pilar *model* é todo construído dentro do models.py onde definimos uma classe que representa o modelo e, dentro dela, as variáveis e funções representam as

[1] *framework* de código aberto criado em Python de alto nível (Roveda, 2021).

suas propriedades e métodos. Já para o pilar *View*, é utilizado o arquivo views.py, onde cada classe representa uma visualização de resposta diferente e, enquanto para o pilar *template* é usado no arquivo *settings.py* do projeto, onde se configura as opções dos *templates* disponíveis.

Outro elemento que devemos considerar no Django é o *settings*, onde configuramos os *middleware*, que permite a inclusão de funcionalidades globais, como autenticação e compressão, no *pipeline* de processamento de solicitações e respostas. Além disso, o arquivo *settings.py* é configurado para gerenciar aspectos importantes do aplicativo, como configurações de banco de dados e outras opções de personalização.

Nesse caso também foi necessário o uso de um arquivo *serializers.py* para conversão de objetos enviados do aplicativo em um formato facilmente armazenável e manipulável na *view*, para assim, usar as bibliotecas para processamento de áudio.

3.4. Algoritmo de Reconhecimento de Voz

O serviço de reconhecimento de voz do *Assembly.AI* foi selecionado para ser o componente principal de análise de oratória. Visto que, é uma solução avançada para transcrição automática de áudio para texto, trazendo não só apenas as palavras usadas no áudio, como também, o tempo delas, tornando possível a análise de velocidade do discurso. A ferramenta funciona da seguinte forma: é enviado o arquivo do áudio para o serviço de transcrição do *Assembly.AI*, o serviço processa o áudio e gera a transcrição, retornando com uma saída formatada em JSON (*JavaScript Object Notation*), mais detalhada no quadro 1.

Quadro 1. Propriedades do objeto retornado do Assembly.AI

Chave	Tipo	Descrição	Exemplo
transcript.text	*string*	A transcrição do arquivo de áudio.	“A família é muito importante para o núcleo interno das pessoas”
transcript.words	*array*	Uma matriz contendo informações sobre cada palavra	[[…], […], […]. ...]
transcript.words[i].text	*string*	O texto da i-ésima palavra na transcrição	[[“A”,…], [“família”,…], [“é”,…]]
transcript.	*number*	O início de	[[80,…],

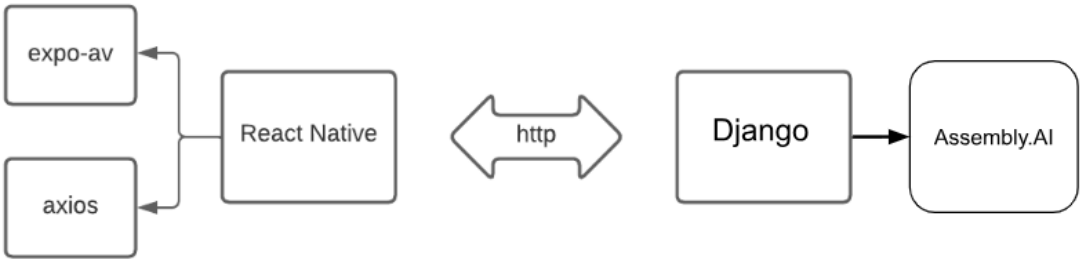
words[i].start		quando esta palavra é falada no arquivo de áudio, em milissegundos	[100,…], [700,…] ...]
transcript.words[i].end	*number*	O final de quando esta palavra é falada no arquivo de áudio, em milissegundos	[[100,…], [600,…], [780,…] ...]
transcript.words[i].confidence	*number*	A pontuação de confiança para a transcrição da i-ésima palavra	[[75.7,…], [60.7,…], [89.6,…] ...]

Fonte: https://www.assemblyai.com/docs/models/speech-recognition

3.5. Arquitetura do sistema

A arquitetura do sistema é composta pelos seguintes elementos: *React Native* como *framework* para aplicativos mobile, *Axios* biblioteca do *React Native* para realizar requisições HTTP (*Hypertext Transfer Protocol*) e *Expo-av* dependência para gravar áudios pelo microfone do celular e manipulá-los.

Figura 2. Arquitetura do sistema



Fonte: Elaborado pelo autor (2023).

Já no *backend* utilizamos Django como servidor e *Assembly.AI*, responsáveis pelos algoritmos de reconhecimento de voz.

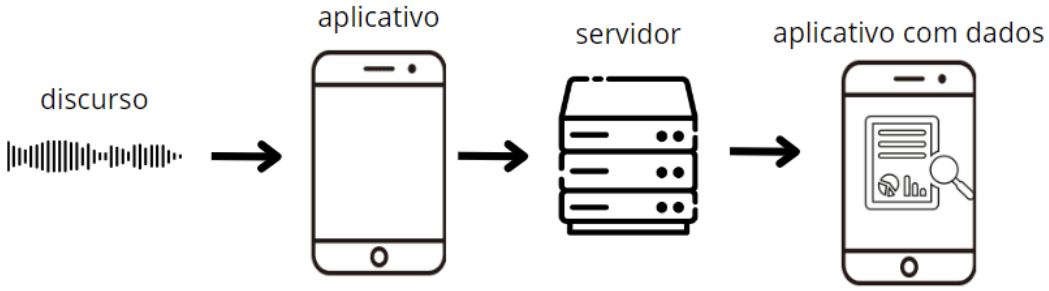
3.6. Desenvolvimento do algoritmo de análise de discurso

Antes do desenvolvimento das telas, foi necessário estabelecer quais dados conseguiríamos extrair do discurso com o *Assembly.AI*. Logo, tornou-se necessário

primeiro desenvolver os *scripts Python* do *backend* com *framework* Django utilizando as bibliotecas citadas anteriormente. Com um ambiente virtual criado com o Python 3.8.0, foi instalado as dependências das ferramentas no mesmo. Como também, realizou-se as configurações do servidor e do algoritmo de análise de discurso.

Para execução do algoritmo de análise de discurso (Figura 3), inicialmente, o áudio passa pelo *Assembly.AI*, onde se obtêm a transcrição da gravação junto com o tempo de cada palavra. Com estes dados, identificam-se características como palavras repetidas e as palavras estendidas.

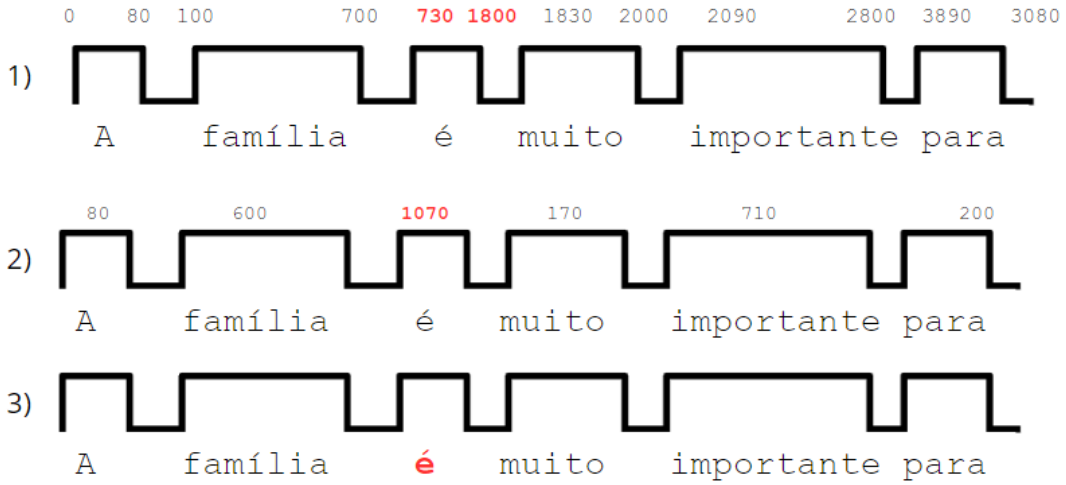
Figura 3. Fluxo de dados



Fonte: Elaborado pelo autor (2023).

Para identificação de palavras prolongadas, como mostrado na Figura 4, o algoritmo funciona da seguinte forma: inicialmente, com o retorno do *Assembly.AI* (1), é calculado quanto tempo cada palavra demorou para ser falada (2). Assumindo que uma palavra estendida, é considerada quando demora mais de 1 segundo para ser pronunciada, realizado o registro das palavras estendidas (3).

Figura 4. Funcionamento da identificação de palavras estendidas



Fonte: Elaborado pelo autor(2023).

Para validação das palavras mais repetidas no discurso, foi utilizado um módulo padrão do *Python* chamado *collections*, onde, enviando o texto da transcrição do áudio para a função *Counter* deste módulo, retorna com as cinco palavras mais repetidas no discurso.

Por fim, com o algoritmo de análise de discurso retorna um objeto com todos os dados anteriores citados acima em formato JSON. Como mostrado no Quadro 2, o objeto engloba informações do discurso como: palavras repetidas e palavras estendidas.

Quadro 2. Propriedades retornadas da análise de discurso da fala

Chave	Tipo	Descrição
palavras_estendidas	*array*	Matriz que armazena as palavras que demoraram para serem ditas
palavras_mais_repetidas	*array*	Vetor com as palavras mais repetidas do discurso

Fonte: Elaborado pelo autor (2023).

3.7. Desenvolvimento da interface

Com o registro deses dados, possibilitou-se o desenvolvimento da interface do aplicativo para demonstrar indicadores sobre o discurso dos usuários. Primeiro, foi desenvolvido o design responsivo do aplicativo utilizando o *Figma* baseando-se nos aplicativos ligados ao ensino da fluência do inglês, como *Stimuler*, e com o objetivo de obter uma referência de usabilidade.

3.8. Criação do aplicativo

Baseando-se na interface de usuário implementada na etapa anterior, foi desenvolvido o *frontend* do *React Native* com elementos nativos do *framework*. Para certas funcionalidades houve a necessidade da utilização de bibliotecas como expo-av, possuindo a funcionalidade principal do aplicativo, responsável por gravação e reprodução de áudio pelo microfone do celular. Axios, para comunicação de requisição HTTPS com o servidor, *react-native-reanimated-table*, para criar tabelas e *react-native-progress/Bar*, para barra de progresso.

Como mencionado anteriormente, a gravação de áudio é realizada pela biblioteca expo-av, onde, primeiro, é necessário validar se o aplicativo possui permissão para usar o microfone do dispositivo, para depois, configurar uma classe para realizar as gravações. Ao iniciar a gravação, somente interrompe-se a gravação e salva o áudio no celular após clicar no botão de parar. Por fim, com o áudio em mãos, usando a biblioteca axios, criamos um formulário de dados onde armazenamos a gravação e, assim, conseguimos enviar em uma requisição POST o arquivo de áudio para o servidor Django, onde é aplicado o algoritmo de análise de discurso.

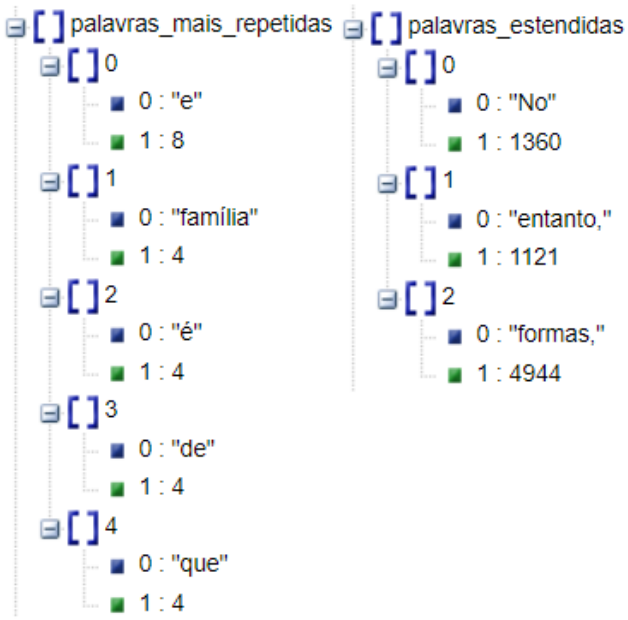
Após o servidor processar a gravação e retornar os dados de análise em JSON para o *app*, vai para a tela de análise da fala. Nessa tela, apresenta-se os dados coletados

da análise de algoritmo de falas, referente ao Quadro 2. A biblioteca *react-native-reanimated-table* foi empregada para apresentar as tabelas das palavras estendidas e palavras repetidas.

3.9. Validação do aplicativo

Com objetivo de validar o aplicativo e seu algoritmo, foi gravado o áudio de um discurso. Para testar a detecção das palavras estendidas, escolhemos algumas palavras em específico que, durante o discurso, demoramos para falar, são elas: “No”, "entanto" e "foram", como também, repetimos mais as palavras: 'família', 'e', 'é', ‘de’, ‘que’, para validar as palavras mais repetidas. O algoritmo teve o seguinte retorno em JSON como mostrado na Figura 5.

Figura 5. Representação do JSON



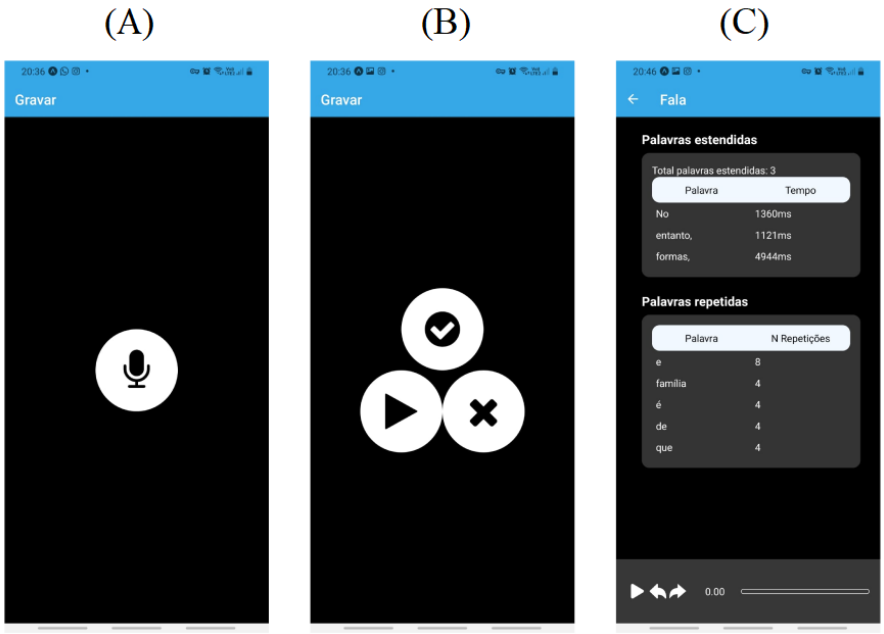
Fonte: Elaborado pelo autor(2023).

Comparando com os dados do áudio gravado, foi indicado um resultado positivo na identificação de palavras repetidas e palavras estendidas. Tendo apenas uma defasagem na contagem de palavras repetidas, devido ao fato do algoritmo ser *case-sensitive* e considerar pontuações. Através dos testes conseguimos comprovar que o aplicativo funcionou corretamente, podendo gravar o discurso, enviá-lo para análise e receber os dados em formato de tabelas no aplicativo, demonstrando precisão.

4. Resultados

Para validações e simulações, usando o *Expo Go* e, junto com o servidor Django rodando na máquina local via *prompt* de comando, realizou-se os testes no aplicativo de análise de oratória.

Figura 6. Telas do aplicativo



Nota: (a) tela de gravação (b) tela de gravação em andamento (c) tela de análise da fala

Fonte: Elaborado pelo autor (2023).

Na Figura 5, pode ser observado as telas do aplicativo. Na tela (A), mostra a tela inicial com o botão para iniciar a gravação do discurso. Ao clicar no botão de gravar, vai para a tela (B), onde aparecem os botões, de cima para baixo, para concluir, pausar e cancelar a gravação. Após encerrarmos a gravação do áudio, é apresentado tabelas, como mostrado na tela (C), que apresentam as palavras mais repetidas e as palavras estendidas no discurso.

5. Considerações Finais

Com a utilização de tecnologias inovadoras como inteligência artificial para analisar elementos da oratória, como repetição de palavras e locuções prolongadas, podem apoiar significativamente pessoas que desejam aprimorar a sua apresentação em público. O aplicativo apresentado no presente artigo se enquadra em um desses exemplos, logo, concluindo com sucesso o objetivo do trabalho.

Uma limitação presente foi a falta de testes em outros sistemas operacionais como iOS, sendo apenas testado no *Android*, como também, falta de testes com outros celulares de telas de tamanhos diferentes.

Ao longo deste estudo, através de métricas específicas, como tempo e ritmo das palavras, retornadas da tecnologia de reconhecimento de voz avançado *Assembly.AI*, foi possível a identificação de palavras estendidas e palavras repetidas no discurso. O projeto pode ser utilizado em vários segmentos da sociedade: educação, para ajudar alunos e professores a aperfeiçoarem suas habilidades orais, comercial, auxiliar em como persuadir clientes através da fala, ampliando o potencial de pesquisa em diferentes

setores.

Recomenda-se para trabalhos futuros adicionar novas funcionalidades para o aplicativo. Um dos exemplos seria a apresentação de um *feedback* mais claro para o usuário sobre como melhorar seu discurso nas telas de análise. Outro exemplo seria identificar outras características do discurso como ritmo de fala e palavras por minuto. Como também, poderia melhorar o algoritmo de identificação de palavras repetidas, ignorando pontuações ou palavras maiúsculas.

Referências

ARORA, Ashish. **The Importance of Public Speaking for Students**. 2021. Disponível em: https://sketchbubble.medium.com/the-importance-of-public-speaking-for-students-d9d06a8942ca. Acesso em: 4 nov. 2023.

CARSON, Clayborne; LEWIS, David Levering. **Martin Luther King, Jr**. 2023. Disponível em: https://www.britannica.com/biography/Martin-Luther-King-Jr/. Acesso em: 4 nov. 2023.

FROSSARD, Fabio. **Diferença entre Pesquisa Básica e Aplicada com Exemplos**. 2022. Disponível em: https://alunoexpert.com.br/pesquisa-basica-e-aplicada/. Acesso em: 10 nov. 2023.

MICHELSON, T.; PELEG, S. **Audio-Visual Evaluation of Oratory Skills**. Disponível em: <https://arxiv.org/abs/2110.01367>. Acesso em: 16 nov. 2023.

MUSHEKE, Mukelabai M.; PHIRI, Jackson. **The Effects of Effective Communication on Organizational Performance Based on the Systems Theory**. Open Journal Of Business And Management. Luzaka, p. 659-671. mar. 2021. Disponível em: https://doi.org/10.4236/ojbm.2021.92034. Acesso em: 11 nov. 2023.

NURMALASARI et al. **THE IMPACT OF SELF-CONFIDENCE ON STUDENTS' PUBLIC SPEAKING ABILITY**. 2023. Disponível em: https://journal.unm.ac.id/index.php/ijobec/article/view/70. Acesso em: 4 nov. 2023.

OCHOA, Xavier; DOMÍNGUEZ, Federico; GUAMÁN, Bruno; MAYA, Ricardo; FALCONES, Gabriel; CASTELLS, Jaime. The RAP System: Automatic Feedback of Oral Presentation Skills Using Multimodal Analysis and Low-Cost Sensors. In: LAK '18, 18., 2018, New York. **Proceedings of the 8th International Conference on Learning Analytics and Knowledge**. Sydney, New South Wales, Australia: Association For Computing Machinery, 2018. p. 360-364. Disponível em: https://dl.acm.org/doi/10.1145/3170358.3170406. Acesso em: 16 nov. 2023.

REDAÇÃO TRIBUNA ONLINE. **Medo de falar em público atinge 60% da população**. 2022. Disponível em: https://tribunaonline.com.br/jornal-reportagem-especial/medo-de-falar-em-publico-atinge-60-da-populacao-127915. Acesso em: 10 nov. 2023.

ROVEDA, Ugo. **O que é Django, para que serve e como usar este framework**. 2021. Disponível em: https://kenzie.com.br/blog/django/. Acesso em: 11 nov. 2023.