

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO TRIÂNGULO MINEIRO**

RANDER FELIPE MIRANDA DE PAULA

**TRADUÇÃO AUTOMATIZADA: A INTERSEÇÃO DA LINGUÍSTICA E
DA INTELIGÊNCIA ARTIFICIAL**

UBERABA

2023

RANDER FELIPE MIRANDA DE PAULA

**TRADUÇÃO AUTOMATIZADA: A INTERSEÇÃO DA LINGUÍSTICA E
DA INTELIGÊNCIA ARTIFICIAL**

Trabalho de conclusão de curso
apresentado ao Instituto Federal de
Educação, Ciência e Tecnologia do
Triângulo Mineiro, Campus Avançado
Uberaba Parque Tecnológico, como
requisito parcial para conclusão do Curso de
Engenharia de Computação.

Orientador: Prof. Camilo de Lelis Tosta
Paula

UBERABA - MG

2023

AGRADECIMENTOS

Aos meus familiares, sou extremamente grato por todo apoio oferecido durante estes longos seis anos, nunca me deixaram faltar nada, seja financeiramente ou emocionalmente, nunca mediram esforços para que eu pudesse seguir meu objetivo. Sou muito grato também por entenderem todos os momentos de ausência. Aos mais novos: meus irmãos, primos e afilhados, nunca deixem de sonhar e estudar, todos os seus sonhos serão alcançados.

Ao Instituto Federal do Triângulo Mineiro e seus servidores, sou grato pela oportunidade e excelência proporcionada, a educação é um direito de todos e o papel de instituições públicas como o IFTM são de extrema importância nessa disseminação de conhecimento. Ao corpo docente, além de meu agradecimento, deixo minha admiração pela didática e empenho mesmo em um momento tão turbulento quanto foi a pandemia, espero que sempre mantenham viva dentro de vocês esse entusiasmo em ensinar.

Ao meu professor orientador Prof. Camilo de Lelis Tosta Paula, agradeço por todo o acompanhamento durante o processo de desenvolvimento deste trabalho, seus conselhos e orientações deixaram tudo mais claro e divertido.

Por último, e não menos importante, aos meus amigos, muito obrigado por todos momentos, sem vocês o processo não seria tão leve e descontraído. Foram muitos momentos de descontração e ajuda mútua que se tornaram aprendizado.

RESUMO

Nos dias atuais, é indispensável o uso de arquivos digitais criados em formato de PDF (*Portable Document Format*) textuais, como artigos acadêmicos, livros no geral e até quadrinhos nesse formato, devido a praticidade de armazenamento e divulgação dos mesmos. Devido a globalização, o mais provável é encontrar a maioria desse conteúdo em inglês, necessitando assim, a maioria dos brasileiros, uma tradução para o português, algo para suprir esse problema. Assim surgiu a proposta desse projeto, que busca o desenvolvimento de um software inicialmente apenas para Desktop de tradução desses textos e quadrinhos para o português, onde o usuário irá inserir o arquivo original, que por sua vez terá todo seu conteúdo extraído por uma inteligência artificial, que fará o reconhecimento do que é texto e do que é imagem dentro do arquivo e, através de uma segunda rede neural atrelada à um banco de dados local, irá realizar a tradução de bidirecional entre português e inglês, retornando então em um arquivo final, com o conteúdo do idioma oposto ao que foi inserido inicialmente, de forma totalmente automatizada. O desenvolvimento desse software utilizará como linguagem principal o Python, com assistência de diversas bibliotecas auxiliares, tanto para o manuseio dos arquivos, quanto para realizar a tradução dos conteúdos.

Palavras-chave: arquivos digitais; tradução; bibliotecas Python

ABSTRACT

In today's world, the use of digital files created in the PDF (Portable Document Format) textual format is indispensable, encompassing academic articles, books, and even comics, owing to the convenience of storage and dissemination. Due to globalization, the majority of this content is likely to be in English, necessitating a translation into Portuguese for the majority of Brazilians to address this issue. Hence, the proposal for this project emerged, aiming to develop software initially for Desktop that translates these texts and comics into Portuguese. In this process, the user will input the original file, and an artificial intelligence will extract its entire content, distinguishing between text and images within the file. Through a second neural network connected to a local database, bidirectional translation between Portuguese and English will be performed, resulting in a final file with content in the opposite language from the one initially inserted, all in a fully automated manner. The development of this software will utilize Python as the main programming language, supported by various auxiliary libraries for file handling and content translation.

Keywords: digital files; translation; Python libraries

LISTA DE FIGURAS

Figura 1: Ícone de aplicação do Adobe Acrobat	16
Figura 2: Amostra prática de escolha de formato de arquivo em PDF	17
Figura 3: Diferenças entre Software e Hardware	19
Figura 4: Logo Python	23
Figura 5: Representação de um neurônio biológico	26
Figura 6: Representação do neurônio artificial	28
Figura 7: SMT x NMT	31
Figura 8: Diagrama de funcionamento do projeto de tradução de PDFs	33
Figura 9: Comando de instalação da Biblioteca Python – PyPDF2	34
Figura 10: Códigos relacionados à biblioteca PyPDF2	35
Figura 11: Comando para instalação da biblioteca de Python de tradução ...	36
Figura 12: Códigos relacionados à biblioteca de tradução	36
Figura 13: Código de varredura de conteúdo	38
Figura 14: Código de tradução do conteúdo	39
Figura 15: Alocação do conteúdo de texto traduzido em um arquivo novo ..	39
Figura 17: Código de conversão de tipo de arquivo TXT para PDF	40
Figura 18: Arquivo “pdf5.pdf” apresentado dentro do diretório do projeto ..	41
Figura 19: Apresentação do conteúdo em inglês presente em “pdf5.pdf”	42
Figura 20: Arquivo “txt” apresentado dentro do diretório do projeto	42
Figura 21: Conteúdo do documento que foi executado no projeto	43
Figura 22: Output.pdf dentro do diretório do projeto	43
Figura 23: Arquivo com o conteúdo traduzido do original	44

LISTA DE SIGLAS E ABREVIACES

APIs	Application Programming Interface
AWS	Amazon Web Services
CD	Compact Disc
IDE	Integrated Development Environment
ISO	International Organization for Standardization
NMT	Neural Machine Translator
PDF	Portable Document Format
RNA	Rede Neural Artificial
SMT	Statistic Machine Translator
TXT	Arquivo de Texto

SUMÁRIO

1 INTRODUÇÃO	10
2 REFERENCIAL TEÓRICO	16
2.1 Informações digitais	16
2.2 Software	20
2.3 Linguagem de Programação	22
2.4 Python	23
2.5 Aprendizado de Máquinas - Machine Learning	25
2.6 Redes Neurais	26
2.7 Sistemas de tradução automática	29
3 MATERIAIS E MÉTODOS	33
3.1 Usuário Escolhe Arquivo	34
3.2 Linguagem destino é escolhida	36
3.3 Texto é extraído do arquivo	38
3.4 Tradução do texto	39
3.5 Geração do arquivo no formato .txt	40
3.6 Arquivo final em PDF	41
4 CONCLUSÃO	46
5 REFERÊNCIAS	47

1 INTRODUÇÃO

Embora, conceitos como globalização tem seus marcos a serem discutidos após a expansão marítima capitalista iniciada no século XV, o contato entre povos com diferentes linguagens possui data praticamente imensurável. Porém a Bíblia, o livro mais vendido da humanidade (Livro Guinness dos Recordes, 2023), também o primeiro a ser impresso, possui datas de escritas que vão de 1500 a.C. e 450 a.C., onde acredita-se que este também tenha sido uma das primeiras traduções importantes no ocidente, onde por exemplo, os judeus haviam passado muito tempo sem contato com a linguagem hebraica e as escrituras tiveram de ser traduzidas para sua compreensão. De acordo com a denominada Carta de Aristeias, ou Carta a Filócrates, uma obra helenística do século II a.C., a Septuaginta foi a versão traduzida do hebreu para o grego das Sagradas Escrituras (Renan Constantino Colli, 2019, p.6).

Para o contexto brasileiro, a tradução teve um impacto gigantesco a toda estrutura catequista jesuíta. Pode-se dizer que os primeiros tradutores foram os denominados "línguas", compostos em sua maioria de criminosos portugueses que eram inseridos de forma induzida aos nativos, tentando aprender de maneira totalmente forçada as formas de comunicação dos povos locais, buscando se tornar posteriormente intérpretes às autoridades portuguesas coloniais (HISTÓRIA DA TRADUÇÃO NO BRASIL: PERCURSOS SECULARES, 2016).

Tão revolucionário quanto a impressão, para a divulgação de conteúdo escrito, foi o surgimento da digitalização dos mesmos, popularmente denominados como E-books, que deram as caras em meados de 1998 por essa denominação, mas em 1987 a empresa Microsoft, desenvolvedora dos sistemas operacionais computacionais da linha DOS e, posteriormente da linha Windows, já havia produzido livros digitais na forma de disco, como por exemplo o Encarta, uma enciclopédia lançada em 1993, que acabou migrando para o formato online com o tempo, até ser descontinuada.(Ednei Procópio, 2013,p.6). Inúmeros títulos no formato de CD (*Compact Disc*) foram lançados no Brasil, incluindo o saudoso Dicionário Aurélio.

Aprofundando-se nas tecnologias utilizadas nas produções bibliográficas digitais, a primeira que se pode destacar é a Plain Vanilla ASCII, utilizada no Projeto Gutenberg de textos eletrônicos, nome que homenageia o criador da impressão. Essa extensão digital

de arquivos evoluiu até se tornar o formato TXT (abreviação para Arquivo de Texto), que se trata de um formato de arquivo utilizado até os tempos atuais, para arquivos de texto que não contém formatação, como por exemplo, não possuindo adição de negrito ou itálico. Buscando permitir que qualquer usuário pudesse captar documentos provindos de qualquer fonte, compartilhamento de documentações eletrônicas, exibição em diversas plataformas digitais e imprimir-los de em modelo formatado em qualquer computador, surgiu assim o modelo PDF (*Portable Document Format*), que acabou se tornando um padrão de arquivos para Ebooks, devido a sua praticidade de manuseio (O que é PDF?. Adobe Systems,2023. Disponível em <https://www.adobe.com/br/acrobat/>. Acesso em 11/11/2023.).

Devido a globalização, o mais provável é encontrar a maioria desse conteúdo em inglês, necessitando assim, a maioria dos brasileiros, uma tradução para o português, algo para suprir esse problema.

De acordo com Cindy Leopoldo, sobre um trabalho de tradução de livros, ela pode levar cerca de quarenta e cinco a sessenta dias para traduzir um livro médio, que possui cerca de trezentas páginas, com conteúdo de complexidade média, o que daria então aproximadamente algo entre seis e sete livros por ano (Investir em tradução para economizar depois PublishNews,2010.Disponível em: <https://www.publishnews.com.br/materias/2010/08/17/59461-investir-em-traducao-para-economizar-depois-12> Acesso em 17/11/2023).

Com o objetivo de dinamizar traduções virtualmente, em 2006 a empresa Google lançou o serviço de tradução Google Translate que, inicialmente, realizava traduções apenas entre inglês e árabe, sendo até os tempos contemporâneos, a maior plataforma gratuita de tradução. Posteriormente, cerca de 130 idiomas foram adicionados e a aplicação foi lançada em forma de aplicativo, tanto para Android quanto para iOS. A princípio, tratava-se de um serviço de tradução automática estatística, usando documentos e transcrições das Nações Unidas e do Parlamento Europeu para coletar dados linguísticos. Em suas primeiras versões, a tradução não ocorria entre os idiomas diretamente, pois a aplicação primeiramente traduzia o texto para o inglês e depois transcrevia o idioma de destino na maioria das combinações de idiomas que coloca em sua grade, procurando padrões em milhões de documentos para ajudar a decidir quais

palavras escolher e como organizá-las no idioma de destino. Já que o sistema automatizado do Google apenas cruzava possíveis traduções literais para expressões e palavras isoladas. Não havia uma preocupação com a gramática, por exemplo, um elemento que as máquinas não eram capazes de entender

Por muito tempo, a tradução provinda do *Google Translate* foi criticada por variar muito de um idioma para outro, além de diversos problemas semânticos, o que gerou uma visibilidade ruim ao se tratar de traduções que ultrapassavam de termos simples, já que o sistema automatizado do *Google* apenas cruzava possíveis traduções literais para expressões e palavras isoladas. Não havia uma preocupação com a gramática, por exemplo, um elemento que as máquinas não eram capazes de entender. Até chegou a ser considerada a hipótese de criar uma equipe de curadores humanos para checar cada tradução e ter certeza de que a ordem gramatical estava sendo respeitada, mas devido a uma demanda de trabalho irreal, seria impossível acompanhar as constantes transformações da linguagem humana.

Buscando então solucionar este problema, a partir de novembro de 2016, o algoritmo foi totalmente reformulado, dessa vez então utilizando um mecanismo de tradução automática neural, baseado em *Machine Learning*, que por sua vez são modelos computacionais inspirados pelo sistema nervoso central de um animal que são capazes de realizar o aprendizado de máquina bem como o reconhecimento de padrões. Com o uso dessa tecnologia, foi então possível realizar a tradução de frases inteiras de uma vez, em vez de apenas parte por parte. Esse contexto mais amplo é utilizado para encontrar a tradução mais relevante, que então é reorganizada e ajustada para ser mais como um humano falando com gramática adequada. (Google, 2023).

Diversas aplicações distintas dentro da própria base do *Google Translate* foram desenvolvidas, como a tradução a partir de reconhecimento de imagens, quanto a tradução provinda e retornada em áudio, além da criação de bibliotecas de programação, que se tratam de coleção de subprogramas utilizados no desenvolvimento de software. A partir de uma biblioteca, é possível implementar em diversos tipos de softwares distintos, o uso da ferramenta *Google Translate* em alguns casos, até sem conexão com a internet.

De acordo com o próprio site da *Amazon* (2023), um excelente exemplo de aplicação de tradução corporativa cotidiana é o *Amazon Translate*, uma ferramenta em forma de serviço que realiza traduções dinâmicas entre diversas plataformas dentro

dos serviços *Amazon*, como arquivos compartilhados entre bate-papos ou e-mails, dentro da plataforma *Amazon Web Services*, popularmente conhecida como AWS.

A partir do ponto que existem diversas plataformas corporativas de tradução automáticas, torna-se então palpável o desenvolvimento de não só serviços pagos, mas também de algum que possa ser distribuído de forma gratuita para a tradução de documentos, objetivo deste projeto. Após toda a imersão possibilitada pelo capítulo de referencial teórico deste artigo, tem-se o capítulo do desenvolvimento prático do projeto tradutor de de PDFs, demonstrando sua funcionalidade.

Este projeto tem como objetivo estudar a viabilidade de desenvolver um sistema tradutor de arquivos digitais de texto, com o foco no formato PDF, tendo em vista que é o mais amplamente difundido ao se tratar de artigos acadêmicos e ebooks, contribuindo para o acervo de aplicações gratuitas de tradução de texto.

Trata-se também de um incentivo a leitura de documentações limitadas a somente quem possui fluência em outro idioma, já que traduzir manualmente seria extremamente trabalhoso para quem não possui vasto conhecimento de determinado idioma.

A análise de viabilidade desse projeto se torna fundamental para avaliar se essa tecnologia é viável tecnicamente e economicamente. Foi necessário investigar a eficiência do sistema, que insere um determinado arquivo e retorna um novo já com tradução, necessitando que fosse conferido a qualidade desse arquivo retornado. Além disso, aspectos como tempo de execução, viabilidade comparando a outros métodos de tradução também foram considerados.

No aspecto econômico, é importante a análise do custo de produção, incluindo os valores sobre determinadas tecnologias externas, tempo de produção de todo o sistema, custeamento de uma possível implementação online. A demanda de mercado e a possibilidade de comercialização do produto a um preço competitivo também foram levadas em conta.

Do ponto de vista social e de acordo com o site British Council(2023), ampliar o acesso a livros e artigos a uma população onde menos de cinco por cento sabe inglês, tem um impacto extremamente relevante, dinamizando um processo que seria muito custoso anteriormente para quebrar essa barreira social.

Diante dessas considerações, este artigo tem como objetivo apresentar uma análise de viabilidade abrangente de um software que traduz arquivos PDF. São explorados aspectos técnicos, econômicos e sociais, buscando avaliar o potencial dessa tecnologia inovadora e sua aplicabilidade em um contexto real. Ao fim da análise, espera-se obter informações relevantes que possam subsidiar a tomada de decisão e a implementação desse projeto sustentável e promissor.

O objetivo geral deste trabalho é realizar desenvolvimento de um tradutor de arquivos PDF, com uma análise de viabilidade técnica, econômica e social sobre o impacto desse protótipo, visando avaliar a possibilidade de sua aplicação como uma forma de dinamizar processos de tradução de artigos e ebooks, difundindo conteúdo anteriormente limitados pela barreira social do idioma.

Como objetivos específicos, pode-se citar então:

- Realização de uma revisão bibliográfica sobre métodos de tradução de conteúdos escritos ao longo da história, desde períodos arcaicos até contemporâneos, contemplando até tecnologias móveis de uso diário;
- Investigação de materiais e métodos para o desenvolvimento de uma aplicação de tradução, analisando diversas tecnologias dispostas no mercado da programação, considerando critérios de eficácia e viabilidade;
- Realização de testes experimentais para avaliar a eficiência do sistema, comparando com métodos manuais já dispostos no mercado;
- Realização de análise econômica detalhada, considerando os custos de produção e potencial de disfunção e comercialização do produto;
- Análise de resultados obtidos e fornecer recomendações sobre a viabilidade técnica e econômica, identificando possíveis melhorias e aplicações futuras;
- Desenvolvimento do software do protótipo, com a escolha da maneira mais simples e eficaz de realizar o objetivo principal, que é a tradução dos conteúdos em PDF.

A ideia de um projeto relacionando a tradução de arquivos PDF surge com a necessidade ampliar o acesso de diversas documentações que geralmente possuem a restrição linguística, que por sua vez dificulta a divulgação de tais conteúdos a maiores públicos. Superando a limitação do idioma, é possível não só que, por exemplo, um número maior de brasileiros consiga consumir artigos acadêmicos internacionais, como

também atingem a possibilidade de divulgar seus conteúdos para fora, amplificando ainda mais a visibilidade de seus projetos.

2 REFERENCIAL TEÓRICO

A compreensão dos princípios das tecnologias, não só as de tradução e manipulação de arquivos, como também das aplicações da linguagem de programação utilizada, são a base para o entendimento deste projeto, sendo fundamental para embasar uma análise e fornecer uma visão abrangente sobre a razão da utilização de cada tecnologia em cada ponto do desenvolvimento e como no fim foi possível obter os resultados esperados.

2.1 Informações digitais

Como artifício de retransmissão de informação, diversos registros do conhecimento humano foram passados ao longo do tempo, como o livro. Quando a escrita foi inventada, não existia o conceito de livro ou de organização de informações.

Há quatro ou cinco mil anos, na antiga Mesopotâmia, atual Iraque, houve um salto na utilização da escrita como método de registro, organização, armazenamento e compartilhamento de ideias. Contudo, somente com a invenção da imprensa, provinda do alemão Johannes Gutenberg, criou-se um método de se registrar informação e conhecimento em um suporte mais barato e popular. Com a tecnologia aprimorada por Gutenberg, o registro do conhecimento humano deu mais um salto quantitativo. Documentos anteriormente à disposição apenas de reis e de uma elite religiosa, ganharam maior acessibilidade depois que a sua manufatura possibilitou economia de escala (A revolução dos Ebooks: A indústria dos livros na era digital, Ednei Procópio, 2016, p.24).

O livro digital não poderia existir se a escrita como forma de registro não tivesse sido passada pelos povos da humanidade até os tempos contemporâneos. Com graças a todas revoluções tecnológicas ao longo da história, conteúdos escritos como livros enfim tem uma nova forma de divulgação, onde seu conteúdo é extraído para o digital, tornando assim os populares *eBooks*, que por sua vez, podem ter seu conteúdo acessados através de diversos aparelhos eletrônicos populares, como smartphones e tablets, substituindo o formato impresso, uma vez que a reação importância está no conteúdo e não no formato em que está expresso.

No campo dos conteúdos escritos, não somente livros possuíram imigração do papel para o digital, como também artigos acadêmicos, cartas e até documentações oficiais, com por exemplo a papelada de um registro civil de cartório.

O formato digital de arquivo mais utilizado, para todos esses exemplos citados anteriormente é o PDF (*Portable Document Format*), criado pela empresa Adobe Systems em 1993, foi desenvolvido com o objetivo de permitir que um documento salvo em um computador pudesse ser aberto em outro sistema com a mesma fidelidade do local de origem. Foi criado para ser um formato universal, estável e portátil. Indo contra toda a ideia desenvolvida anteriormente sobre a migração do conteúdo físico para o digital, uma das principais funções (mas não única) do PDF é ser o formato padrão para arquivos a serem enviados para a impressão, o que acaba agregando na materialização de diversos documentos oficiais. Mas o PDF é muito flexível, podendo ser usado como apresentações (com vídeos, áudio e tudo mais), documentos de texto, formulários interativos e etc. (O que é PDF?. Adobe Systems, 2023. Disponível em <https://www.adobe.com/br/acrobat/>. Acesso em 11/11/2023).

Figura 1: Ícone de aplicação do Adobe Acrobat, ferramenta oficial da Adobe para PDFs

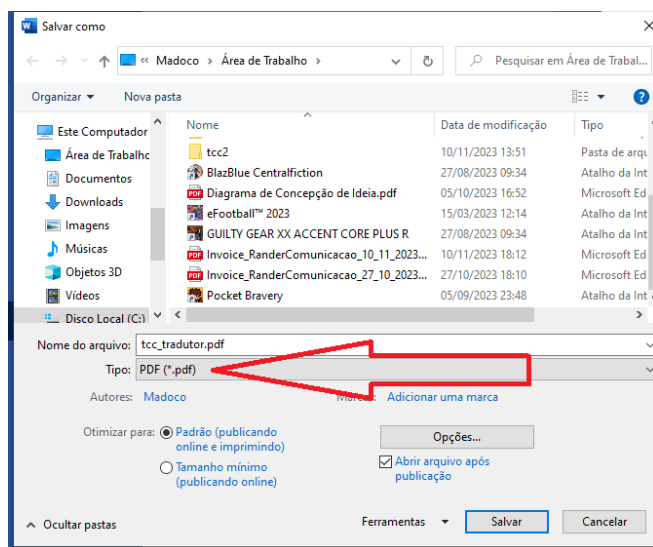


Fonte: Adobe Educa (2023)

Embora as plataformas da Adobe Systems sejam as oficiais para o manuseio de tal, diversas empresas possuem sua versão de geração de arquivos no formato PDF, como por exemplo os sistemas Microsoft Office e Google Docs, que possibilitam que um arquivo escrito em “.docx”(formato de documentos de texto para edição, que funciona

apenas nas versões mais novas do editor de textos da Microsoft, ao contrário dos antigos “.doc”), seja posteriormente convertido para PDF, para que possa usufruir de todas as vantagens que o formato oferece.

Figura 2: Amostra prática de escolha de formato de arquivo em PDF



Fonte: Autor (2023)

Diversas vantagens em se usar este formato de arquivo justificam no motivo de ele ser tão difundido. Pode -se citar:

- **Universalização:** O PDF é um formato amplamente reconhecido e suportado por diferentes sistemas operacionais, dispositivos e softwares. Isso significa que os arquivos PDF podem ser abertos e visualizados consistentemente em diferentes plataformas, garantindo que o conteúdo seja acessível para todos.
- **Mantém a formatação:** Este tipo de arquivo mantém a formatação original do documento, que vai desde estilos, *layout* até tipo e tamanho de fontes de escrita, concluindo então a existência de uma estabilidade do conteúdo a ser apresentado pelo documento, independentemente de estar na forma impressa, digital num computador de mesa ou num dispositivo móvel, como um celular;
- **Interatividade:** O formato permite que sejam inclusos diversos recursos que podem realizar certa interação com aplicações tanto internas como externas ao próprio, como *hiperlinks* (*link* de direcionamento), botões e formulários. Isso torna o documento mais interativo e dinâmico, dando ao usuário a possibilidade

de navegar entre as páginas por um sumário, acessem sites relacionados, preencham formulários e realizem outras diversas ações;

- **Compactação:** O PDF suporta a compressão de dados, sendo a ele então permitido reduzir o tamanho original de determinado arquivo, sem que haja uma perda significativa de qualidade, facilitando o compartilhamento e o armazenamento de documentos eletrônicos, especialmente quando há restrições de largura de banda ou de espaço de armazenamento. Geralmente compartilhamento via determinadas redes sociais de comunicação direta, como WhatsApp, por exemplo, possuem determinada restrição de tamanho de compartilhamento, tendo o PDF então, uma vantagem a ser usado no mesmo;
- **Vasculhável:** É possível que o arquivo PDF possa ser criado com utilização de recursos de OCR (Optical Character Recognition), permitindo que a transformação de texto em imagens em texto pesquisável, facilitando a localização de informações específicas dentro do documento, tornando-o mais eficiente para a pesquisa e a recuperação de informações. No geral, ferramentas de leitura para o formato PDF, possuem uma barrinha de pesquisa de informações, assim como nas navegações WEB;
- **Segurança:** Por fim, um dos benefícios mais famosos do PDF, ao se tratar de documentações oficiais, é o fato do formato suportar recursos de segurança, como criptografia, que permitem proteger o conteúdo do documento contra acesso não autorizado. É possível que haja definição de senhas para a abertura do arquivo, edição ou até impressão, garantindo então integridade dos dados do arquivo.

2.2 Software

É possível definir que Software é uma sequência de instruções escritas para serem interpretadas por um sistema computacional para executar tarefas específicas. Também é possível definir como programas, dados e instruções que comandam o funcionamento de um computador, smartphone, tablet e outros dispositivos eletrônicos. É responsável por mensurar a compreensão e execução dos comandos do usuário para o sistema computacional em que está aplicado. Em questão mais ampla, sobre a computação no geral, classifica-se como a parte imaterial e lógica, possuindo como função o

fornecimento das instruções que são executadas com o auxílio do hardware, que por sua vez é toda a parte física de determinado dispositivo. (Edson Nascimento Silva Júnior, 2016, p.24).

Figura 3: Diferenças entre Software e Hardware



Fonte: Gyga Systems (2014)

Por sua vez, o sistema operacional é o principal software de sistema de uma máquina. *Drivers*, *firmwares*, utilitários e tradutores de linguagem de programação também estão inclusos nessa categoria.

Sobre *Softwares* de Programação entende-se como conjuntos de ferramentas voltadas ao desenvolvimento de outros softwares, com utilização de linguagens de programação, a partir de um ambiente visual de desenvolvimento integrado, popularmente conhecidos como IDE (*Integrated Development Environment*).

Software de Aplicativos são os programas utilizados nos dispositivos que permitem ao usuário executar uma série de tarefas nas mais diversas áreas de atividade, podendo ir desde simples calculadoras digitais até um editor de vídeos e imagens.

Sistemas de Comunicação são programas que estabelecem a comunicação direta e em tempo real entre os usuários. Com ampliação do acesso à internet e a dispositivos

digitais, tornaram-se parte do cotidiano das pessoas, utilizados tanto para fins pessoais quanto comerciais, incluindo toda e qualquer rede social.

Utilizações voltadas para fins recreativos estão relacionadas por exemplo a *Softwares* de jogos, contudo, os mesmos ainda podem ser utilizados em finalidades educacionais.

Programas hospedados em servidores na internet e que podem ser acessados através de navegadores de internet, podem ser denominados de *Softwares Web* (ou apenas *Web App*). Geralmente suas funcionalidades são semelhantes de outros sistemas, contudo, requerindo apenas a conexão com a internet para serem executados, não necessitando de um *hardware* robusto.

Como este projeto trata-se de uma aplicação a ser executada em um computador localmente, sendo um tradutor de arquivos digitais, classifica-se como *Software* de Aplicativo, porém, caso a aplicação seja futuramente hospedada online, pode também entrar na classificação de *Web App*.

2.3 Linguagem de Programação

Pode-se definir que uma linguagem de programação é uma linguagem escrita e formal que, através de uma série de instruções e regras usadas, permite que um programador escreva um conjunto de ordens, ações consecutivas, dados e algoritmos para criar programas que controlam o comportamento físico e lógico de uma máquina. A comunicação que ocorre entre o programador e o dispositivo ocorre por meio dessa linguagem, permitindo especificar, com precisão, aspectos como quais dados um software deve operar, como esses dados devem ser armazenados ou transmitidos e quais ações o software deve executar, de acordo com cada circunstância variável. (O que é uma linguagem de programação e quais os tipos existem?, 2019. Disponível em: <https://rockcontent.com/br/blog/linguagem-de-programacao/> . Acesso em 17/11/2023).

Resumindo, a linguagem de programação é um sistema de comunicação estruturado, composto por conjuntos de símbolos, palavras-chave, regras semânticas e sintáticas que permitem o entendimento entre um programador e uma máquina. Nem todas as linguagens computacionais são de programação, mas todas as linguagens de

programação são computacionais, já que sempre existe a comunicação entre hardware e software.

Quanto a tipagem das linguagens de programação, é possível resumir em dois tipos simples, sendo o idioma de baixo nível, que permite a comunicação interna da máquina e cada instrução tem seu código de operação exclusivo, trabalhando com os impulsos elétricos do hardware e seus bits na forma digital (valores apenas expressos em 0 e 1), e linguagens de alto nível, que facilitam a aquisição das instruções que o programador fornece à máquina, sendo extremamente mais palpáveis ao programador comum pois, enquanto o profissional insere dados no idioma conhecido, a máquina os absorve na linguagem de máquina através de tradutores ou compiladores, permitindo reduzir o tempo de programação, entender mais facilmente a tarefa a ser executada, permitir que o programador se desconecte da operação interna da máquina, entre outros. Em outras palavras, a linguagem de baixo nível está próxima das linguagens de máquina, enquanto a linguagem de alto nível está mais próxima da compreensão e da linguagem humana.

A escolha de uma linguagem de programação no desenvolvimento de determinada aplicação, depende de vários fatores, como os requisitos do projeto, preferências pessoais de quem está programando, ecossistema de desenvolvimento, eficiência, entre outros.

2.4 Python

Cada vez mais a implementação da linguagem Python tem ganhado destaque no ramo do desenvolvimento de sistemas, possuindo uma comunidade ativa e engajada de desenvolvedores. Isso significa que há uma abundância de recursos, bibliotecas e frameworks disponíveis. Além disso, é possível encontrar facilmente ajuda em fóruns, tutoriais e documentação online.

Python é conhecido por ter uma sintaxe clara e legível, o que facilita a leitura e escrita de código. Isso contribui para uma curva de aprendizado mais suave, especialmente para iniciantes em programação (Flávio Codeço Coelho, 2007, p.20). A linguagem é tão versátil a ponto que pode ser usada em uma variedade de domínios, como inteligência artificial, desenvolvimento web, aprendizado de máquina, análise de dados, automação de sistemas, entre outros. Essa versatilidade é uma vantagem para desenvolvedores que desejam trabalhar em diferentes áreas.

Python é executado em várias plataformas, como Windows, macOS e Linux. Isso significa que o código escrito em Python geralmente é portátil entre diferentes sistemas operacionais sem a necessidade de grandes modificações, sendo extremamente simples realizar a instalação da linguagem e de todos os seus pacotes via terminal em todos esses sistemas operacionais. Como baseado em Linux, o sistema Android também possui suporte de desenvolvimento para Python, muito embora seja mais complexo criar linhas de código sem um teclado físico, partindo do ponto que a grande maioria das utilizações em Android encontram-se no uso de funcionalidades voltadas ao digital.

Figura 4: Logo Python



Fonte: Python Org (2001)

Python possui um ecossistema rico de bibliotecas que facilitam o desenvolvimento de uma ampla gama de aplicações. O uso das bibliotecas Python é bastante simples. Dentro de cada biblioteca podem existir módulos. Dentro dos módulos pode ter uma função específica que você deseja para uma finalidade. Por exemplo, o Django é popular para desenvolvimento web, o NumPy e o pandas são amplamente usados para manipulação de dados, e o TensorFlow e o PyTorch são poderosos para aprendizado de máquina. Existem bibliotecas por exemplo, que possuem a função de dar suporte para o desenvolvimento de interfaces gráficas, para utilização de determinada aplicação pelo usuário, como é o caso da PysimpleGUI.

Para realizar a instalação de qualquer biblioteca externa no Python, utiliza-se o no terminal do computador ou da própria IDE, o comando “*pip install*”, que é seguido pelo nome da biblioteca. No geral, no fórum de cada uma das bibliotecas, é possível encontrar comandos que ensinam a instalar, desinstalar, verificar versões, alterar para uma anterior ou mais recente, etc..

Neste projeto, no capítulo de Materiais e Métodos, destaca-se a utilização de bibliotecas de manipulação de arquivos, como a PyPDF2, que possui a finalidade de fazer uma varredura em todo o conteúdo no formato PDF, auxiliando no manuseio dos dados lá presentes (PyPDF2, 2023, Disponível em: <https://pypdf2.readthedocs.io/en/3.0.0/index.html>, Acesso em 17/11/2023). Por fim, importante também destacar a biblioteca Translate, que por sua vez, localmente, sem uso de uma conexão à internet, consegue traduzir os conteúdos em que é aplicada em tempo real, seguindo seus formatos pré-definidos de código, dando suporte fundamental para este projeto (translate, 2023, Disponível em: <https://pypi.org/project/translate/>, Acesso em 17/11/2023) .

Importante ressaltar que, mesmo que as bibliotecas tenham um papel essencial na redução de linhas de código, uma vez que se tratam de subprogramas prontos que são instalados nos pacotes dos projetos em que são implementados, todo o manuseio destes subprogramas exige leitura da documentação que geralmente é disponibilizado em seus respectivos fóruns, já que em aplicações específicas elas não vão resolver tudo sozinhas, necessitando que o desenvolvedor entenda seu funcionamento e abstraia da biblioteca as funções que irão auxiliá-lo no momento correto.

Ao se discutir sobre produtividade, importante ponto no mercado, principalmente seguindo modelos baseados em metodologias ágeis, totalmente difundidas no mercado da tecnologia da informação, a sintaxe concisa e a facilidade de uso do Python destacam-se. O desenvolvimento em Python muitas vezes requer menos linhas de código do que em outras linguagens, o que pode acelerar o ciclo de desenvolvimento.

Além da estrutura base da linguagem já ser mais simples do que a maioria das demais, a utilização das bibliotecas citadas anteriormente tem contribuição gritante, já que um código que poderia gastar uma quantia gigantesca de linhas se reduz muitas vezes a menos de duas, como é demonstrado no capítulo de Materiais e Métodos deste projeto.

2.5 Aprendizado de Máquinas – *Machine Learning*

Buscando técnicas que objetivam prover os softwares com a habilidade de melhorar seu desempenho em determinada atividade, através da experiência, a teoria de Aprendizado de Máquina é baseada nos princípios do aprendizado indutivo, partem de um ponto onde os modelos são determinados a partir de um conjunto de dados ou representações de execuções anteriores do sistema, analisando tempo de execução, quantidade de acertos e erros, por exemplo.

Geralmente, o aprendizado indutivo encontra-se aplicado por algoritmos que processam um conjunto de dados e extraem um modelo capaz de explicar ou representar os dados sob algum aspecto. Com este modelo, é possível então realizar uma análise que explique ou represente um novo dado, que é apresentado posteriormente (Lucas F. Brunialti, et al, 2015, p.203).

Resumindo, sobre o aprendizado indutivo, é possível classificar em três subtipos bem intuitivamente denominados de supervisionado, não supervisionado e semi supervisionado. Ao se tratar dos supervisionados, pode se dizer que nele os algoritmos realizam uma parametrização de um certo modelo a partir do erro medido entre respostas obtidas e esperadas. Já nos não supervisionados, a parametrização de um modelo ocorre com o ajuste com base na maximização de medidas de qualidade das respostas obtidas. Por fim, nos sistemas semi supervisionados, existe a caracterização pelo uso de algoritmos híbridos, que fazem uso dos recursos de correção de erro do supervisionado e da maximização de medidas de qualidade, provenientes dos sistemas não supervisionados, conseguindo extrair o melhor de ambos os modelos conforme a necessidade.

O aprendizado de máquina tem uma ampla gama de aplicações em diversos setores, incluindo reconhecimento de padrões, processamento de linguagem natural, visão computacional, diagnóstico médico, finanças, entre outros. À medida que os modelos são treinados com mais dados e refinados, eles podem melhorar sua capacidade de realizar tarefas específicas sem intervenção humana direta.

2.6 Redes Neurais

Mesmo com a desenfreada evolução tecnológica cada vez mais perceptível, implementada de forma agressiva socialmente, é esperado o aparecimento cada vez mais no cotidiano de novos hardwares e softwares cada vez mais potentes e com poder de processamento maior, contudo, ainda pode-se dizer que processador mais potente que se tem é um componente biológico que existe a milhares de anos: o cérebro humano (Felisbino, 2012).

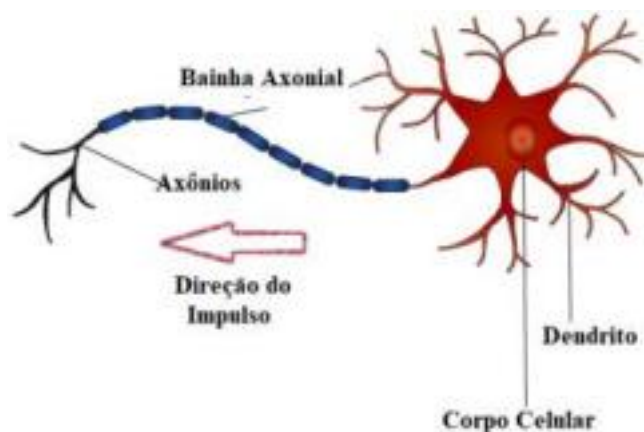
Independente de quão básica ou complexa seja uma atividade humana, nenhuma consegue fugir da necessidade de ser coordenada por algum órgão do corpo e essa, função essa que é atribuída ao sistema nervoso, composto por mais de 10 bilhões de neurônios que trabalham em conjunto para realizar estas funções com excelência. A parte mais fundamental desse sistema gira em torno dos neurônios que apresentam a propriedade de responder a estímulos, também analisados como diferenciais de potencial elétrico sobre o organismo, de forma instantânea. Cada impulso gera uma resposta a ser passada para uma próxima cadeia de neurônios (Furtado, 2019).

É perceptível que a capacidade de aprendizagem, analisada pelo cérebro humano se dá por quatro fatores:

- Capacidade de aprendizagem: capacidade de aprender algo novo a todo instante;
- Tolerância a falhas: mesmo que algum neurônio erre, ele não interfere nos demais;
- Processamento de informação inconcreta: mesmo com uma informação que possa ser incompleta ou contraditória, é possível processá-la e gerar um raciocínio correto;
- Paralelismo: A quantidade de neurônios trabalhando simultaneamente (Raubert, 2005).

O neurônio biológico é composto basicamente por três componentes essenciais: os dendritos, o corpo celular e o axônio, cada um possuindo uma função única e de extrema importância. Na figura abaixo é possível visualizar o corpo celular:

Figura 5: Representação de um neurônio biológico



Fonte: Furtado (2019)

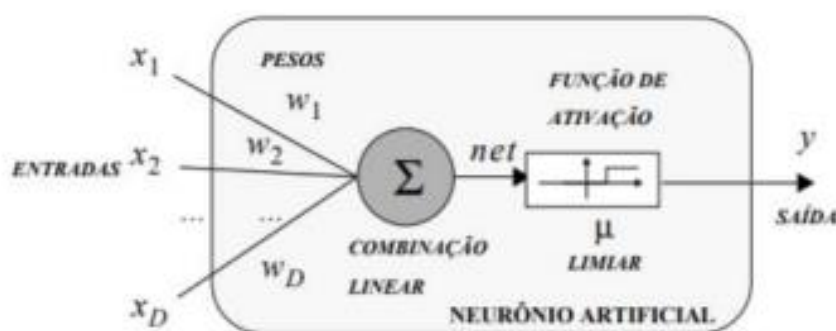
Quando uma informação é recebida ao organismo, ocorre um processamento da mesma dentro do cérebro por cerca de 10^{11} neurônios, onde a saída da informação do corpo celular é feita através do axônio que possui diversas ramificações em suas pontas, de tal forma que permite um neurônio transmitir uma mesma informação a diversos outros neurônios. Essa informação é transmitida por meio de pulsos elétricos que vão chegar ao receptor por via da sinapse (comunicação entre neurônios) que existe no encontro do axônio emissor e do dendrito receptor, sendo este um ciclo que acontece durante todo tempo (Rauber, 2005).

Inspirado nas redes neurais biológicas, ocorreu o surgimento então das Redes Neurais Artificiais, ou RNA, um dos campos mais antigos da Inteligência Artificial, tendo início nos anos 40 com Walter Pitts e McCulloch, um matemático e um neurologista. É o estudo no qual se busca assimilar a forma de pensar dos seres humanos em máquinas, baseando-se no funcionamento dos neurônios e a transmissão de dados por sinais sinápticos (Felisbino, 2012). RNAs estruturam-se em modelos computacionais que buscam realizar similaridade matematicamente entre estrutura e funcionamento do cérebro biológico inteligente, de tal forma que busca possibilitar que o computador em questão seja capaz de aprender e tomar decisões com base em seu aprendizado (Fleck; Tavares; Eyng; Helmann; Andrade, 2016).

Os principais aspectos ao se tratar de RNA são a arquitetura e o algoritmo de aprendizagem. Esta divisão aparece naturalmente pela forma como a rede é treinada, pois, diferente de um computador normal, redes neurais artificiais não são programadas e sim

treinadas através de exemplos de treinamento, onde ela percorre diversas vezes todas suas ligações neurais, afim de realizar diversos testes entre os mesmos. O algoritmo de aprendizagem é quem generaliza este dado e guarda em parâmetros chamados de pesos (Rauber, 2005). O modelo básico de um neurônio artificial foi proposto por Pitts e McCulloch em 1943, e busca modelar matematicamente a forma de funcionamento de neurônios biológicos para as máquinas, sua estrutura está retratada na figura abaixo (Rauber, 2005).

Figura 6: Representação do neurônio artificial



a

Fonte: Rauber (2005)

O aprendizado, como discutido também já apresentado anteriormente a se tratar do aprendizado de máquinas no geral, é uma parte de extrema importância no processo de construção de uma RNA, pois é nesta etapa que o sistema adquire todo o “conhecimento” processado através do processamento de informações. (Bishop, 1995).

Discutindo o aprendizado supervisionado no campo das RNAs, nota-se que sofrem interferência externa, pois seu treinamento é realizado de forma a alcançar um alvo pré-determinado para que chegue a resposta buscada, ou seja, é entregue a RNA os dados e um observador externo monitora para manter os resultados dentro do padrão esperado. Caso o resultado tenda a se desviar do resultado esperado, este supervisor ajusta os pesos sinápticos, resolvendo a questão do “erro”, como discutido anteriormente sobre o Aprendizado de Máquinas. Já em aprendizado não supervisionado, a RNA não tem indicação das classes existentes e nem a qual classe pertence uma amostra, ela recebe a

carga de dados e deve procurar padrões para tentar agrupar as informações. (Cardon; Muller, 1994).

2.7 Sistemas de tradução automática

Com o objetivo de dinamizar os processos de tradução virtualmente, buscando quebrar as barreiras comunicativas pela internet a fora, foram desenvolvidos diversos sistemas de tradução automática, que por sua vez são aplicações ou serviços online que utilizam tecnologias de aprendizagem de máquinas para traduzir grandes quantidades de texto de e para qualquer um dos seus idiomas suportados. Geralmente, a ideia é que esses sistemas realizem a tradução de uma língua fonte de um determinado texto, para um diferente idioma destino, podendo realizar vise versa posteriormente também.

Com uma análise bem superficial sobre os sistemas de tradução automática, é possível dizer que conceitos por trás da tecnologia de tradução de máquina e as interfaces para usá-lo são relativamente simples, contudo possuindo tecnologias por trás dele que são extremamente complexas e a reunião de várias tecnologias de ponta, citando principalmente a denominada *Deep Learning* (Aprendizagem Profunda), tecnologia provinda da inteligência artificial, mas que trabalha também com grandes dados, linguística, computação em nuvem e APIs (*Application Programming Interface*) da *Web*.

Os primeiros modelos de tradutores automáticas se esbanjavam da tecnologia conhecida como SMT (*Statistic Machine Translator*), em português traduzida como Máquina de Tradução Estatística, que realiza uma análise estatística avançada para estimar as melhores traduções possíveis para uma palavra dada o contexto de algumas palavras, geralmente tendo seus dados armazenados em um banco de dados da qual ela faz a conferência. Ao invés de escrever regras artesanais para traduzir entre línguas, os modernos sistemas de tradução abordam a tradução como um problema de aprendizagem da transformação do texto entre línguas a partir das traduções humanas existentes e aproveitando os avanços recentes nas estatísticas aplicadas e na aprendizagem automática. SMT tem sido usado desde meados da década de 2000 por todos os principais provedores de serviços de tradução, indo desde o Google Translator até o Microsoft Translator. (Tradução automática, Microsoft, 2023, Disponível em:

<https://www.microsoft.com/pt-br/translator/business/machine-translation/>, Acesso em 17/11/2023).

O desenvolvimento das NMTs (*Neural Machine Translator*) acarretou uma mudança radical no mercado da tecnologia de tradução, resultando em traduções com uma qualidade extremamente superior aos modelos anteriores, buscando cada vez mais se aproximar da fala real humana, uma vez que existiam diversas críticas aos modelos de tradução anteriores, já que por se tratarem muitas vezes de traduções literais dos termos, facilmente um texto perdia seu contexto semântico, dificultando aplicações mais práticas de tradução, como a de traduzir textos grandes ou auxiliar em diálogos estrangeiros. As tecnologias de tradução com uso das NMTs tiveram seu início de implementação para usuários e desenvolvedores no fim de 2016 (Tradução automática, Microsoft, 2023, Disponível em: <https://www.microsoft.com/pt-br/translator/business/machine-translation/>, Acesso em 17/11/2023).

As traduções automáticas com uso de rede neural possuem toda fundamentabilidade distinta em como são estruturadas e executadas ao se comparar com as arcaicas traduções por estatística. Ao se realizar treinamento de rede neural, cada uma das inserções de texto é codificada, representando suas características únicas dentro de um determinado par de idiomas para a aplicação de tradução, mas também poderia se tratar de diversas outras dimensões, dependendo de onde a rede é aplicada.

Sobre o método de funcionamento, existem algumas etapas para a tradução, onde:

1. Inicialmente cada palavra (sua codificação gerada, mais precisamente), codificada em um vetor de pequenas dimensões, atravessa uma primeira camada de neurônios artificiais, codificando-a em vetores de dimensões maiores, representando a palavra dentro do contexto das outras palavras da frase inserida ao tradutor;
2. Para realizar um ajuste mais apurado do texto inserido, é realizado um processo repetido, recodificando as inserções anteriores. A matriz de saída final é usada por uma camada que irá usar esta matriz e a saída de palavras previamente traduzidas para definir qual a palavra, a partir da frase que foi inserida no início, será traduzida a seguir;
3. Por fim, a camada de tradução, realiza seu papel sobre a palavra selecionada, em seu equivalente de idioma destino mais apropriado. Ocorre também uma realimentação no sistema, onde a saída obtida anteriormente é utilizada para

calcular a próxima palavra da frase de origem a ser traduzida, para garantir qualidade semântica na frase.

Figura 7: SMT x NMT



Fonte: Translab TV, Machine Translation (2023)

Usufruindo desta aplicabilidade, os resultados obtidos com utilização das NMT's se provam mais eficazes que os que são gerados com uso das SMT's, aproximando cada vez mais de uma tradução humanizada.

Como citado no tópico sobre as bibliotecas Python, a bibliotecas Translator foi selecionada para cumprir a função de traduzir o texto, dentro da execução do projeto desenvolvido em conjunto deste artigo. Esta biblioteca por sua vez, bebe da fonte do algoritmo de tradutores NMT's, uma vez que absorve todo seu funcionamento da parte aberta das tecnologias de tradução disponibilizadas no mercado, como o Microsoft Translator e Google Translator, tendo então a precisão desejada na realização da tradução.

3 MATERIAIS E MÉTODOS

A razão principal do projeto é o desenvolvimento de uma aplicação que realize a tradução de arquivos em formato PDF, formato de arquivo de texto totalmente difundido, utilizando a linguagem de programação Python com diversas bibliotecas auxiliares, devido a simplicidade de implementação de código.

O software desenvolvido irá receber o arquivo PDF proveniente do usuário, realizará a extração da parte do texto do arquivo, realizará a tradução desse texto e, posteriormente retornará um novo arquivo PDF já com esse texto traduzido.

A escolha da linguagem de programação Python, deve-se principalmente por se tratar de uma linguagem de programação dinâmica e orientada a objetos, que pode ser implementada em qualquer tipo de aplicação, sendo ela científica ou não (Flávio Codeço Coelho, 2007, p.20). Frequentemente tem-se o uso dessa linguagem relacionado a grandes ganhos de produtividade, devido a versatilidade de aplicações, incontáveis bibliotecas que podem ser utilizadas, simples aprendizado e manutenção de código. A linguagem também pode ser instalada em todo e qualquer tipo de plataforma computacional, englobando todas as plataformas de uso pessoal.

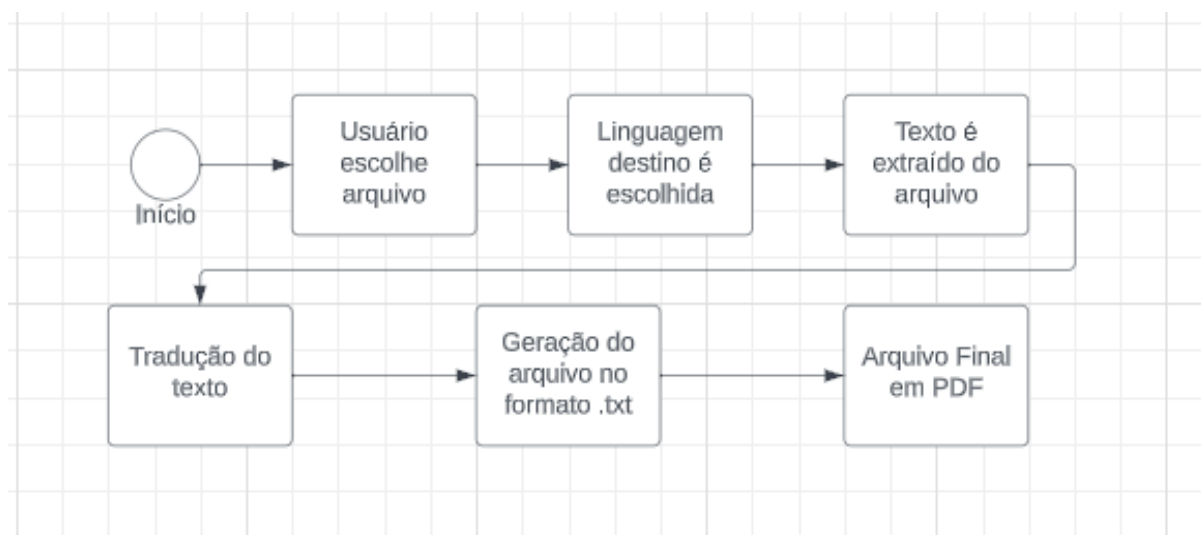
Aplicativos desenvolvidos em Python podem ser distribuídos em várias plataformas das em que foram desenvolvidos, mesmo sem necessariamente tenham o Python previamente instalado. Embora existam sim algumas bibliotecas que não sejam *OpenSource* (código aberto), a linguagem em si é um software livre, não impondo qualquer venda para sua utilização e implementação. Por fim também é importante citar sua extensibilidade, onde é possível utilizar até rotinas escritas em outras linguagens, como C ou Fortran.

Após a escolha da linguagem de programação, é de suma importância a escolha também da interface para o desenvolvimento do código, onde para Python, existem além de ferramentas genéricas para confecção de código, interfaces dedicadas, o que para determinadas funções matemáticas extremamente complexas, aceleram o tempo de execução de código, como por exemplo o Spyder - Anaconda e o Pycharm. Contudo, para esta aplicação, foi escolhida a interface Visual Studio Code, por se tratar habitualmente utilizada pelo desenvolvedor do projeto em questão, devido a ser um software mais leve de execução que as ferramentas dedicadas, além de extremamente customizável.

A aplicação em si foi desenvolvida visando ser executada inicialmente apenas em computadores de mesa e de forma local, já que o objetivo gira em torno de provar a possibilidade do projeto. Contudo posteriormente pode alcançar outras plataformas, podendo ser implementada de forma *Web* e *Mobile*.

Detalhando melhor o processo realizado pelo projeto, tem-se a figura, ilustrando em forma de diagrama a linha de raciocínio presente no código.

Figura 8: Diagrama de funcionamento do projeto de tradução de PDFs



Fonte: Autor (2023)

3.1 Usuário Escolhe Arquivo

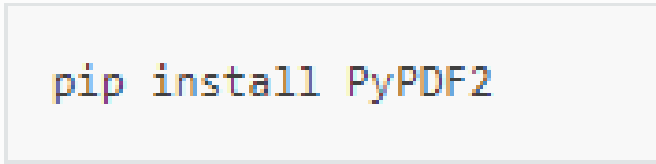
O usuário escolhe um arquivo .pdf em seu computador, para então ser inserido no sistema. Foram utilizados nesse projeto apenas arquivos escritos apenas pelo autor, devido a diversos PDFs disponibilizados na internet possuírem criptografia para a proteção de seus dados. Arquivos PDF podem ser facilmente gerados utilizando a maioria dos sistemas processadores de arquivos de texto, como o Microsoft Word ou o Google Docs, onde inclusive, todos os testes gerados para esse projeto são provenientes, visto que assim é possível burlar a criptografia citada.

Em quesitos técnicos, essa inserção do arquivo no sistema ocorre graças à biblioteca Python conhecida como PyPDF2, que realiza o reconhecimento do arquivo PDF e de todo o seu conteúdo. Trata-se de uma biblioteca gratuita e de código aberto, capaz de dividir, mesclar, cortar e transformar páginas de arquivos PDF. Ele também pode

adicionar dados personalizados, opções de visualização e senhas a arquivos PDF. PyPDF2 também pode recuperar texto e metadados de PDFs (Site PyPDF). A versão PyPDF2 é a que obteve mais suporte, tendo sua última versão lançada em 2022, sendo perfeitamente compatível com o Python 3.10, versão mais amplamente utilizada e melhorada atualmente da linguagem.

É importante ressaltar que, em uma interface de desenvolvimento para Python, ao utilizar uma biblioteca nova, é necessário que além de que ela seja importada, ela deve ser instalada no terminal da interface, para que todos os seus pacotes de subprogramas sejam embutidos nela. Com o propósito de unificar esse processo, as bibliotecas podem ser instaladas pelo código “pip install”, seguido pelo nome da biblioteca. Para este caso então, a linha de código a ser usada no terminal da IDE será:

Figura 9: Comando de instalação da Biblioteca Python – PyPDF2

A screenshot of a terminal window with a light gray background. The text 'pip install PyPDF2' is displayed in a monospaced font. The word 'pip' is in blue, 'install' is in green, and 'PyPDF2' is in red. The text is centered within a white rectangular box that has a thin gray border.

Fonte: PyPDF2 (2008)

Todas as bibliotecas em seus respectivos fóruns de suporte, possuem um guia de instalação, desde para essa forma de instalação dinâmica utilizada diretamente no terminal, quanto para a instalação manual, geralmente utilizada quando se tem algum problema em versionamento.

Após já ter a biblioteca instalada, é necessário também que ela seja importada ao código fonte, para que só assim possa ter funções implementadas na aplicação. Na imagem a seguir, é possível notar a importação da biblioteca ao código e em seguida o momento em que ela faz de forma exemplificada a imputação de um arquivo PDF a ser executado.

Figura 10: Códigos relacionados à biblioteca PyPDF2

```
from PyPDF2 import PdfReader  
reader = PdfReader("pdf5.pdf")
```

Fonte: Autor (2023)

Para todas as importações de bibliotecas em Python, utiliza-se o método “*from*”, sendo esse um padrão da linguagem. A segunda linha por sua vez trata-se já de um método da biblioteca, que é a função *reader*, que realiza o reconhecimento do arquivo exemplo chamado “pdf5.pdf”.

3.2 Linguagem destino é escolhida

A segunda parte do processo está relacionada a um dos principais pontos do projeto, que é o idioma a qual o arquivo será traduzido. Não só para a tradução, mas também para o reconhecimento da língua em que o texto está, foi utilizado desta vez a biblioteca Python denominada como Translate. Translate é uma ferramenta de tradução simples, mas poderosa, escrita em Python com suporte para vários provedores de tradução. Até o presente momento, em seu fórum está descrito oferece integração com API de tradução da Microsoft, API Translated MyMemory, LibreTranslate e APIs gratuitas e profissionais do DeepL, que embora, não sejam tão difundidas como a do Google Translate, possuem reconhecimento comercial.

A biblioteca Translate se provou extremamente significativa, pois não só realiza a tradução, mas já reconhece automaticamente o idioma em que o texto original está, o que economiza mais linhas de código ainda. Devido a vasta maioria das documentações em PDF serem encontradas em inglês, esse idioma se tornou o mais testado para este projeto.

Para realizar a instalação desta biblioteca, é necessário usar a seguinte linha de código no terminal:

Figura 11: Comando para instalação da biblioteca de Python de tradução

```
$ pip install translate
```

Fonte: Pypi Translate (2017)

Como citado anteriormente, em Python é necessário que após a instalação da biblioteca, ela deve ser importada ao código, utilizando o método “*from*”, como na imagem abaixo:

Figura 12: Códigos relacionados à biblioteca de tradução

```
from translate import Translator
translator= Translator(to_lang="pt")
```

Fonte: Autor (2023)

A escolha do idioma final por sua vez é realizada de forma manual, utilizando um padrão técnico conhecido como ISO 639, uma norma técnica padrão da "Organização Internacional para Padronização", que instituiu códigos que representam cada língua (idiomas) do planeta. É aplicada em áreas como linguística, lexicografia, terminologia e bibliografia. Na Internet, por exemplo, é usada para indicar a língua em que se encontra um documento para websites como HTML (*HyperText Markup Language*) ou um trecho do mesmo - o código da língua portuguesa, por exemplo, é “pt”; o do inglês, “em” (Infoterm, 2011. Disponível em http://www.infoterm.info/standardization/iso_639_1_2002.php. Acesso em 17/11/2023).

3.3 Texto é extraído do arquivo

Com o arquivo já imputado no sistema, e o idioma já definido, o próximo passo trata do processamento do texto a ser manipulado.

Embora a biblioteca PyPDF, desvendada no primeiro passo da aplicação realize a imputação do arquivo, a extração do seu conteúdo ainda precisa ser moldada de forma

palpável para o desenvolvedor. A maneira mais simples encontrada foi: armazenar todo o conteúdo extraído na forma de texto do arquivo e inseri-lo em uma variável, um objeto capaz de reter e representar um valor ou expressão, por sua vez que será convertida tipo *string*, onde possa armazenar cadeias de caracteres como palavras e textos em geral. (Denomina-se nas ciências da computação, cadeia de caracteres uma sequência de símbolos como letras, números, sinais de pontuação etc.). Então, é realizado uma varredura via código de tudo o que é considerado como texto e um armazenamento numa variável denominada pelo autor como “conteúdo”, possuindo este nome intuitivo para facilitar o entendimento de qualquer usuário que venha a ler o código fonte da aplicação.

A biblioteca PyPDF2 também realiza a contabilização das páginas do arquivo PDF, o que facilita ainda mais a organização do texto.

O código demonstrado na figura a seguir, mostra o momento em que ocorre:

1 - Criação das variáveis de conteúdos e contabilização dos conteúdos em suas respectivas páginas são criadas, seguindo o padrão simplificado da linguagem Python;

2 - A varredura e extração para essas variáveis, utilizando o método “*for*”, que cria um laço (Popularmente denominado como loop), que via estrutura de repetição, analisa dado por dado do texto a ser extraído e vai alocando-o nas variáveis;

3 - Codificação do conteúdo para o formato Unicode, o que o deixa de forma muito mais agradável aos padrões técnicos internacionais, sendo configurados no formato UTF-8(*Unicode Transformation Format*);

4 - Conversão do conteúdo para o tipo string e separação de frase por frase do texto, onde tem-se o entendimento que cada frase deve se finalizar por um ponto final, explícito na utilização do método Python conhecido como “*split*”, que consegue fazer o reconhecimento e separação de conteúdo, o que para os próximos passos do código será de extrema importância, visto que a biblioteca de tradução utilizada, a Translate, possui limitação de apenas quinhentos caracteres traduzidos por vez (Citado em sua documentação), o que impede que um texto grande seja traduzido todo de uma vez, mas que ainda é um valor totalmente aceitável para o valor de caracteres para uma frase, sendo essa a razão pela qual foi escolhido a separação do conteúdo em “frase por frase”. Para caso, o método escolhido fosse “palavra por palavra”, ocorreriam diversas falhas semânticas na tradução, já que se trataria de uma tradução literal em grande parte do

tempo, desconectando-se totalmente da proposta do projeto de realizar traduções agradáveis.

Figura 13: Código de varredura de conteúdo

```
conteudo = ''

pag_conteudo = {}

for indx, pdf_pag in enumerate(reader.pages):
    pag_conteudo[ indx + 1 ] = pdf_pag.extract_text().encode("utf-8")

pag_conteudo = str(pag_conteudo).replace("\\n", "").split(".")
```

Fonte: Autor (2023)

3.4 Tradução do texto

É neste passo onde a tradução ocorre. Utilizando um “*loop*” com o método “*for*” para a varredura do conteúdo da variável que cuida dos conteúdos das páginas, já separados em frases, devido ao processo anterior, se traduz agora com método “*translate*”, proveniente da biblioteca Translator, cada uma das frases do conteúdo extraíndo, alocando-as em sequência, possuindo agora nesta segunda variável, o conteúdo traduzido neste caso para o idioma português, com as devidas noções semânticas.

Embora a biblioteca possua uma utilização gratuita, ela possui suas limitações, não só como a já citada limitação de quinhentos caracteres por vez, mas também sobre um limite diário para usuários anônimos de cinco mil caracteres diários, cinquenta mil caracteres para usuários que possuam o e-mail cadastrado com os desenvolvedores e também a possibilidade de utilizar um plano pago, que busca atender ao menos três vezes o valor dos usuários cadastrados.

De acordo com um artigo da Revista de Educação Básica Brasileira (UFMG), cinco mil caracteres resultam em aproximadamente quatro páginas de puro conteúdo textual, onde embora trata-se de pouco conteúdo, já pode-se citar que o plano de usuários cadastrados consiga suprir cerca de quarenta páginas de texto, o que atende diversos artigos acadêmicos, podendo então traduzir um por dia, levando em consideração apenas os pacotes gratuitos.

Figura 14: Código de tradução do conteúdo

```
for frases in pag_conteudo:
    conteudo = conteudo + translator.translate(frases)+'.'

conteudo = conteudo.replace("{1: b'", "").replace('"}.', "")
```

Fonte: Autor (2023)

3.5 Geração do arquivo no formato .txt

De forma nativa, em Python é possível alocar conteúdos *string* de forma bem simples em arquivos TXT. Utilizando o método “open” o arquivo é gerado e com o método “write” ele é alocado.

Figura 15: Alocação do conteúdo de texto traduzido em um arquivo novo

```
arquivo = open("arquivo_traduzido.txt", "a")
arquivo.write(conteudo)
```

Fonte: Autor (2023)

3.6 Arquivo final em PDF

Finalizando a descrição do projeto seguindo o fluxo diagramado, tem-se então a conversão do arquivo TXT para um arquivo PDF, concluindo então a ideia de traduzir arquivos PDFs.

Aqui também foi utilizado uma biblioteca Python, desta vez a Aspose.words, que consegue realizar diversas conversões de arquivos, não só de TXT para PDF, como a utilizada no projeto. Ela é uma poderosa biblioteca de classes local que pode ser usada para inúmeras tarefas de processamento de documentos. Permite que os desenvolvedores aprimorem seus próprios aplicativos com recursos como geração, modificação, conversão, renderização e impressão de documentos, sem depender de aplicativos de terceiros, por exemplo, Microsoft Word, ou automação.

Para realizar sua instalação, utiliza-se:

Figura 16: Comando de instalação da biblioteca de conversão de arquivos

```
pip install aspose-words
```

Fonte: Aspose-words (2021)

Após realizar sua importação ao código fonte, apenas se usa o método “*Document*” para selecionar o arquivo original e o “*save*” para gerar o arquivo final, colocando cada um em seu formato desejado. Abaixo segue o exemplo de código, onde o “arquivo_traduzido.txt”, gerado no passo anterior, agora é convertido para o “Output.pdf”, arquivo final já em PDF, com o conteúdo do arquivo anterior.

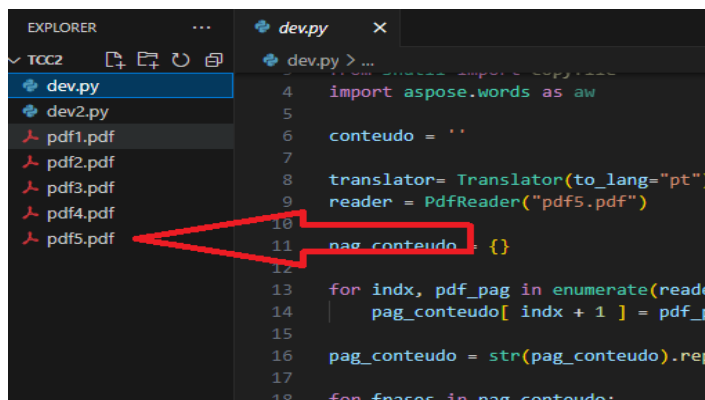
Figura 17: Código de conversão de tipo de arquivo TXT para PDF

```
import aspose.words as aw
doc = aw.Document("arquivo_traduzido.txt")
doc.save("Output.pdf")
print('Fim da geracao do pdf')
```

Fonte: Autor (2023)

Para demonstração prática, será utilizado o arquivo pdf5.pdf. É um arquivo simples, possuindo apenas duas páginas e conteúdo em inglês. Para simplificar o entendimento da aplicação, todos os arquivos de teste foram colocados no mesmo diretório dos códigos.

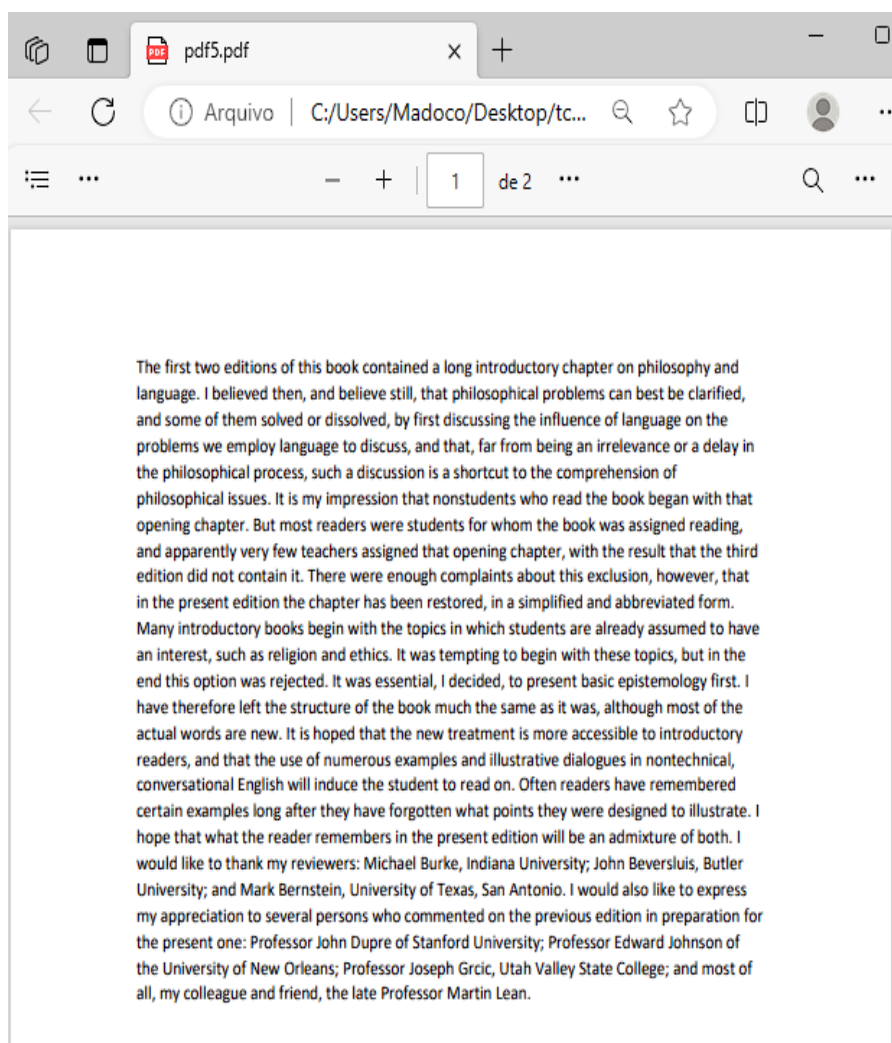
Figura 18: Arquivo “pdf5.pdf” apresentado dentro do diretório do projeto



Fonte: Autor (2023)

Abaixo está a demonstração do arquivo original:

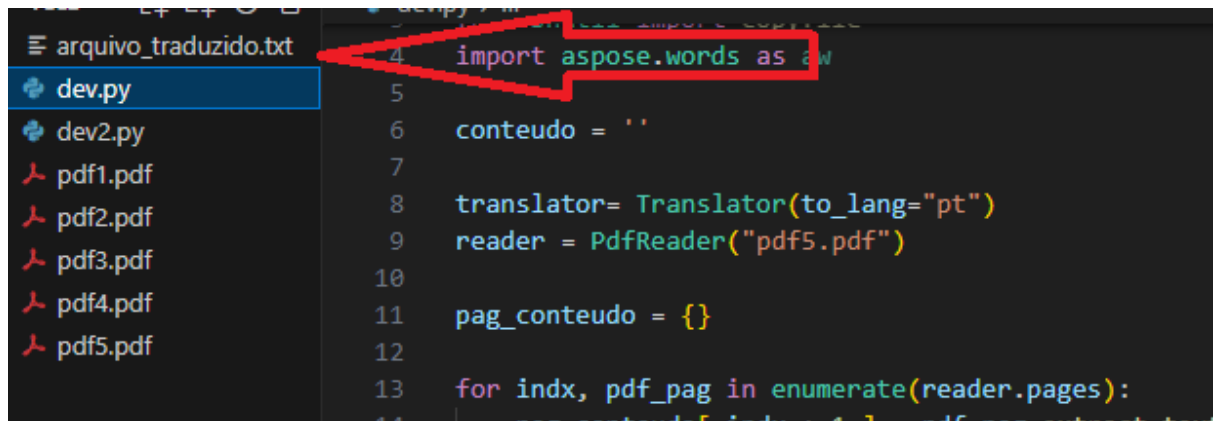
Figura 19: Apresentação do conteúdo em inglês presente em “pdf5.pdf”



Fonte: Autor (2023)

Após a execução é então possível notar o surgimento do arquivo “txt” gerado:

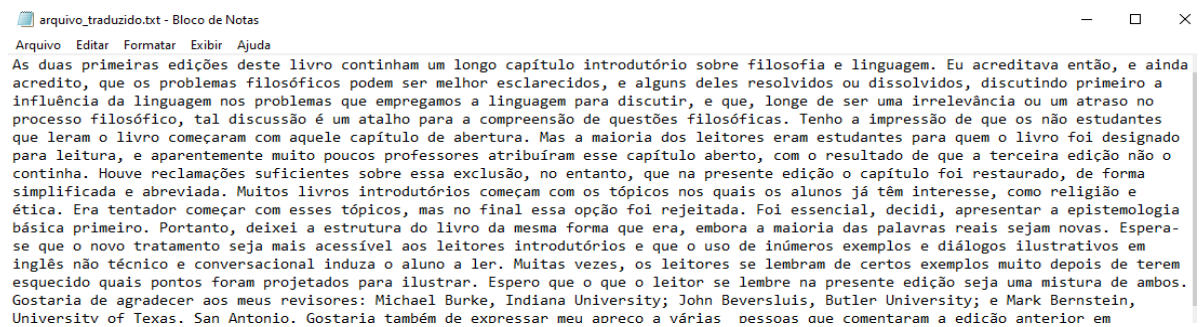
Figura 20: Arquivo “arquivo_traduzido.txt” apresentado dentro do diretório do projeto



Fonte: Autor (2023)

Então pode-se enfim comprovar que o conteúdo foi traduzido:

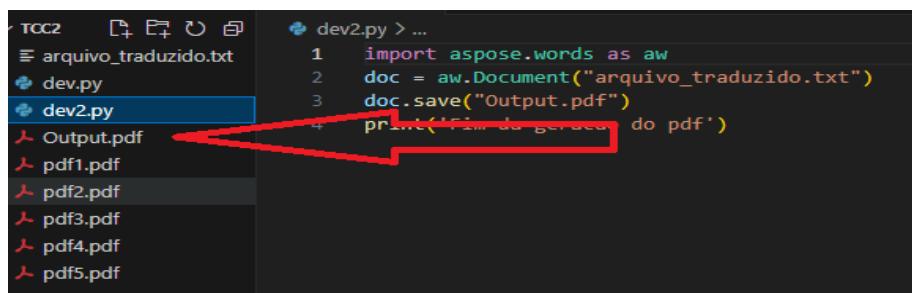
Figura 21: Conteúdo traduzido do documento que foi executado no projeto



Fonte: Autor (2023)

Executando então o código para conversão, tem-se o “Output.pdf”, que é apresentado na figura abaixo:

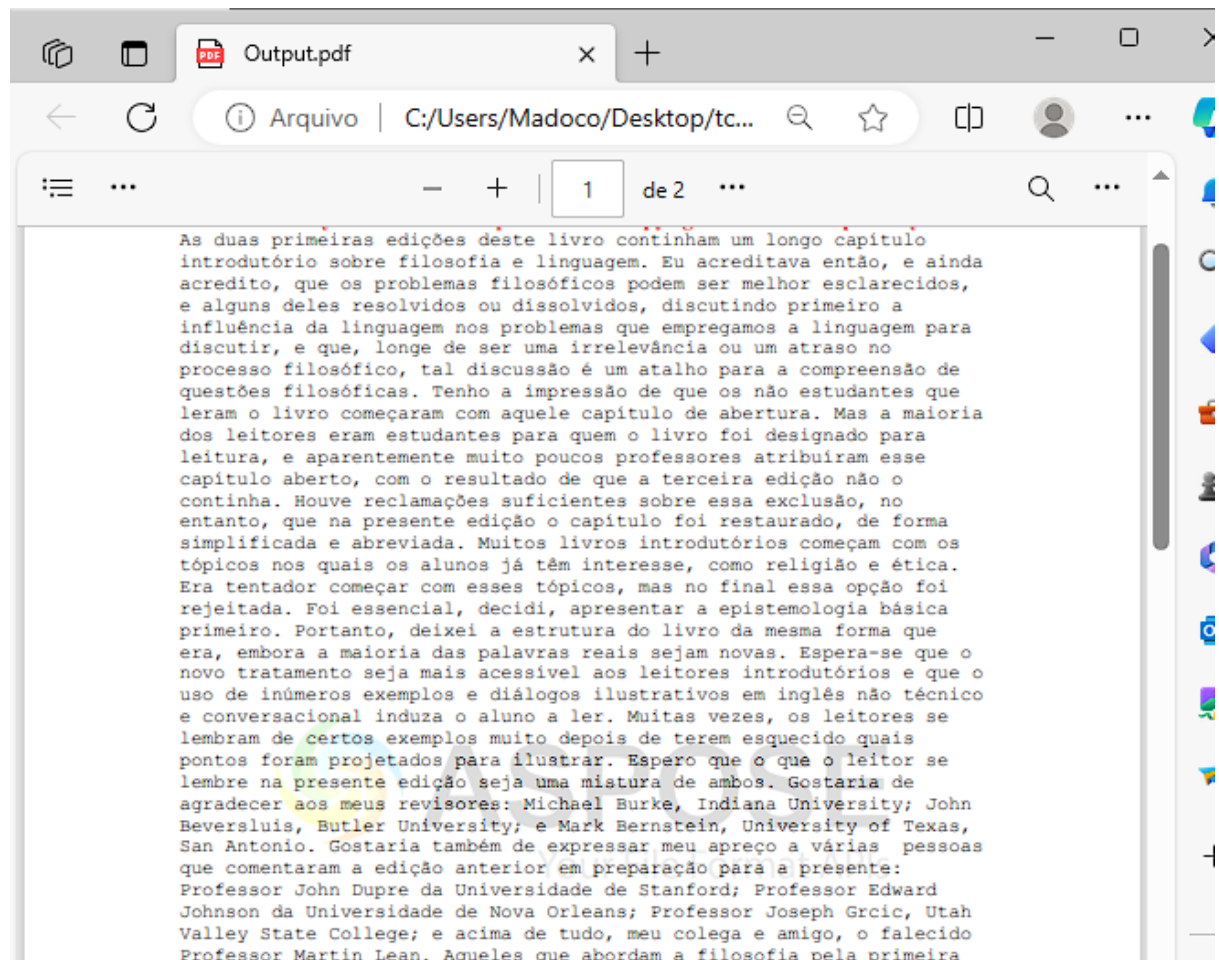
Figura 22: Output.pdf dentro do diretório do projeto



Fonte: Autor (2023)

E enfim o resultado final:

Figura 23: Arquivo com o conteúdo traduzido do original



Fonte: Autor (2023)

4 CONCLUSÃO

Ao decorrer deste trabalho acadêmico, foi explorado em detalhes o desenvolvimento de um software tradutor de arquivos PDF, desde a análise de motivações, concepção do projeto até a implementação. Ao longo desse processo, foram abordados desafios significativos, soluções inovadoras e decisões de design que moldaram a criação deste produto.

Buscando trazer o melhor desempenho e agilidade na produção do protótipo, foram examinados diversos modelos que poderiam trazer o resultado de tradução do objeto alvo, realizando análise de projetos que já existem no mercado, onde foi possível retirar inspirações que agregaram no desenvolvimento deste projeto.

Ao alcançar os marcos estabelecidos, como a prova de que realmente é possível de forma automatizada traduzir arquivos PDF, utilizando ferramentas relativamente simples, do ponto de vista de desenvolvedor de sistemas, fica evidente a possibilidade de aplicação do projeto na internet à fora.

Além disso, é necessário ressaltar a importância do ciclo de *feedback* contínuo, não apenas durante o desenvolvimento, mas também após a implementação. A adaptação às mudanças nas demandas dos usuários e a prontidão para ajustes contínuos são fatores críticos para a longevidade e sucesso contínuo do software, podendo então, possuir um repositório público no GitHub, por exemplo.

5 REFERÊNCIAS

- Adobe Systems. **O que é PDF?**. 2023. Disponível em: <<https://www.adobe.com/br/acrobat/>>. Acesso em 11 nov. 2023.
- BISHOP, Christopher M. **Neural Networks for Pattern Recognition**. 15. ed. Cambridge/UK: Oxford University Press, 2005. Acesso em: 17 nov. 2023.
- BRUNIALTI, Lucas F., et al. **Aprendizado de Máquina em Sistemas de Recomendação Baseados em Conteúdo Textual: Uma Revisão Sistemática**, 2015, p.203.
- CARDON, André; MULLER, Daniel Nehme. **Introdução às Redes Neurais Artificiais**. 1994. 17f. (Tese de Pós Graduação). Curso de Pós-Graduação em Ciências da Computação em Ciências da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 1994.
- CARRER, Felipe. **O que é uma linguagem de programação e quais os tipos existem?**.2019. Disponível em: <<https://rockcontent.com/br/blog/linguagem-de-programacao/>>. Acesso em 17 nov. 2023.
- COELHO, Flávio Codeço. **Computação Científica com Python**, 2007, p.20.
- COLLI, Renan Constantino. **Teorias a respeito do objetivo da carta de Aristeias a Filócrates**, 2019, p.6.
- FELISBINO, Rafael. **Inteligência Artificial e Redes Neurais: Conceitos e Aplicações**. Orientador: Prof. Dr. Luiz Carlos Begosso. 2012. 24f. TCC (Graduação)-Bacharel em Ciências da Computação, Departamento de Informática, Fundação Educacional do Município de Assis, Assis/SP. 2012. Disponível em:<<https://cepein.femanet.com.br/BDigital/arqTccs/0911270619.pdf>>. Acesso em: 17 nov. 2023.
- FLECK, Leandro; TAVARES, Maria Hermínia Ferreira; EYNG, Eduardo; HELMANN, Andrieli Cristina; ANDRADE, Minéia Aparecida de Moares. **Redes Neurais Artificiais: Princípios Básicos**. **Revista Eletrônica Científica Inovação e Tecnologia**. Medianeira-PR, v.1, n.13, p. 47-57, 2016. Acesso em: 17 nov. 2023.

FURTADO, Maria Inês Vasconcellos. **Redes Neurais Artificiais**: Uma Abordagem Para a Sala de Aula. 1. ed. Ponta Grossa/PR: Atena Editora, 2019. Acesso em: 17 nov. 2023.

Giga Systems. **Diferenças entre Software e Hardware**, 2014. Disponível em: <<https://www.gigasystems.com.br/artigo/33/diferenca-entre-hardware-e-software>>. Acesso em 17 nov. 2023.

Guinness World Records 2023, HarperCollins Brasil, 2023.

Infoterm, 2011. Disponível em: <http://www.infoterm.info/standardization/iso_639_1_2002.php>. Acesso em 17 nov.2023.

Microsoft, **Tradução automática**, 2023, Disponível em: <<https://www.microsoft.com/pt-br/translator/business/machine-translation/>>. Acesso em 17 nov. 2023.

Procópio, Ednei. **A revolução dos Ebooks: A indústria dos livros na era digital**, São Paulo/SP, 2013, p.6-13.

PyPDF2, 2023, Disponível em: <<https://pypdf2.readthedocs.io/en/3.0.0/index.html>>. Acesso em 17 nov. 2023.

Python Org. Disponível em: < <https://www.python.org>>. Acesso em 17 nov. 2023.

PublishNews, **Investir em tradução para economizar depois**.2010. Disponível em: <<https://www.publishnews.com.br/materias/2010/08/17/59461-investir-em-traducao-para-economizar-depois-12>>. Acesso em 17 nov. 2023.

RAUBER, Thomas Walter. **Redes Neurais Artificiais**. Vitória/ES, 2005. Disponível em: <https://www.researchgate.net/profile/Thomas-Rauber/2/publication/228686464_Redes_neurais_artificiais/links/02e7e521381602f2bd000000/Redes-neurais-artificiais.pdf>. Acesso em: 17 nov. 2023.

Translab. Machine translation (SMT VS NMT). Disponível em: <<https://translab.lv/en/machine-translation-smt-vs-nmt/>> . Acesso 17 nov.2023.

Translate, 2023, Disponível em: <<https://pypi.org/project/translate/>>. Acesso em 17 nov. 2023.