

# **Desenvolvimento de um sistema de leitura e reprodução de textos em PDF**

**Wallace Luiz Ferreira da Conceição, Ricardo Pires Mesquita**

Departamento de Ciência da Computação – Centro Universitário Carioca  
(Unicarioca)

**Abstract.** *The article in question involves the creation of a computer system that automatically extracts and converts text from documents into PDF format for audio reproduction. This system aims to make document content more accessible, especially for people with visual impairments or some reading difficulties. Combining natural language processing (NLP) and speech synthesis techniques, the system allows users to listen to the content of documents, improving accessibility and enabling digital inclusion. Furthermore, the development of this system can contribute to areas such as assistive technologies, inclusive education and the use of computational tools that promote equal access to information.*

**Resumo.** O artigo em questão envolve a criação de um sistema computacional que extrai e converte automaticamente o texto de documentos em formato PDF para reprodução em áudio. Esse sistema tem como objetivo tornar conteúdos de documentos mais acessíveis, especialmente para pessoas com deficiência visual ou alguma dificuldade de leitura. Combinando técnicas de processamento de linguagem natural (PLN) e síntese de voz (TTS), o sistema possibilita que os usuários ouçam o conteúdo de documentos, melhorando a acessibilidade e permitindo a inclusão digital. Além disso, o desenvolvimento desse sistema pode contribuir para áreas como tecnologias assistivas, educação inclusiva e uso de ferramentas computacionais que promovem a igualdade de acesso à informação.

## 1. Introdução

Nos últimos anos, a busca por acessibilidade e inclusão se tornou uma prioridade para a sociedade. Onde 85% das organizações do mundo que participaram da pesquisa global de diversidade, equidade e inclusão declararam que tratam a DE&I (Diversidade, equidade e inclusão) como um valor ou prioridade [PWC 2022]. O acesso à informação e à comunicação é um direito fundamental [LBI 2015], porém, diversas vezes, as barreiras tecnológicas dificultam o alcance desse objetivo para determinado público.

Este artigo aborda o desenvolvimento de um sistema computacional projetado para resolver um problema real que afeta milhões de pessoas: a barreira do acesso a textos de documentos em formato *PDF* por deficientes visuais ou com alguma dificuldade de leitura.

A tecnologia está evoluindo rapidamente e conseguindo promover mais igualdade de acesso à informação, e o sistema apresentado neste artigo representa uma pequena contribuição para essa missão. Tal processo foi possível através de combinações de técnicas de processamento de linguagem (PLN) e síntese de voz (TTS), deixando aberto um potencial de melhorar ainda mais a forma como as pessoas interagem com arquivos em *PDF*. Tornando conteúdos de documentos mais acessíveis para pessoas com deficiência visual, e estendendo também para pessoas com transtornos neurológicos, como dislexia, ampliando seu alcance e impacto.

Será abordada a motivação por trás do desenvolvimento desse sistema, apresentando dados que exteriorizam a urgente necessidade de soluções de acessibilidade. Além disso, este artigo fornece detalhes da arquitetura, do fluxo de trabalho, dos desafios enfrentados e das soluções que foram feitas durante o desenvolvimento. Por fim, ele apresenta os resultados de uma pesquisa feita com um pequeno grupo de pessoas, que destaca a eficácia do sistema e recomendações de melhorias futuras.

O objetivo deste artigo é dar uma clara compreensão de como o sistema abordado pode fazer a diferença na vida de pessoas, que enfrentam, além dos desafios normais de uma pessoa com deficiência visual ou algum distúrbio neurológico, também as dificuldades para acessar um simples texto de um documento em *PDF*.

## 2. Motivação

A acessibilidade, há alguns anos, vem se tornando crucial na busca por igualdade e inclusão. Pessoas com deficiência visual ou alguma dificuldade de leitura enfrentam desafios significativos ao tentar acessar informações contidas em documentos em *PDF*, que muitas vezes, não são compatíveis com leitores de tela [Passos, Vieira e Saheki 2008] e outros dispositivos de auxílio. Isso causa uma exclusão involuntária de uma parcela significativa da população de recursos importantes, como materiais educacionais, documentos governamentais e conteúdo *online*, pois segundo Noleto (2022), o formato *PDF* fez muito sucesso e na atualidade é visto como um dos padrões mais necessários.

Com base nos resultados da pesquisa nacional de saúde (PNS) mais recente de 2019, informa que aproximadamente 3,4% da população brasileira (o que equivale a cerca de 6,978 milhões de indivíduos), é afetada por algum tipo de problema visual [IBGE 2021]. Contudo, quando consideramos a importância da inclusão, é essencial abranger o máximo de pessoas possíveis para garantir que elas se beneficiem de

recursos que facilitem o acesso à informação. Isso é particularmente relevante, uma vez que a porcentagem de pessoas que podem se favorecer dessas soluções aumenta significativamente quando levamos em conta outros desafios relacionados à leitura.

Além das dificuldades visuais, devemos considerar também as pessoas que enfrentam obstáculos devido a transtornos neurológicos, como a dislexia. De acordo com a *International Dyslexia Association (IDA)*, estima-se que esse transtorno afeta de 10 a 15% da população global. Este é um Transtorno Específico de Aprendizagem, de origem neurológica, caracterizada pela dificuldade de decifrar a escrita ou símbolos, sendo um transtorno que dificulta a habilidade de leitura, escrita e, por consequência, a linguagem [ABD 2022].

Portanto, ao desenvolver soluções que auxiliam na leitura e no acesso a conteúdos, devemos ter em mente não apenas as questões de deficiência visual, mas também as necessidades de pessoas com diferentes desafios, promovendo uma verdadeira inclusão. Isso significa que o alcance e o impacto de tais recursos se estendem a um grupo muito mais amplo, tornando-os úteis para estudantes, profissionais e indivíduos que buscam informações de maneira mais acessível e eficaz em diversos cenários, como durante o trabalho ou estudos na faculdade.

### 3. Trabalhos Relacionados

Existem diversas opções gratuitas ou pagas disponíveis para facilitar a acessibilidade de documentos em *PDF* e outros formatos, tornando-os mais amigáveis para indivíduos com deficiência visual ou dificuldades na leitura. Algumas destas alternativas oferecem funcionalidades abrangentes, incluindo suporte para múltiplos idiomas e narrações assistidas por inteligência artificial.

#### 3.1 Alternativas

- **NaturalReader** – Este, talvez, seja o sistema mais completo do mercado atualmente com essa finalidade. Disponível para *web*, *Android*, *iOS* e como extensão do *Google Chrome*, possui suporte para arquivos *PDF* e mais de 20 formatos, 16 idiomas, mais de 200 vozes geradas por IA, tecnologia assistida para usuários disléxicos entre outras funcionalidades que o torna extremamente versátil.
- **Adobe Acrobat Reader DC** – Este é o leitor de *PDF* mais popular em todo o mundo. Ele está disponível gratuitamente para *Windows*, *macOS*, *iOS* e *Android*. Além das funcionalidades padrões de um leitor de *PDF*, como visualizar, criar e compartilhar *PDFs*, ele também possui no seu menu de exibição a opção "Ler em voz alta" que realiza a leitura do documento.
- **ReadSpeaker speechMaker** – Uma alternativa um pouco diferentes das apontadas acima, *speechMaker* é mais voltado ao meio comercial, para tornar produtos mais acessíveis, e auxiliar na produção de conteúdo. Basicamente, ele converte um texto escrito de forma manual (ou copiado e colado) em áudio utilizando a tecnologia *text-to-speech* de forma rápida e simples, e gera um arquivo de áudio que possui suporte atualmente para 12 idiomas.
- **@Voice Aloud Reader** – Simples e focado exclusivamente para *Android*, este lê em voz alta o texto exibido em um aplicativo, como páginas *web*, *e-mails*, *PDFs* entre outros. Também utiliza o mecanismo *text-to-speech* de terceiros, instalados

no celular, para poder funcionar. Concede a liberdade ao usuário de usar o mecanismo da *Google*, que normalmente já vem pré-instalado nos celulares *Android*, ou baixar um de sua preferência.

Foi mencionado acima algumas opções de destaque que fornecem a mesma funcionalidade que estamos procurando, embora com abordagens e objetivos distintos. Além dessas, há muitas outras alternativas que disponibilizam esse objetivo, oferecendo uma ampla variedade de escolhas para o usuário, permitindo-o selecionar a que mais lhe agradar.

## 4. Proposta de Solução

Nesta seção, será descrito a implementação do sistema. Abordando sua arquitetura e tecnologias utilizadas, (com breve comentário de seus pontos fortes que as tornaram escolhas), o fluxo de trabalho, a *interface* do usuário, o processamento de texto e a conversão de áudio, o gerenciamento de *threads*. Também será comentado os desafios enfrentados e suas soluções.

### 4.1 Arquitetura do Sistema

A arquitetura do sistema é composta por componentes interconectados que se comunicam para permitir que os usuários convertam arquivos *PDF* em áudio. Os principais componentes são:

**Interface do usuário:** A *interface* onde os usuários podem carregar os arquivos *PDF* e reproduzir o áudio, criada a partir da biblioteca do *tkinter* [Tkinter 2023]. Esta é uma biblioteca do *Python*, conhecida por possuir milhares de recursos que permitem criar *interfaces* gráficas das mais simples a mais complexas. Tem uma vasta documentação e o pacote básico do *Python* já o carrega [Labaki 2006].

**Processamento de texto:** Este recurso extrai o texto do *PDF*. Para isso, foi utilizada a biblioteca *Python PyPDF2* [PyPDF2 2022], é uma biblioteca de código aberto que, de acordo com Ramos (2023), é uma ferramenta poderosa e fácil de usar para o trabalho com arquivos *PDF* em *Python*.

**Conversão do texto para fala:** Tal procedimento envolve a sintetização do texto extraído do *PDF* para fala. Foi feito utilizando a biblioteca *gTTS* (*Google Text-to-Speech*) [GTTS 2023] que permite converter texto em fala através do serviço *Text-to-Speech* da *Google* e pode ser salvo como arquivo *MP3*. Também permite personalizar velocidade e idioma da fala, pois tem suporte a vários idiomas [Goel 2023].

**Reprodução de áudio:** A reprodução do áudio gerado é feita com o auxílio da biblioteca *Pygame* [Pygame 2023]. Tal biblioteca é voltada para desenvolvimento de *games* sob a linguagem *Python*, porém, nesse protótipo, foi utilizada, em particular, para reprodução do arquivo *MP3* gerado a partir do texto extraído do arquivo *PDF*. Segundo Ribeiro (2023), é uma biblioteca muito versátil na sua utilização, mas uma implementação simples é utilizá-la para tocar arquivos no formato *MP3*.

**Gerenciamento de *Threads*:** É utilizado o gerenciamento de *threads* para processar o *PDF* em segundo plano, garantindo a responsividade da *interface* do usuário, mesmo em operações demoradas como essa. Esse procedimento é feito com o auxílio da biblioteca *threading* [Threading 2023]. Em conformidade com Henrique (2023), é

especialmente importante em aplicativos que envolvem operações de entrada e saída intensivas, como processamento de arquivos. Sem o *threading*, o programa seria bloqueado enquanto espera a conclusão de uma tarefa de entrada ou saída.

As tecnologias utilizadas foram escolhidas com base em sua eficácia e facilidade de uso, que atenderiam o objetivo principal do projeto; tendo isso em vista, *Python* [Python 2023] é a linguagem de programação principal do mesmo. Além da importância de aprender *Python* ainda estando na faculdade, de acordo com *Van Rossum* (2005), *Python* é simples de usar, sem deixar de ser uma linguagem de programação de verdade, oferecendo muito mais estruturação e suporte para programas extensos do que *shell scripts* oferecem. Por outro lado, *Python* também oferece melhor verificação de erros do que C, e por ser uma linguagem de alto nível, ela possui tipos nativos de alto nível: dicionários e vetores(*arrays*) flexíveis que lhe custariam dias para obter uma implementação eficiente em C. Devido ao suporte nativo a tipos genéricos, *Python* é aplicável a um domínio de problemas muito mais vasto do que *Awk* ou até mesmo *Perl*, ainda assim *Python* é tão fácil de usar quanto essas linguagens sob diversos aspectos.

## 4.2 Fluxo de Trabalho

O fluxo de trabalho do sistema é bem simples, intuitivo e compreensível para o usuário. Começa quando o utilizador por meio da *interface* do usuário seleciona um arquivo *PDF*. Em seguida o sistema extrai o texto do arquivo, converte o texto em áudio, e finalmente, habilita o botão para reproduzir o áudio gerado. Para melhor ilustração do informado, visualize o fluxograma abaixo:

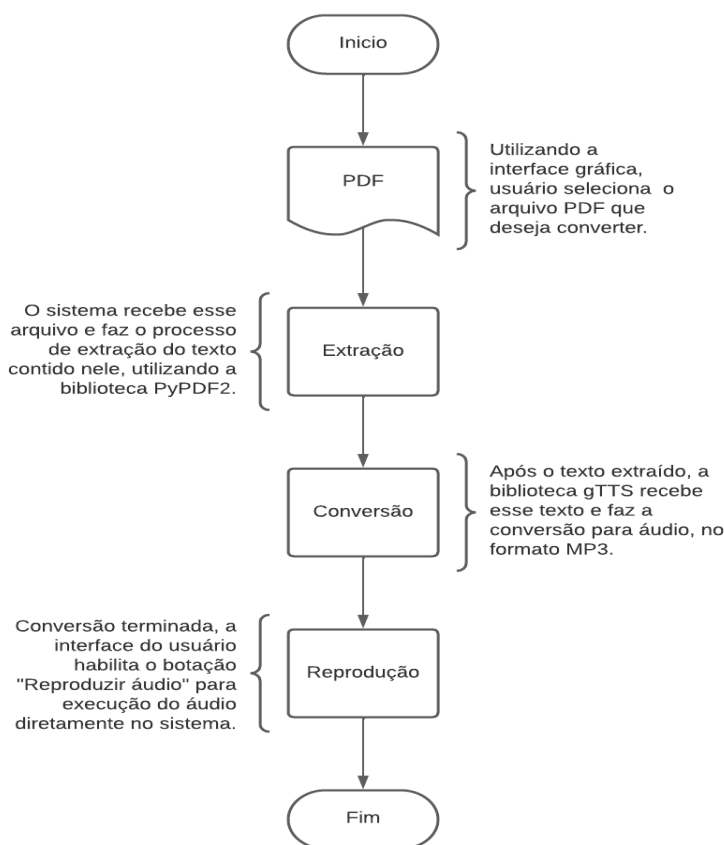


Figura 1: Fluxograma do sistema

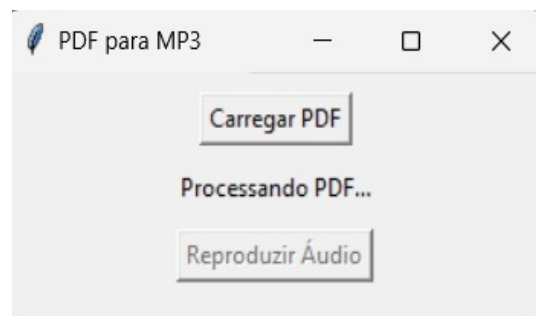
### 4.3 Interface do Usuário

A *interface* do usuário foi projetada pensando em simplicidade. Permitindo que os usuários carreguem facilmente arquivos *PDF*, visualizem informações sobre o áudio gerado e o reproduzam quando desejarem. Como mostram as imagens abaixo:



**Figura 2:** Interface do usuário na sua tela inicial

- Em sua tela inicial, a *interface* habilita apenas o botão de “Carregar *PDF*”, para que o usuário escolha qual documento *PDF* que deseja processar.



**Figura 3:** Interface do usuário durante o processamento do *PDF*

- Após a seleção do documento *PDF* desejado, a *interface* mostra ao usuário que o documento está sendo processado.



**Figura 4:** Interface do usuário após gerar o áudio

- Ao finalizar o processamento, é exibido ao usuário que o arquivo *MP3* foi gerado; indicando o nome do arquivo e habilitando o botão “Reproduzir Áudio”.



**Figura 5: Interface do usuário durante a reprodução do áudio**

- Ao clicar no botão “Reproduzir Áudio”, o sistema inicia a reprodução do arquivo *MP3* que foi gerado. A *interface* informa que o áudio está sendo reproduzido e exibe o nome do arquivo.

#### **4.4 Extração de Texto e Conversão de Áudio**

Esse processo envolve basicamente a extração do texto do *PDF* e a aplicação de melhorias, como alterar linguagem, velocidade da pronúncia e a localização do servidor que o *gTTS* utilizará para melhorar a qualidade da fala.

Para manter a responsividade da *interface* do usuário durante esse procedimento, foi utilizado *threads* para processar o *PDF* em segundo plano. Deixando o usuário à vontade para manipular a janela como quiser.

#### **4.5 Desafios e Soluções**

Os desafios foram significativos durante a implementação, como tornar a qualidade da fala a melhor possível; tendo em vista algumas limitações da ferramenta utilizada e garantir a responsividade da *interface* do usuário. As soluções envolveram o uso de configurações específicas de idioma, velocidade de fala e região durante a implementação da biblioteca responsável pela síntese de voz e a implementação de *threads* para o processamento do *PDF* em segundo plano.

### **5. Análise de Resultados**

O processo de análise de resultados foi baseado na disponibilização do protótipo para um grupo de pessoas de diferentes perfis, e realizado simulações com dois documentos de texto em *PDF* pré determinados, e um de escolha do participante. Em seguida foi fornecido uma pesquisa no *Google Forms* com perguntas diversas sobre expectativas, funcionalidade e satisfação.

#### **5.1 Perfil dos Participantes**

A faixa etária dos participantes que responderam a pesquisa variou entre 20 e 55 anos, com idade média de 36 anos. Desses, 58% são do sexo masculino e 42% do sexo feminino. Dos perfis, 50% possuem algum diploma de graduação.

## 5.2 Avaliação Geral do Protótipo

Foram obtidos os seguintes dados durante a realização da pesquisa:

- 100% dos participantes disseram que o protótipo atendeu as suas expectativas;
- 100% dos participantes classificaram a *interface* do usuário como intuitiva;
- 83,3% dos participantes avaliaram a qualidade da saída de áudio gerada a partir do *PDF* como MUITO BOM; 8,3% como BOM e 8,3% como NEUTRO;
- 100% dos participantes relataram não ter ocorrido nenhum problema durante a reprodução do áudio;
- 91,7% dos participantes relataram que o protótipo conseguiu recitar corretamente os textos e 8,3% relataram que pronunciou corretamente, mas com algumas dificuldades ocasionais, como ilustrado abaixo:



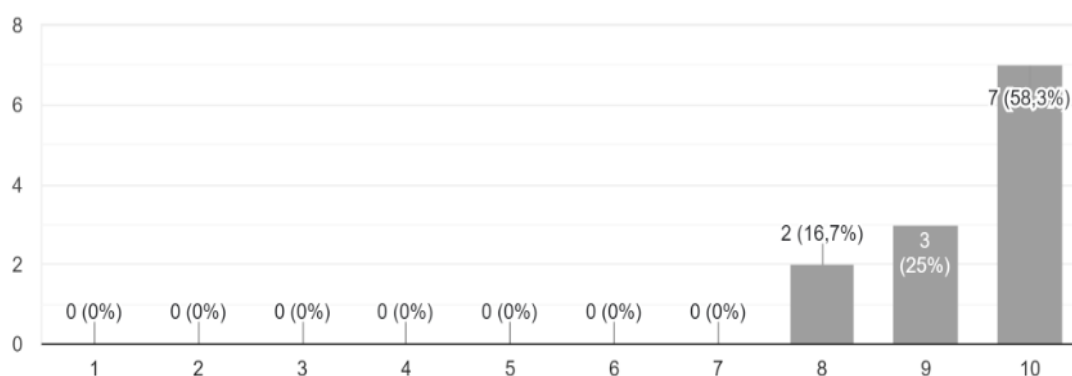
Figura 6: Gráficos das respostas da pesquisa



- Por fim, como ilustrado na escala de satisfação geral abaixo, onde 0 é insatisfeito e 10 é muito satisfeito, 58,3% dos participantes declarou como nota 10 de muito satisfeito, 25% como nota 9 e 16,7% como nota 8.

Em uma escala de 1 a 10, o quão satisfeito você está com o sistema?

12 respostas



**Figura 7: Gráfico de escala de satisfação geral**

### 5.3 Problemas Encontrados

Durante a realização do projeto e da pesquisa, foi relatado por um dos participantes uma variação na entonação e pronunciamento da palavra “acessíveis”. Por ser uma palavra de classificação paroxítona, em vez de pronunciar sua acentuação normal, a biblioteca a pronuncia como “acessivéis”.

Foram testadas outras palavras da mesma classificação de acentuação e as mesmas foram lidas sem demais problemas, sendo um caso específico com elemento em questão. Provavelmente proveniente da versão da biblioteca *gTTS* utilizada, que pode ainda não estar atualizada com a pronúncia correta.

### Interpretação dos Resultados

Os resultados da pesquisa sugerem que o protótipo do sistema de leitura e reprodução de textos em *PDF* foi muito bem recebido pelos participantes. A maioria avaliou positivamente a sua usabilidade, e expressou satisfação com o resultado que o sistema entrega. Isso indica que o protótipo alcançou seu objetivo de fornecer uma experiência agradável a quem procura facilidade, ou mais acessibilidade ao lidar com documentos de textos em *PDF*.

A principal recomendação de melhoria apontada pelos participantes foi na pronúncia de algumas palavras, mais especificamente na palavra “acessíveis”. Tal informação é valiosa para o desenvolvimento futuro do sistema.

### Conclusão e Recomendações dos Resultados

Com base no resultado e em todo processo envolvendo a pesquisa, foi verificado durante o contato presencial, (mesmo os participantes não relatando no *Google Forms*) uma lentidão ocasional no momento da extração de alguns textos, e até mesmo

travamento e mau funcionamento ao tentar realizar o processo com documentos com volumoso número de páginas. Além de claro, como relatado anteriormente, a dificuldade na pronúncia de uma palavra específica. Com isso, é recomendável:

- A implementação de medidas de otimização no código para melhorar o desempenho da aplicação no momento da extração e tratamento do texto. Pois o processo de extração de texto no código do protótipo é feito através de um *looping* utilizando “*for*” para extrair página por página do arquivo *PDF*. Porém, o consenso é que muito *looping* em qualquer linguagem de programação não é uma coisa boa, o que coloca pressão desnecessária sobre o seu servidor [Eck 2015].
- Verificação e análise de outras bibliotecas que convertam os textos extraídos em falas, tendo como o objetivo um suporte melhor para o idioma português brasileiro. Pois com o *gTTS* a conversão de texto em fala do *Google Tradutor* pode falar em diferentes “sotaques” locais, dependendo do domínio do *Google*, conforme descrito na documentação da biblioteca [GTTS 2023]. Entretanto, mesmo no código do sistema com o padrão em português brasileiro, algumas palavras ficaram com pronúncias diferentes.

Tais melhorias podem contribuir para uma experiência ainda melhor do usuário e aumentar a satisfação e a adoção do sistema.

## 6. Conclusão

Por fim, o protótipo desenvolvido neste artigo para permitir a leitura e reprodução de textos em *PDF*, é um pequeno esforço para aumentar a acessibilidade e a inclusão digital.

Os resultados da pesquisa com o grupo de pessoas, mostram que o protótipo foi bem recebido pelos participantes e é eficaz. Considerando a grande porcentagem da população que possui problemas para acessar documentos em *PDF*; soluções como essa se tornam extremamente importantes.

Entretanto, como relatado durante todo o artigo, este sistema é apenas um protótipo. Tendo em vista o curtíssimo período de tempo disposto para desenvolvê-lo, não é o fim da viagem. A otimização do desempenho da aplicação e da pronúncia de palavras específicas são problemas visíveis.

Este artigo deixa explícito que a tecnologia tem uma grande força de inclusão, ajudando àqueles que enfrentam dificuldades a superá-las e garantindo que todos possam ter acesso igual à informação. Espera-se que esse trabalho inspire ainda mais desenvolvedores, pesquisadores e a sociedade como um todo a continuar em busca de soluções tecnológicas que tornem, (na medida do possível) a jornada mais acessível e justa para todos.

### 6.1 Trabalhos Futuros

Como proposta de melhorias em trabalhos futuros, é importante citar a necessidade de atingir o maior número de pessoas, da melhor maneira possível. Com isso, podemos apontar alguns pontos necessários para tal objetivo:

- Otimização do desempenho para arquivos com número demasiado de páginas: Possivelmente, o problema que mais incomodou aos usuários no momento dos testes. Tal processo pode ser testado através da troca do “for” no momento da extração do texto[Eck 2015], ou talvez através de otimizadores estáticos para a linguagem *Python*, pois de acordo com Cabral (2013) é uma linguagem de programação de propósito geral, de alto nível, com tipagem dinâmica e que suporta múltiplos paradigmas de programação. Todavia, se dependesse do desempenho do seu principal interpretador (*Cpython*) não teria tanto sucesso, e como resultado dos seus teste com o otimizador utilizado por ele, conseguiu uma melhora de até 60 vezes no desempenho.
- Inclusão de novas tecnologias que melhorem a experiência do usuário: A biblioteca de síntese de voz utilizada (*gTTS*) não funciona caso o usuário não tenha acesso à internet, já que a mesma se comunica com os servidores da *Google* para acessar os recursos do *Text-to-speech*. Para garantir a inclusão social, que de acordo com Bessa (2019), é o conjunto de medidas direcionadas a indivíduos excluídos do meio social, seja por alguma deficiência física ou mental, cor da pele, orientação sexual, gênero ou poder aquisitivo dentro da comunidade; a utilização de recursos que não necessitem de acesso à internet para funcionar, seria o ideal para cumprir o objetivo buscado por esse projeto. Nesse caso, uma opção seria utilizar a biblioteca *Pyttsx3*, que realiza conversões e funciona sem conexão à internet[Narula 2023].
- Tornar a aplicação multiplataforma: O protótipo em questão, atualmente, funciona apenas no ambiente de trabalho onde foi programado. Seria de extrema importância estender essa aplicação para outras plataformas, como *web* e *mobile*, para atender o máximo de pessoas possíveis.

## 7. Referências

- PWC (2022) “Diversity, Equity & Inclusion Benchmarking Survey”  
<https://www.pwc.com/gx/en/services/people-organisation/global-diversity-and-inclusion-survey/global-report-2022.pdf>, Novembro.
- LBi (2015) “Lei brasileira de inclusão de pessoas com deficiência”  
[https://www.cnmp.mp.br/portal/images/lei\\_brasileira\\_inclusao\\_pessoa\\_deficiencia.pdf](https://www.cnmp.mp.br/portal/images/lei_brasileira_inclusao_pessoa_deficiencia.pdf), Novembro.
- Passos, J., Vieira, R. e Saheki, Y. (2008) “LEITORES DE TELAS: ferramenta de Documentos acessíveis”,  
[http://repositorio.febab.libertar.org/files/original/30/4253/SNBU2008\\_102.pdf](http://repositorio.febab.libertar.org/files/original/30/4253/SNBU2008_102.pdf), Novembro.
- Noletto, C. (2022) “O que é extensão de arquivo e quais as mais usadas?”  
<https://blog.betrybe.com/tecnologia/extensao-de-arquivo/>, Novembro.
- ABD (2022) “Outubro: o mês da conscientização da dislexia”  
<https://www.dislexia.org.br/outubro-o-mes-da-conscientizacao-da-dislexia/>, Outubro.

IBGE (2021) “PNS 2019: país tem 17,3 milhões de pessoas com algum tipo de Deficiência”, <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/31445-pns-2019-pais-tem-17-3-milhoes-de-pessoas-com-algum-tipo-de-deficiencia>, Outubro.

NaturalReader “*The #1 text to speech solution for personal, commercial, and Educational use*”, <https://www.naturalreaders.com>, Outubro.

Adobe “O visualizador de PDF gratuito mais confiável do mundo” <https://get.adobe.com/br/reader/>, Outubro.

ReadSpeaker “Converta o seu texto escrito em áudio de alta qualidade para que possa Ser usado em várias aplicações”, <https://www.readspeaker.com/pt/solutions/producao-de-voz/readspeaker-speechmaker/>, Outubro.

Hyperionics “*@Voice Aloud Reader reads aloud the text displayed in an Android App, e.g. web pages, news articles, long emails, sms, PDF files and more.*”, <https://hyperionics.com/atVoice/>, Outubro.

Labaki, J. (2006) “Introdução a Python – Modulo C, Tkinter” <https://www.dcc.ufrj.br/~fabiom/mab225/tutorialtkinter.pdf>, Setembro.

Tkinter (2023) “*Interfaces Gráficas de Usuário com Tk*” <https://docs.python.org/pt-br/3/library/tk.html>, Setembro.

Ramos, V. (2023) “Manipulando arquivos PDF com *Python*” <https://pythonacademy.com.br/blog/manipulando-arquivos-pdf-com-python>, Novembro.

PyPDF2 (2022) “*Welcome to PyPDF2*” <https://pypdf2.readthedocs.io/en/3.0.0/>, Setembro.

GTTS (2023) “*gTTS*” <https://gtts.readthedocs.io/en/latest/>, Setembro.

Goel, A. (2023) “*Convert Text to Speech in Python*” [https://www.geeksforgeeks.org/convert-text-speech-python/?ref=header\\_search](https://www.geeksforgeeks.org/convert-text-speech-python/?ref=header_search), Setembro.

Ribeiro, A. (2023) “Tocar mp3 com a biblioteca *pygame*” <https://py013.com.br/2023/06/05/tocar-mp3-com-a-biblioteca-pygame/#:~:text=A%20biblioteca%20pygame%20é%20muito,no%20seu%20sistema%20ou%20game.>, Setembro.

Pygame (2023) “*pygame module for loading and playing sounds*” <https://www.pygame.org/docs/ref/mixer.html>, Setembro.

Henrique, P. (2023) “THREADING NO PYTHON”

<https://pydetodos.com/threading-no-python/#:~:text=A%20biblioteca%20threading%20em%20Python,outras%20tarefas%20em%20segundo%20plano.>, Setembro.

Threading (2023) “*threading — Thread-based parallelism*”

<https://docs.python.org/3/library/threading.html>, Setembro.

Van Rossum, G. (2005) “Tutorial Python Release 2.4.2”

<https://www.dcc.ufjf.br/~fabiom/mab225/tutorialpython.pdf>, Outubro.

Python (2023) “*Python 3.12.0 documentation*”

<https://docs.python.org/3/>, Setembro.

Eck, R. (2015) “6 formas de melhorar o seu desempenho em *Python*”

<https://imasters.com.br/back-end/6-formas-de-melhorar-o-seu-desempenho-em-python>, Novembro.

Cabral, G. T (2013) “Um otimizador estático para a linguagem *Python*”

Trabalho de Conclusão – Universidade Federal do Rio Grande do Sul. Curso de Bacharelado em Ciência da Computação, Porto Alegre, BR-RS, 2013. Orientador: Marcelo Johann.

Bessa, L. (2019) “O que é inclusão social?”

<https://www.politize.com.br/inclusao-social/#:~:text=A%20inclusão%20social%20é%20o,poder%20aquisitivo%20dentro%20da%20comunidade.>, Novembro.

Narula, M (2023) “Converter texto em fala em *Python*”

<https://www.delftstack.com/pt/howto/python/python-text-to-speech/>, Novembro.