SOFTWARE DESKTOP PARA CONVERSÃO DE ARQUIVOS: APLICAÇÃO DO PYTHON E USO DA INTELIGÊNCIA ARTIFICIAL PARA APRENDIZADO

João Padilha Moreira, Silvia Marmontel, Luis Chamorro

RESUMO

Este projeto apresenta um sistema casual de conversão de arquivos, no qual o usuário pode converter seus arquivos em sua própria máquina, sem precisar do uso de sites e, consequentemente, aumenta sua produtividade em ambientes profissionais quando necessita deste serviço. Para isto, detalhamos como o sistema foi projetado, construído e como, no fim, houve a relação entre uso de inteligências artificiais para aprendizado e o desenvolvimento do código, incluindo as compreensões de lógicas para conversão de diversos formatos de arquivos. As tecnologias utilizadas no sistema serão apresentadas e explicadas exibindo a maneira como foram aplicadas e revelando brevemente suas origens e, principalmente, suas utilidades e funções no sistema. Ao final, são apresentados os resultados obtidos com o sistema em vista de seus objetivos iniciais, a avaliação desses resultados por parte dos desenvolvedor e qual a continuidade e melhorias futuras aplicáveis ao projeto.

Palavras chaves: Conversão de arquivos, tecnologia, lógica, inteligência artificial, Python

DESKTOP SOFTWARE FOR FILE CONVERSION: APPLICATION OF PYTHON AND USE OF ARTIFICIAL INTELLIGENCE FOR LEARNING

ABSTRACT

This project presents a casual file conversion system, where users can convert their files on their own machines, without the need for websites, thus increasing productivity in professional environments when this service is required. To do so, we detail how the system was designed, built, and how artificial intelligence was used for learning and code development, including the understanding of logic for converting various file formats. The technologies used in the system will be presented and explained, showcasing how they were applied and briefly revealing their origins, as well as their utilities and functions within the system. Finally, the results obtained with the system will be presented in view of its initial objectives, the evaluation of these results by the developer, and the continuity and future improvements that can be applied to the project.

Keywords: File conversion, technology, logic, artificial intelligence, Python

1. INTRODUÇÃO

O avanço tecnológico tem promovido mudanças perceptíveis na sociedade atual, tendo tal afirmativa revelada ao analisar o desenvolvimento exponencial de novos projetos na sociedade entre as mais diversas áreas. Com tais avanços, tornou-se necessário uma nova maneira de lidarmos com dados e informações. Em diversos setores, a manipulação eficiente e inteligente de grandes volumes de dados tem se tornado essencial para otimizar processos e garantir a qualidade da informação gerada (Lutz, 2013). Nesse contexto, a conversão de arquivos e a automação de processos se destacam como áreas de grande potencial para melhorar a eficiência, permitindo que dados em diferentes formatos sejam acessados e utilizados de forma mais prática e integrada (Vander Plas, 2016).

Este trabalho tem como objetivo apresentar o desenvolvimento de um projeto de conversão de arquivos utilizando Python e Inteligência Artificial (IA). O foco está na utilização dessas tecnologias para automatizar a transformação de diferentes formatos de dados (como textos, planilhas e imagens) em informações estruturadas e acessíveis, simplificando processos e facilitando o intercâmbio de informações entre diferentes sistemas. A IA foi utilizada principalmente para apoiar o desenvolvedor no processo de construção da solução, ajudando na organização do código e na identificação de melhores abordagens para resolver problemas relacionados à conversão de arquivos, tornando o desenvolvimento mais eficiente (Bishop, 2006).

Para o desenvolvimento do projeto, foi escolhida a utilização do Python, uma linguagem de programação amplamente reconhecida por sua simplicidade e vasto ecossistema de bibliotecas, foi escolhida por sua flexibilidade e capacidade de escalar conforme a complexidade do projeto (Granger & Hunter, 2014). Além disso, a aplicação de Inteligência Artificial foi um diferencial importante no suporte ao processo de desenvolvimento, ajudando a otimizar as decisões tomadas durante a criação do sistema e oferecendo sugestões que contribuíram para o aprimoramento contínuo da solução (Chollet, 2018), além de auxiliar fortemente no desenvolvimento de novos conhecimentos e habilidades com o uso da ferramenta focado no lecionamento de conteúdos.

O projeto apresentado neste trabalho busca não apenas automatizar a conversão de arquivos, mas também proporcionar uma forma eficiente de organizar e estruturar os dados resultantes, facilitando seu uso em diferentes contextos e sistemas. A solução desenvolvida tem grande aplicabilidade em diversas áreas, permitindo, por exemplo, a integração de sistemas que lidam com dados em formatos distintos e a melhoria da acessibilidade das informações (Kelleher & Tierney, 2018).

Por meio da combinação de Python e IA, este projeto propõe um avanço significativo na forma como os dados são convertidos, oferecendo uma solução mais eficiente e adaptável às necessidades de diferentes tipos de arquivos e sistemas (Sutton & Barto, 2018).

1. Definições do Tema ou Problema

Torna-se fato que é necessário lidar com dados e informações de maneira mais inteligente. Desta forma, será apresentado um sistema desktop que facilite
o processo de conversão de arquivos para usuários que necessitam utilizá-los nos formatos mais comuns que são utilizados atualmente.

2. Delimitações do Tema

O projeto terá o foco para conversão de arquivos mais comumente usados no cotidiano atual. Visando isto, o sistema perimitirá que o usuário realize a conversão de arquivos principalmente de PDF (O tipo de arquivo mais utilizado no mundo devido a sua alta compatibilidade para abertura do arquivo nos mais diversos dispositivos) para outros formatos, como arquivos Word (.docx) e PNG (.png). Junto a isso, o sistema converterá outros tipos de arquivos, como Planilhas Excel (.xlsx), Documentos Word (.docx) e imagens (.png) para o formato PDF.

Para acessar o sistema, construído inteiramente com Python, tanto no processamento e conversão de arquivos (backend) quanto na interface gráfica desenvolvida (frontend), o usuário deverá apenas abrir o *software* e realizar a conversão dos arquivos.

No backend, utilizaremos várias bibliotecas para conversão de arquivos, que realizarão o processamento e transformação dos arquivos dentre os formatos disponibilizados. No frontend, teremos o uso de bibliotecas nativas da linguagem para melhor processamento dos dados enviados e criação da interface gráfica do usuário, da qual se trata de uma única tela para realização das funções do sistema, já que o foco do projeto é simplificar e, assim, facilitar o processo de conversão de arquivos.

Em suma, o projeto visa desenvolver um sistema de conversão de arquivos focado nos formatos mais comumente utilizados no cotidiano, como PDF, Word (.docx), Excel (.xlsx) e PNG (.png). O sistema permitirá que o usuário converta arquivos, como PDFs para formatos Word e PNG, e também o contrário, ou seja, converta planilhas Excel, documentos Word e imagens para PDF. Construído inteiramente com Python, o sistema integrará bibliotecas específicas para o processamento e conversão de arquivos (backend) e usará bibliotecas nativas para a criação de uma interface gráfica simples e intuitiva (frontend), com uma única tela para facilitar a experiência do usuário. O objetivo principal é simplificar o processo de conversão de arquivos, proporcionando uma solução prática e eficiente para usuários que necessitam de uma conversão rápida e sem complicações.

3. Objetivos

O projeto possui como principal objetivo a conversão de arquivos de maneira simples e eficaz, assim como apresentar praticamente a maneira da qual o uso correto de IA pode auxiliar concomitantemente no desenvolvimento de um projeto e ensinar novos conhecimentos aos desenvolvedores.

1. Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema de conversão de arquivos.

2. Objetivos Específicos

- Identificar o uso de sistemas desktop para conversão de arquivos;
- Conferir a funcionalidade das mais diversas bibliotecas Python para esta função;
- Estudar e pontuar a união das esferas do desenvolvimento, aprendizado e uso de inteligências artificiais para incremento no processo.
- Descrever a criação de um sistema para conversão de arquivos.

4. Justificativa

Este trabalho se justifica pela crescente necessidade de soluções que automatizam a conversão de arquivos e melhorem a gestão de dados, especialmente em um contexto onde a quantidade de informações e a diversidade de formatos são cada vez maiores. A utilização de Python, por sua flexibilidade e vasto ecossistema de bibliotecas, aliada à Inteligência Artificial, que apoia o desenvolvimento e otimiza decisões, permite a criação de um sistema eficiente e adaptável. A conversão automatizada e a organização dos dados resultantes proporcionam maior integração entre sistemas e facilitam o uso das informações em diferentes contextos. Essa solução tem grande aplicabilidade em diversas áreas, como gestão empresarial, saúde e educação, impactando positivamente a eficiência, acessibilidade e qualidade dos dados.

2. REVISÃO BIBLIOGRÁFICA

A IA tem suas raízes na década de 1950, quando Alan Turing introduziu a ideia de que as máquinas poderiam simular a inteligência humana. Ele propôs o famoso "teste de Turing" como forma de avaliar se uma máquina poderia exibir comportamentos semelhantes aos humanos. A partir daí, a IA passou por várias fases de desenvolvimento, com avanços em algoritmos e modelos computacionais, embora tenha enfrentado períodos de estagnação devido às limitações tecnológicas da época. Foi só no início dos anos 2000, com o aumento da capacidade de processamento e o acesso a grandes volumes de dados, que a IA experimentou um verdadeiro avanço, especialmente com a popularização das redes neurais profundas.

Atualmente, a IA tem um impacto significativo em diversas áreas, especialmente no desenvolvimento de software. Ferramentas que utilizam IA, como assistentes de programação e sistemas de automação, estão mudando a forma como os desenvolvedores escrevem e otimizam código. Esses recursos ajudam a acelerar o processo de desenvolvimento, oferecendo sugestões de código, corrigindo erros e automatizando tarefas repetitivas, o que permite aos programadores se concentrar em tarefas mais criativas e desafiadoras. Além disso, a IA também tem sido fundamental no aprimoramento das plataformas de aprendizado, oferecendo abordagens mais personalizadas e eficientes para o ensino de novas tecnologias e práticas de programação. Assim, a IA não só melhora a produtividade no desenvolvimento de software, mas também acelera o aprendizado de novas habilidades, contribuindo para o progresso constante da tecnologia (Russell & Norvig, 2010).

1. Tecnologia, educação, desenvolvimento de softwares e utilização de inteligências artificiais na agilização do processo

O uso das IA no aprendizado de novas tecnologias e no desenvolvimento de software tem se mostrado uma ferramenta essencial para otimizar o processo de ensino e aumentar a produtividade dos desenvolvedores. No contexto educacional, a IA permite uma personalização do aprendizado, adaptando os materiais e atividades às necessidades individuais de cada aluno, o que aumenta o engajamento e a motivação. Como afirmado por Siemens (2005), "a aprendizagem conectivista é um processo de conexão de informações, e as ferramentas de IA podem ajudar a formar essas conexões de maneira mais eficaz e personalizada", além de auxiliar como introdutores destes novos aprendizados a serem formados, trazendo informações básicas de maneira clara e pertinente ao momento. Ferramentas como assistentes de codificação baseados em IA, como o GitHub Copilot, oferecem sugestões em tempo real, corrigem erros e ajudam na aplicação de boas práticas, permitindo que o desenvolvedor aprenda enquanto trabalha no projeto.

A IA também facilita a automação de tarefas repetitivas no desenvolvimento de software, como a configuração de ambientes e a geração

de testes automatizados, permitindo que o desenvolvedor se concentre em aspectos mais criativos e desafiadores, tornando o processo de desnvolvimento mais aceitável ao padrão social atual, do qual requer maior agilidade na entrega de novos resultados. Como apontado por Mayer (2019), "tecnologias como a IA são essenciais para reduzir o tempo gasto em tarefas mecânicas e repetitivas, liberando mais tempo para atividades que exigem criatividade e pensamento crítico". Isso reduz a curva de aprendizado ao mesmo tempo em que acelera o desenvolvimento, criando um ciclo contínuo entre aprendizado e execução.

O engajamento no processo de aprendizado tem um impacto direto na qualidade do desenvolvimento de software. Quando o desenvolvedor se sente motivado e bem orientado, ele consegue resolver problemas com mais facilidade, aplicar novos conhecimentos de forma eficaz e melhorar a qualidade do código. Dewey (1938) afirma que "a educação que promove o engajamento e o envolvimento profundo do aluno é a mais eficaz para o aprendizado duradouro e a aplicação prática dos conhecimentos". A IA contribui para esse engajamento ao fornecer feedback imediato e ajudar na resolução de obstáculos de forma ágil, o que, por sua vez, aprimora o desempenho do projeto como um todo.

Portanto, a integração de IA no aprendizado e desenvolvimento de software não só acelera o domínio de novas tecnologias, mas também eleva a qualidade dos projetos, oferecendo um ambiente mais dinâmico e eficaz para a aprendizagem contínua e a criação de soluções inovadoras.

O projeto foi desenvolvido visando tais pontos, e buscou mostrar na prática as afirmativas acima citadas, produzindo um software dos quais o desenvolvedor não possuía conhecimentos sólidos de suas tecnologias, dentro de um período de tempo ágil, e podendo, ao final, ter ambos os os objetivos (desenvolver uma aplicação e aprender novos conhecimentos) atingidos completamente.

2. Usabilidade de sistemas Desktop para o desenvolvimento do projeto

A usabilidade de sistemas desktop foi um fator essencial no desenvolvimento deste projeto, garantindo que a aplicação fosse acessível e eficiente para os usuários. Diferente de sistemas baseados em navegador, um aplicativo desktop permite maior controle sobre os recursos do sistema, como acesso direto a arquivos locais, melhor desempenho em operações complexas e integração com softwares específicos, como o Microsoft Word e Excel. Além disso, a interface desenvolvida com CustomTkinter proporciona um design moderno e personalizável, oferecendo uma experiência mais agradável e intuitiva, mesmo para usuários sem conhecimentos técnicos avançados. A adoção de Python como linguagem de desenvolvimento reflete a flexibilidade e a vasta gama de bibliotecas disponíveis para manipulação de arquivos e automação de processos, características amplamente discutidas por Lutz (2013) e VanderPlas (2016) em suas obras sobre a aplicação da linguagem.

A escolha por um sistema desktop também facilita a execução de tarefas que exigem processamento intenso, como conversões de arquivos que envolvem OCR ou manipulação de imagens. Enquanto aplicações web podem enfrentar limitações de desempenho e segurança ao lidar com arquivos locais, um software desktop tem a vantagem de processar os arquivos diretamente na máquina do usuário, sem a necessidade de uploads para servidores externos. Isso garante maior privacidade e rapidez nas conversões, aspectos essenciais para usuários que lidam com documentos sensíveis ou grandes volumes de arquivos. Além disso, como destacado por Bishop (2006) e Kelleher e Tierney (2018), a aplicação de técnicas de processamento e reconhecimento de padrões em dados textuais e imagens depende de um ambiente computacional robusto, algo mais facilmente alcançado em sistemas locais do que em ambientes distribuídos.

A interação com o sistema operacional é mais eficiente em uma aplicação desktop, permitindo a utilização de bibliotecas específicas para cada tipo de conversão. A integração com o Windows, por exemplo, possibilitou a automação do Microsoft Word e Excel para a geração de PDFs sem perda de qualidade. Esse nível de integração dificilmente seria alcançado em uma aplicação web sem dependências adicionais ou restrições de compatibilidade. Conforme demonstrado por Graham, Knuth e Paterson (2005), a escolha adequada da plataforma de desenvolvimento impacta diretamente na eficiência dos processos computacionais, reforçando a decisão de desenvolver o projeto como um software desktop.

Além disso, a utilização de bibliotecas como EasyOCR para reconhecimento de caracteres em imagens e Pandas para manipulação de planilhas reflete o avanço das ferramentas de automação na área de processamento de arquivos, permitindo que o sistema execute conversões com alta precisão e eficiência. Segundo Granger e Hunter (2014), a manipulação de dados em Python se destaca pela simplicidade e poder analítico, tornando a linguagem uma das mais indicadas para projetos que envolvem transformação e análise de documentos. Dessa forma, a combinação entre uma interface intuitiva e algoritmos robustos garante que o usuário tenha uma experiência fluida e confiável ao utilizar o software.

Por fim, a interface gráfica foi projetada para ser simples e intuitiva, permitindo que os usuários selecionem arquivos e opções de conversão com poucos cliques. A implementação de feedbacks visuais, como a exibição dos arquivos selecionados e a atualização da barra de progresso, melhora a experiência do usuário, tornando a interação mais transparente e previsível. Com isso, o sistema atinge um equilíbrio entre funcionalidade e facilidade de uso, garantindo que qualquer pessoa possa realizar conversões de arquivos sem complicações. Essa abordagem segue princípios de design de software voltados para a experiência do usuário, como descrito por Russell e Norvig (2010), que enfatizam a importância da interatividade e usabilidade na construção de sistemas inteligentes e eficientes.

3. Tecnologias usadas na construção do sistema

De forma geral, o sistema é inteiramente construído em Python, uma das principais linguagens de programação para desenvolvimento desktop. Esta linguagem possui uma sintaxe simples, muitas bibliotecas e aplicações prontas desenvolvidas pela comunidade e é muito versátil para as mais diversas funcionalidades imaginavéis. Entre as bibliotecas utilizadas neste processo, temos o CustomTkinter (FrontEnd), o PyMuPDF (BackEnd), entre outras.

1. Python

O Python é uma linguagem de programação de alto nível, conhecida por sua simplicidade e legibilidade, o que a torna uma das mais populares no mundo da tecnologia. Criada por Guido van Rossum e lançada em 1991, sua filosofia de design enfatiza a clareza do código, o que facilita o aprendizado para iniciantes e a produtividade para desenvolvedores experientes. Sua sintaxe, que se aproxima da linguagem humana, permite que os programadores escrevam código de maneira mais intuitiva e com menos linhas, o que favorece a manutenção e a escalabilidade dos projetos.

Com uma vasta comunidade de desenvolvedores e uma enorme base de bibliotecas e frameworks, o Python tem sido amplamente adotado em diversos campos, como ciência de dados, inteligência artificial, desenvolvimento web, automação e até mesmo em áreas de pesquisa acadêmica. A facilidade de integração com outras linguagens e ferramentas, como C, C++ e JavaScript, também é um dos pontos fortes do Python, permitindo que ele seja utilizado de maneira complementar em projetos mais complexos.

Outro fator que contribui para a popularidade do Python é sua versatilidade. Ele pode ser usado tanto para pequenos scripts simples quanto para grandes sistemas de software. No campo da ciência de dados, por exemplo, bibliotecas como Pandas, NumPy e Matplotlib são fundamentais para análise e visualização de dados. Já no desenvolvimento web, frameworks como Django e Flask proporcionam uma estrutura robusta e eficiente para a criação de aplicativos web dinâmicos.

Além disso, o Python tem se destacado por seu papel em áreas emergentes, como aprendizado de máquina e inteligência artificial. Com bibliotecas poderosas, como TensorFlow, PyTorch e scikit-learn, a linguagem tem sido a escolha predominante para o desenvolvimento de algoritmos de aprendizado profundo, processamento de linguagem natural e redes neurais, entre outros.

Portanto, a popularidade do Python não é apenas uma tendência passageira, mas sim o resultado de sua capacidade de atender a uma ampla gama de necessidades no desenvolvimento de software, aliado à sua sintaxe

limpa e à comunidade vibrante que constantemente contribui para a evolução da linguagem. Em um cenário tecnológico em constante mudança, Python continua sendo uma escolha sólida para desenvolvedores e empresas que buscam uma linguagem confiável, acessível e eficiente.

2. CustomTkinter (CTk)

O CustomTkinter (CTk) é uma biblioteca para o desenvolvimento de interfaces gráficas de usuário (GUIs) em Python, que aprimora o Tkinter com recursos modernos de personalização e novos widgets. Ao contrário do Tkinter, que possui uma aparência padrão e simples, o CTk permite criar interfaces mais atraentes, com cores personalizáveis, bordas arredondadas e novos elementos gráficos, como botões com ícones e sliders estilizados (Granger & Hunter, 2014).

A principal vantagem do CustomTkinter é sua facilidade de integração para quem já utiliza o Tkinter. Ele mantém a estrutura de código do Tkinter, permitindo que desenvolvedores migrem para uma interface mais moderna sem precisar aprender uma nova biblioteca. Além disso, oferece compatibilidade multiplataforma, funcionando em sistemas operacionais como Windows, macOS e Linux (Lutz, 2013).

Embora ofereça melhorias visuais e facilidade de uso, o CustomTkinter ainda é limitado em relação a bibliotecas mais avançadas, como PyQt ou Kivy, que oferecem mais recursos gráficos e controle sobre a performance (VanderPlas, 2016). No entanto, para a maioria das aplicações que exigem interfaces modernas e simples, o CustomTkinter é uma solução eficiente e de fácil implementação.

3. EasyOCR

O EasyOCR é uma ferramenta de reconhecimento ótico de caracteres (OCR) de código aberto, desenvolvida com base em redes neurais profundas, que permite a extração de texto de imagens com alta precisão. Sua principal vantagem é a capacidade de reconhecer texto em mais de 80 idiomas, o que a torna uma solução versátil e aplicável em diversas situações. O modelo foi projetado para lidar com textos em diferentes fontes e formatos, o que permite um bom desempenho mesmo em imagens com variações de qualidade ou layouts complexos.

Além de sua facilidade de uso, o EasyOCR se destaca por ser uma solução eficiente e acessível, o que a torna uma excelente escolha para quem precisa integrar a tecnologia OCR a sistemas personalizados. Sua aplicação é especialmente relevante em processos de digitalização, onde a conversão de documentos físicos ou de imagens em arquivos editáveis se torna necessária. O software também é capaz de reconhecer textos em documentos digitalizados, facilitando a organização e o acesso a informações que antes estavam restritas a arquivos físicos.

No que tange ao projeto desenvolvido, o EasyOCR se torna uma ferramenta fundamental, pois possibilita a transformação de imagens ou PDFs

contendo texto em formatos mais manipuláveis, como arquivos de texto ou documentos editáveis. Isso torna a ferramenta extremamente útil em diversas áreas, como escritórios, escolas e empresas, onde a digitalização de documentos é uma prática comum. Com a integração do EasyOCR, os conversores de arquivos conseguem otimizar o processo de extração de texto, economizando tempo e recursos ao eliminar a necessidade de transcrição manual. Em termos práticos, esta tecnologia foi utilizada para leitura de informações e elementos no documento e sua transformação em outros tipos de arquivos - principalmente PDF - de maneira precisa e simples.

4. Pandas

O Pandas é uma biblioteca de código aberto amplamente utilizada em Python para manipulação e análise de dados. Ela oferece estruturas de dados flexíveis, como o DataFrame, que permite a organização de dados de forma tabular, similar a uma planilha ou banco de dados. Com o Pandas, é possível realizar uma vasta gama de operações, como filtragem, agregação, transformação e análise estatística de grandes volumes de dados, tudo de forma intuitiva e eficiente. A biblioteca é projetada para lidar com dados heterogêneos e ausentes, tornando-a ideal para ambientes onde a qualidade e a estrutura dos dados podem variar.

Além de suas funcionalidades poderosas de manipulação de dados, o Pandas se integra bem com outras bibliotecas da ciência de dados, como NumPy, Matplotlib e Scikit-learn, o que facilita a criação de fluxos de trabalho completos para análise e visualização. Ele também oferece suporte a operações de leitura e escrita em diversos formatos, como CSV, Excel, SQL, JSON e HDF5, tornando a integração com diferentes fontes de dados simples e rápida. Isso torna o Pandas uma ferramenta essencial para qualquer profissional que
trabalhe com análise de dados, seja para projetos de pesquisa, análises de negócios ou desenvolvimento de soluções automatizadas.

No contexto do conversor de arquivos, o Pandas é uma ferramenta extremamente útil para o processamento e transformação de dados extraídos de diferentes formatos. Por exemplo, ao lidar com dados provenientes de arquivos CSV ou planilhas Excel, o Pandas facilita a limpeza, transformação e análise dos dados antes de sua exportação para outros formatos. A biblioteca permite que dados sejam organizados, filtrados e estruturados de maneira adequada, o que torna o processo de conversão mais eficiente e preciso, essencial para ambientes que dependem da qualidade e da integridade das informações extraídas.

5. PyMuPDF

O PyMuPDF é uma biblioteca Python que oferece uma interface poderosa para trabalhar com documentos PDF e outros formatos de arquivo, como XPS, EPUB e imagens. Ela permite a leitura, manipulação e criação de documentos PDF de maneira rápida e eficiente. Com o PyMuPDF, é possível extrair texto, imagens e metadados de arquivos PDF, além de realizar tarefas como modificação de conteúdo, fusão de documentos, adição de anotações e conversão de páginas para diferentes formatos de imagem, como PNG e JPEG. Sua flexibilidade e desempenho tornam-na uma excelente opção para quem busca automatizar o processamento de documentos em projetos de análise e manipulação de PDFs.

Uma das principais vantagens do PyMuPDF é a sua velocidade e baixo consumo de memória ao trabalhar com documentos grandes, o que a torna adequada para aplicações que exigem processamento de grandes volumes de dados. A biblioteca oferece suporte a funcionalidades avançadas, como extração de texto com a preservação da estrutura e formatação original, além de permitir o acesso a elementos específicos do documento, como fontes, tabelas e gráficos. Isso proporciona um controle detalhado sobre o conteúdo do PDF, o que é especialmente útil em processos de extração de dados ou na criação de relatórios automatizados.

No contexto do conversor de arquivos, o PyMuPDF é uma ferramenta essencial para transformar documentos PDF em outros formatos de maneira eficiente e sem perder a qualidade do conteúdo original. Sua capacidade de extrair e manipular textos, imagens e outras informações contidas nos PDFs facilita a adaptação de documentos para diferentes necessidades, como a conversão para formatos de texto ou a criação de novos arquivos PDF a partir de dados extraídos. A flexibilidade do PyMuPDF no tratamento de arquivos PDF o torna uma solução robusta para qualquer aplicação que envolva a manipulação ou conversão de documentos digitais. No projeto, esta biblioteca junto ao EasyOCR, tornou a funcionalidade de conversão para formatos PDF precisa, rápida e eficiente.

6. NumPy

O NumPy é uma biblioteca fundamental para o cálculo numérico em Python, amplamente utilizada em ciência de dados, análise de dados e aprendizado de máquina. Sua principal característica é o suporte a arrays multidimensionais, chamados de ndarray, que oferecem uma maneira eficiente e flexível de armazenar e manipular grandes volumes de dados numéricos. O NumPy permite realizar operações matemáticas complexas, como álgebra linear, manipulação de matrizes e transformações de Fourier, de forma rápida e otimizada. Ele também oferece uma ampla gama de funções e métodos que facilitam a análise estatística, a geração de números aleatórios e a manipulação de dados em geral.

Uma das grandes vantagens do NumPy é sua velocidade, já que suas operações são realizadas em C, garantindo um desempenho superior em comparação com listas nativas do Python. Além disso, o NumPy se integra facilmente a outras bibliotecas da ciência de dados, como Pandas e Matplotlib, tornando-se uma peça chave em qualquer projeto que envolva processamento e análise de grandes volumes de dados numéricos. Sua eficiência e simplicidade permitem que desenvolvedores e analistas realizem tarefas complexas de forma otimizada, economizando tempo e recursos.

No contexto de um conversor de arquivos, o NumPy é útil para manipular grandes conjuntos de dados antes ou após a conversão entre formatos. Ele pode ser utilizado para realizar transformações matemáticas e pré-processamento de dados extraídos de arquivos CSV, Excel ou outros formatos numéricos. A capacidade do NumPy de lidar com grandes volumes de dados de maneira rápida e eficiente torna-o indispensável em aplicações que exigem análise e conversão de dados numéricos, como na transformação de dados entre formatos de planilhas ou na análise de conjuntos de dados científicos.

7. Pillow

O Pillow é uma biblioteca de manipulação de imagens em Python que permite abrir, editar e salvar imagens em diversos formatos, como PNG, JPEG, BMP, GIF e TIFF. Ele é uma versão aprimorada da Python Imaging Library (PIL), oferecendo uma ampla gama de funcionalidades para o processamento de imagens, como redimensionamento, rotação, corte, aplicação de filtros e ajustes de cor. O Pillow também permite a conversão entre diferentes formatos de imagem e a extração de informações como metadados e pixels individuais, facilitando a personalização do processamento de imagens para diversos fins.

Entre as principais vantagens do Pillow é sua simplicidade de uso, tornando-o acessível tanto para iniciantes quanto para profissionais que precisam manipular imagens de maneira rápida e eficaz. Além disso, o Pillow é otimizado para trabalhar com imagens de alta resolução, o que é particularmente útil em aplicações que envolvem processamento em lote ou automação de tarefas de edição de imagem. Sua integração com outras bibliotecas Python, como NumPy e Matplotlib, amplia suas capacidades, tornando-o uma ferramenta versátil para análise visual de dados e projetos de visão computacional.

No contexto de conversores de arquivos, o Pillow desempenha um papel essencial na transformação de imagens entre diferentes formatos. Ele permite não apenas converter entre formatos, mas também realizar ajustes de qualidade, como compressão ou alteração de resolução, antes de salvar a imagem no novo formato. Sua versatilidade na manipulação de imagens o torna uma escolha ideal para conversores de arquivos que exigem otimização ou adaptação de imagens para diferentes necessidades, como no caso de ajustes de tamanho ou conversão para formatos específicos para web ou impressão.

3. METODOLOGIA

Para a construção do software proposto, o primeiro passo foi a seleção das bibliotecas e tecnologias a serem utilizadas. Durante o desenvolvimento do projeto, algumas dessas escolhas foram alteradas, uma vez que novas necessidades surgiram, exigindo a implementação de outras bibliotecas para atender às demandas emergentes.

A principal base de pesquisa para a programação do sistema foi voltada para as bibliotecas disponíveis na internet. Para garantir que as bibliotecas escolhidas fossem compatíveis com o Python e que não causassem erros ou bugs no projeto, foram realizados vários testes de compatibilidade e verificações sobre sua viabilidade. Além disso, pesquisas adicionais foram feitas para solucionar problemas e erros que surgiram durante o desenvolvimento, com foco na utilização de ferramentas de Inteligência Artificial (IA) para detectar falhas no código e na lógica do projeto.

Para tornar o processo de desenvolvimento mais eficiente, foi fundamental explorar as bibliotecas disponíveis na comunidade Python. Isso envolveu a pesquisa de funcionalidades dessas bibliotecas e o estudo de todas as opções possíveis. O uso de Inteligências Artificiais, vídeos e artigos sobre essas ferramentas foi muito útil, especialmente para adquirir conhecimentos básicos e rápidos sobre determinados assuntos. Essa abordagem, integrada ao uso de IA, permitiu otimizar o tempo dedicado ao projeto, tornando o processo de aprendizado mais eficaz e inteligente.

Para explorar todo o potencial das tecnologias escolhidas, o desenvolvedor se aprofundou na documentação de cada biblioteca, o que foi essencial para entender as soluções que poderiam ser aplicadas à problemática identificada. O uso de IA para ampliar as possibilidades e descobrir novas soluções, aliado ao estudo detalhado das bibliotecas, tornou a metodologia de desenvolvimento mais segura, confiável e eficiente, além de acelerar a aquisição de conhecimentos necessários.

Com a base de conhecimentos estabelecida, o desenvolvimento do

projeto seguiu com as bibliotecas selecionadas, começando pelo backend e, posteriormente, avançando para o frontend. Durante todo o processo, foi possível implementar atualizações que ajudaram a evitar erros fatais e bugs, garantindo a estabilidade e o sucesso do projeto.

4. DESCRIÇÃO DA SOLUÇÃO

O software desenvolvido surgiu inicialmente com uma ideia de criação de um sistema que auxiliava o desenvolvedor em sua demanda com conversão de arquivos, porém com o passar do tempo, esta ideia foi adaptada, tendo então como objetivo não somente auxiliar um usuário em específico, mas sim qualquer pessoa que necessitasse de um sistema simples e fácil de utilizar para converter o formato de seus documentos digitais, focando apenas na conversão de formatos de arquivos mais usuais no cotidiano. Como a ideia do sistema foi alterada, e mudou de um público alvo específico para um mais abrangente, o sistema sofreu mudanças de design, contendo uma interface gráfica de estilo mais amigável e simples para os usuários, sobre técnicas de design amigáveis Mark afirma que hoje em dia existem várias informações sobre técnicas de design de interface e padrões que você pode usar quando elabora suas interfaces de usuário, soluções para problemas comuns e recomendações gerais de usabilidade (NUNES, 2017), além disso o sistema contou com alterações em sua programação, tendo funcionalidades novas que permitem conversões em lotes e, ainda, que melhoram o desempenho na conversão de tais arquivos.

O sistema também sofreu mudanças nas escolhas de tecnologias. Inicialmente a biblioteca principal do frontend, ou seja, a que seria responsável pela construção da interface gráfica foi mudado de TKinter, uma biblioteca nativa da linguagem, porém limitada para uso, para a CustomTKinter, que possui layouts mais modernos e minimalistas, tornando a interface mais simples e intuitiva para uso do usuário. A troca de bibliotecas foi fundamental para que houvesse uma maior facilidade no desenvolvimento do sistema.

Já no backend, temos a construção de várias funções específicas para cada tipo de conversão de arquivo utilizando diversas bibliotecas nativas do Python, além de outras criadas pela comunidade ao longo do tempo, mas que possuem uma incrível eficácia e eficiência comprovadas por diversos desenvolvedores que as utilizaram. Entre elas, destacamos o uso de bibliotecas de análise de dados em arquivos em planilha e outros estilos de documentos, como a Pandas, Pytorch, Word32Client, PyMuPDF, entre outras.

Estas foram escolhidas por possuírem um ótimo reconhecimento de elementos em arquivos, tornando o processo mais preciso e evitando eventuais erros desnecessários, além de várias delas serem nativas da própria linguagem de programação, tendo, desta forma, uma compatibilidade maior com o processo geral para conversão de arquivos no sistema, além de melhor se encaixarem no quesito facilidade e velocidade que foram buscados no desenvolvimento de tal projeto.

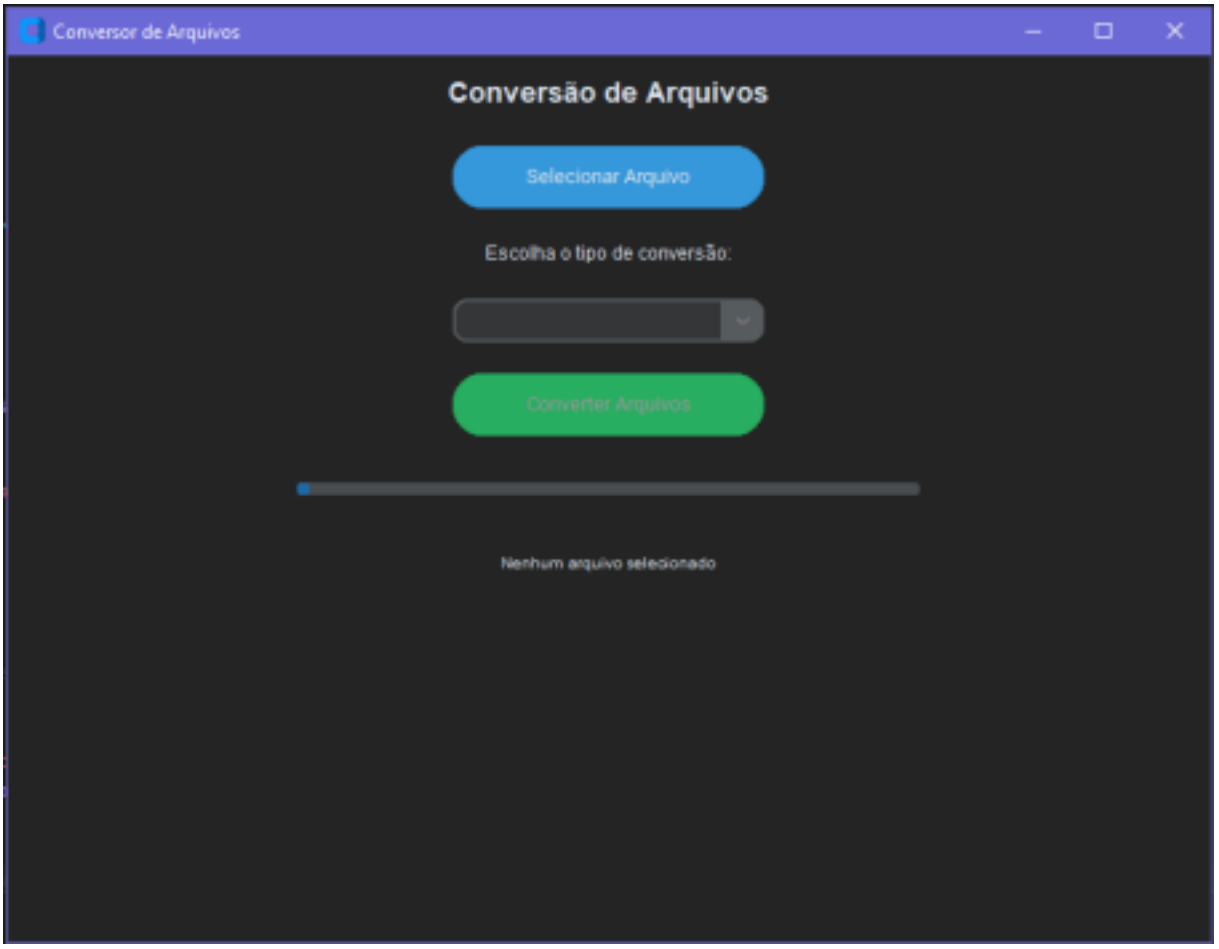
1. Construção do sistema

O projeto iniciou-se com a construção do backend do sistema, ou seja, as diversas funções para conversão de arquivos. Entre as principais funcionalidades que podem ser destacadas, temos os módulos de conversão de arquivos PDF (.pdf) para Word (.docx), Planilhas Excel (.xlsx) para PDF (.pdf), entre outras. Todos estes componentes foram programados utilizando as bibliotecas do Python, tanto nativas, quanto outras criadas pela comunidade, conforme descritas ao longo deste trabalho.

Após isto, foi implementada uma interface gráfica intuitiva, simples e amigável para uso de outros usuários. Ela começou sendo desenvolvida como uma tela muito simples, apenas para seleção dos arquivos e sua conversão, utilizando o TKinter, biblioteca nativa da linguagem. Entretanto, a fim de tornar o sistema mais robusto e convidativo para uso, com o passar do projeto, houve a mudança desta biblioteca pela CustomTKinter, procurando atender tais fins.

A principal função da tela Home, Figura 1, é, como já dito, integrar as funções do sistema e permitir ao usuário utilizá-las de forma mais fácil. Para casos em que o usuário deseja converter mais de um arquivo, foi adicionado uma barra de porcentagem para que este se oriente quanto ao progresso já realizado até então.

FIGURA 1 - Interface Gráfica Do Projeto



Fonte: Captura de tela feita pelo autor.

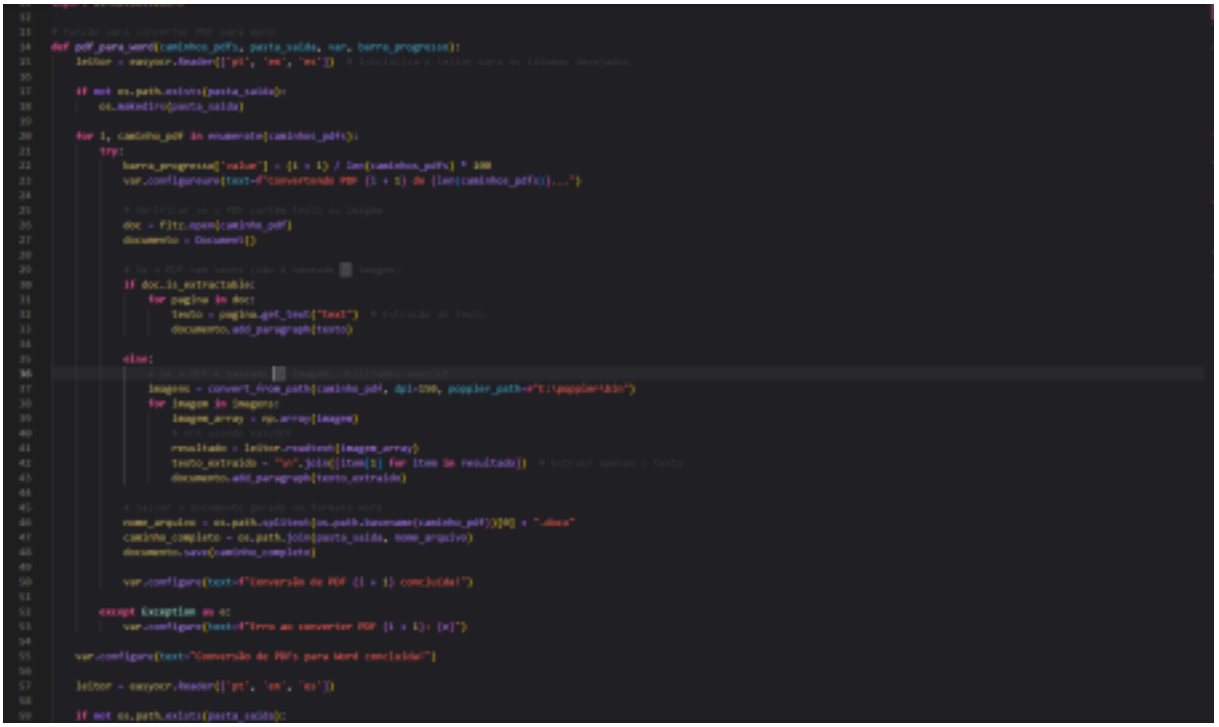
1. Processamento em Segundo Plano e Conversão de Arquivos em Lotes

O sistema foi projetado para garantir uma experiência de usuário fluida e sem interrupções, utilizando processamento em segundo plano. Essa abordagem permite que os processos de conversão de arquivos ocorram sem bloquear a interface gráfica, permitindo que o usuário continue interagindo com o sistema enquanto as tarefas são executadas. O uso de threads e a execução de funções em segundo plano ajudam a distribuir a carga de trabalho, evitando que o sistema trave ou apresente lentidão durante operações de conversão, o que é fundamental para melhorar a performance e a experiência geral do usuário (Graham et al., 2005).

Ao longo do desenvolvimento, percebeu-se a necessidade de implementar a funcionalidade de conversão de arquivos em lotes. Esse recurso foi adicionado para atender à demanda de usuários que precisam processar múltiplos documentos simultaneamente, seja em formato PDF, PNG, Word ou Excel. O processamento em lote oferece vantagens significativas em termos de eficiência, já que permite que vários arquivos sejam convertidos de uma só vez, reduzindo o tempo total necessário para concluir tarefas de conversão. Além disso, o processamento em lote também possibilita a automação de fluxos de trabalho, o que é um recurso valioso em contextos corporativos ou educacionais (Miller, 2007).

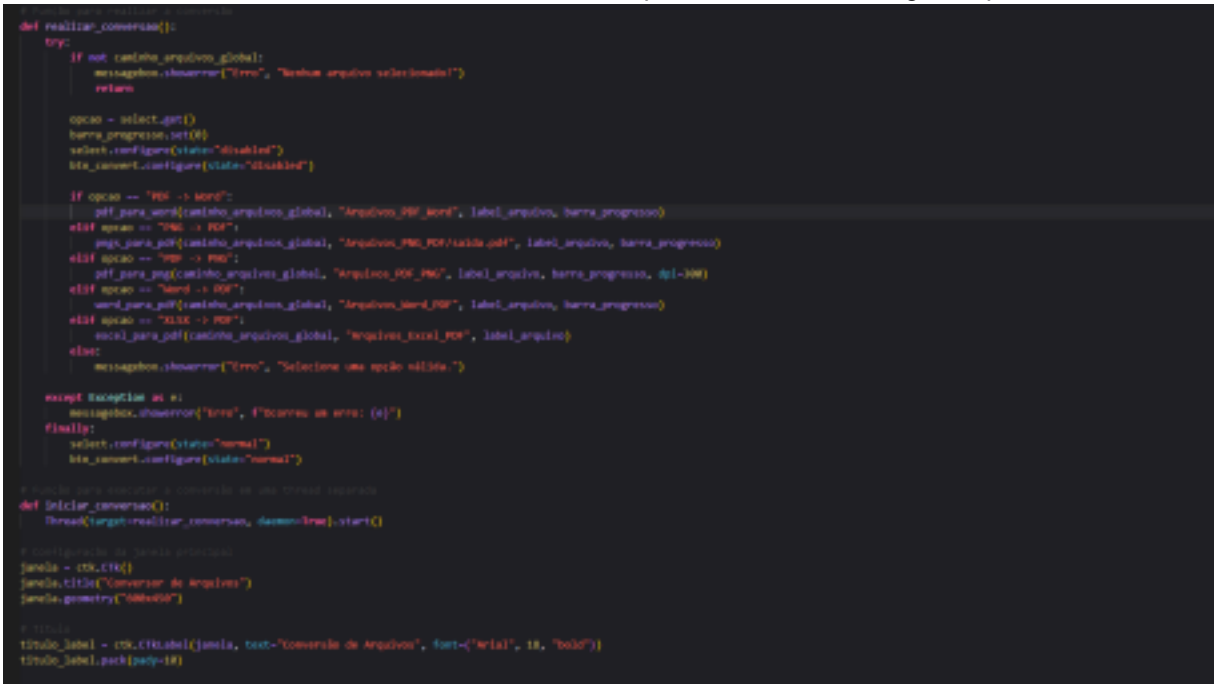
Essa funcionalidade de conversão em lote foi implementada em resposta à necessidade de otimizar o tempo de processamento e melhorar a usabilidade do sistema. Durante o projeto, percebeu-se que, sem essa funcionalidade, os usuários teriam que processar arquivos individualmente, o que poderia ser ineficiente, especialmente quando lidam com grandes volumes de documentos. A conversão em lote permite que o sistema realize várias tarefas de forma simultânea, aproveitando o poder de processamento do computador para aumentar a velocidade e reduzir a sobrecarga de trabalho do usuário. Além disso, a capacidade de gerenciar múltiplos arquivos simultaneamente também oferece uma experiência mais ágil e eficiente para os usuários que precisam lidar com muitos documentos em um único processo (White & McNabb, 2011). Podemos ver parte deste processo de produção e adaptação do sistema para tais fins na FIGURA 2.

FIGURA 2 Tela mostrando o processo de conversão de arquivos em lote na função de transformação de PDF para WORD



Fonte: Captura de tela feita pelo autor

FIGURA 3 Tela mostrando a funcionalidade de processamento em segundo plano



Fonte: Captura de tela efetuada pelo autor.

2. Conversão de PDF para Word

O sistema desenvolvido é capaz de converter arquivos PDF em documentos Word de maneira eficiente. Esta funcionalidade foi projetada para lidar tanto com PDFs que contêm texto como com aqueles baseados em imagens. No caso dos PDFs baseados em texto, o sistema extrai o conteúdo textual diretamente, garantindo que o documento convertido tenha a mesma formatação e clareza do original. Já para os PDFs baseados em imagens, o sistema utiliza a tecnologia OCR (Reconhecimento Óptico de Caracteres) por meio da biblioteca EasyOCR para realizar a extração de texto das imagens, o que possibilita a conversão de documentos digitalizados.

A ferramenta oferece um grande benefício, pois muitos documentos digitais ainda são salvos em PDF, dificultando a edição e a manipulação. A conversão para o formato Word facilita o trabalho com esses documentos, permitindo a modificação e personalização do conteúdo de forma simples. Além disso, o sistema permite que vários arquivos sejam processados simultaneamente, o que torna a tarefa mais ágil e menos propensa a erros humanos. A conversão é realizada de maneira sequencial, com feedback contínuo ao usuário, exibindo o progresso de cada arquivo à medida que é processado.

Essa funcionalidade é essencial em diversos cenários, como quando se precisa editar um relatório, um contrato ou qualquer outro documento PDF que não tenha sido originalmente criado em Word. A versatilidade do sistema também é um ponto positivo, pois ele pode ser utilizado em diferentes contextos, desde tarefas acadêmicas até processos empresariais.

FIGURA 4 Código da função de Conversão de PDF para WORD

Fonte: Captura de tela efetuada pelo autor.

FIGURA 5 Restante do código da função específica

Estas informações são armazenadas na máquina do cliente, em pastas organizadas para tais finalidades, além disso, ela gera pastas específicas para separar e organizar cada arquivo convertido separadamente.

3. Conversão de PNG para PDF

Outro recurso importante do sistema é a conversão de imagens PNG para o formato PDF. Este processo é útil quando se deseja reunir várias imagens em um único arquivo PDF, facilitando o armazenamento e a distribuição de documentos. Para isso, o sistema utiliza a biblioteca Pillow, que permite a manipulação de imagens e a conversão delas para o formato adequado. O procedimento é simples, pois o usuário seleciona as imagens desejadas e o sistema automaticamente cria o arquivo PDF contendo todas as imagens na ordem em que foram escolhidas.

A funcionalidade é bastante vantajosa para quem precisa gerar relatórios ou apresentações em PDF, agregando imagens como parte de documentos formais. A possibilidade de unir imagens em um único arquivo PDF elimina a necessidade de trabalhar com múltiplos arquivos e facilita o compartilhamento e o arquivamento das informações. Além disso, o sistema permite que as imagens sejam convertidas em alta resolução, garantindo que a qualidade visual seja preservada ao longo do processo de conversão.

O sistema também assegura que o processo de conversão seja realizado de forma rápida e sem a necessidade de softwares externos complexos. O usuário apenas escolhe as imagens e o caminho de destino para o PDF gerado. Isso torna o sistema altamente eficiente e fácil de usar, mesmo para aqueles que não possuem experiência em manipulação de arquivos PDF.

FIGURA 6 Componente do sistema para função

Fonte: Captura de tela efetuada pelo autor.

4. Conversão de PDF para PNG

A conversão de arquivos PDF para imagens PNG é outra funcionalidade implementada no sistema, sendo uma solução eficaz para aqueles que precisam extrair conteúdo visual de documentos PDF. Essa função é especialmente útil quando se deseja trabalhar com gráficos, tabelas, ou imagens contidas dentro de um arquivo PDF, mas não se deseja ou não se pode utilizar o arquivo original em PDF. O sistema usa a biblioteca PDF2Image para realizar essa conversão, permitindo que cada página do PDF seja transformada em uma imagem independente.

Esse recurso oferece uma maneira prática de criar imagens de alta qualidade a partir de documentos, o que é bastante útil para apresentações, envio de partes de um documento por e-mail, ou simplesmente para arquivar seções específicas de um conteúdo PDF. O sistema permite que o usuário defina a resolução das imagens, garantindo que a conversão atenda às suas necessidades específicas em termos de qualidade visual.

Além disso, a funcionalidade é muito eficiente, uma vez que o sistema consegue converter múltiplos arquivos PDF em várias imagens PNG de forma automática e rápida. Cada página do PDF é salva como um arquivo PNG individual, tornando a visualização e o compartilhamento do conteúdo muito mais práticos, especialmente para quem não quer lidar com o formato PDF diretamente.

5. Conversão de Word para PDF

Converter arquivos do Microsoft Word para PDF é uma funcionalidade essencial, especialmente para documentos que precisam ser distribuídos ou arquivados de forma segura e profissional. O sistema permite que arquivos do formato DOCX sejam transformados em PDFs, mantendo a formatação original do documento, como fontes, imagens e tabelas. Para realizar essa conversão, o sistema utiliza a automação do Microsoft Word por meio do COM (Component Object Model) no Windows, garantindo que a conversão seja feita com a mesma qualidade de uma conversão manual realizada diretamente no Word.

Este recurso é altamente valorizado em ambientes corporativos e acadêmicos, onde a distribuição de documentos em formato PDF é padrão. O PDF é preferido devido à sua capacidade de manter a formatação independentemente do sistema ou dispositivo utilizado para visualizá-lo. A conversão é realizada de maneira simples, onde o usuário escolhe o arquivo de Word e o diretório de saída para o PDF gerado. O processo é ágil e permite a conversão de múltiplos documentos de uma vez.

Além disso, o sistema lida automaticamente com qualquer ajuste necessário para garantir que o formato final seja adequado para impressão ou distribuição digital. A utilização desse recurso facilita enormemente a tarefa de preparar documentos finais para distribuição, sem a necessidade de manipulação adicional de formatação.

6. Conversão de Excel para PDF

A conversão de arquivos Excel para PDF também é uma funcionalidade fundamental do sistema, sendo ideal para quem precisa gerar relatórios e tabelas em formato PDF a partir de planilhas Excel. O sistema permite que os dados de uma planilha sejam convertidos de forma precisa, mantendo a estrutura e o layout da tabela. Para isso, o sistema utiliza a biblioteca Pandas para ler os arquivos Excel e a ferramenta pdfkit para gerar o PDF a partir de um arquivo HTML gerado a partir da planilha.

Essa funcionalidade é crucial em vários cenários, especialmente em ambientes empresariais e financeiros, onde relatórios periódicos precisam ser apresentados de forma organizada e formal. O formato PDF, com seu layout fixo, garante que o conteúdo do Excel seja visualizado da mesma forma em qualquer dispositivo, sem a possibilidade de modificação, o que aumenta a segurança e a integridade dos dados.

A conversão é realizada de forma automática, tornando o processo rápido e eficiente. O sistema permite que vários arquivos Excel sejam processados simultaneamente, economizando tempo e trabalho manual. Essa funcionalidade é particularmente vantajosa para profissionais que frequentemente precisam gerar documentos em PDF a partir de grandes volumes de dados armazenados em planilhas Excel.

5. VALIDAÇÃO

A análise deste projeto foi feita do ponto de vista de desenvolvimento, isto é, foram feitas avaliações por parte dos desenvolvedores em relação a criação do sistema e objetivos propostos.

Todas as funcionalidades propostas foram implementadas no sistema, com o uso e auxílio das as tecnologias mencionadas nos capítulos anteriores. As tecnologias usadas tiveram um bom desempenho para a construção do projeto, cumprindo corretamente com as funções pretendidas e gerando alguns bugs e erros a serem consertados ao longo do desenvolvimento. O que proporcionou maior fluidez no processo de construção do sistema.

1. Análise das tecnologias utilizadas

O sistema desenvolvido utiliza uma combinação de bibliotecas e ferramentas para realizar a conversão de arquivos de diferentes formatos, permitindo a automação de processos que normalmente exigiriam softwares específicos. A interface gráfica foi criada com a biblioteca CustomTkinter, uma versão moderna do Tkinter que suporta temas escuros e personalizados, proporcionando uma experiência mais agradável para o usuário. Além disso, o uso de multithreading garante que as conversões ocorram sem travamentos, permitindo que o usuário continue interagindo com o programa enquanto os arquivos são processados.

Para a conversão de documentos, foram utilizadas bibliotecas especializadas de acordo com cada formato. A conversão de PDFs para Word, por exemplo, envolve a extração de texto com PyMuPDF e, em casos de documentos baseados em imagem, o reconhecimento óptico de caracteres (OCR) é realizado com o EasyOCR, garantindo que textos embutidos em imagens também sejam extraídos corretamente. A conversão de arquivos PNG para PDF é feita com a biblioteca PIL, enquanto a transformação de PDFs em imagens PNG utiliza a pdf2image, que converte cada página em arquivos de imagem de alta qualidade. Já a conversão de arquivos do Microsoft Word para PDF é realizada por meio do COM do Windows, utilizando o win32com para interagir diretamente com o Microsoft Word, garantindo alta fidelidade no resultado final.

Além disso, a conversão de planilhas do Excel para PDF é feita com a biblioteca Pandas, que permite a leitura dos dados e a sua conversão para um formato HTML, posteriormente transformado em PDF utilizando o pdfkit. Essa abordagem permite preservar a estrutura da planilha e gerar um documento final organizado. O uso dessas tecnologias demonstra a flexibilidade e eficiência do sistema, tornando-o uma ferramenta útil para usuários que precisam converter
documentos de forma rápida e automatizada, sem depender de soluções pagas ou softwares complexos.

Por fim, a organização do código busca manter um padrão modular, separando as funções de conversão para cada tipo de arquivo e garantindo que novos formatos possam ser adicionados com facilidade. A interface gráfica foi projetada para ser intuitiva, permitindo que usuários sem conhecimento técnico possam realizar as conversões sem dificuldades. Com essa combinação de tecnologias, o sistema consegue oferecer um desempenho eficiente e uma experiência amigável, consolidando-se como uma solução prática para conversão de arquivos.

2. Análise geral

Foram observados inúmeros pontos positivos no sistema construído, que foi capaz de alcançar todos os objetivos propostos de desenvolvimento e de forma que todas as funcionalidades se encaixaram perfeitamente com a proposta de um sistema para conversão. Além de ter tomado um tamanho maior do que primeiramente planejado pelo desenvolvedor, foi analisado que o sistema foi finalizado com grandes possibilidades de mudanças e atualizações futuras, aprimorando-o cada vez mais.

No que toca às atualizações do sistema, consideramos também a forma com que o sistema foi construído, gerando uma maior facilidade na busca e correção de erros e bugs. Esta facilidade está ligada com as funcionalidades do

Python, no geral, devido a sua sintaxe mais simples para desenvolvimento e sua eficácia e versatilidade comprovada e debatida na subseção 2.3.1, isto significa que mesmo concluído, o sistema tem uma boa eficiência quando se trata de reparos, fazendo com que possam ser realizados com maior rapidez caso surjam bugs e erros.

Além de quesitos técnicos de programação, fez-se uma análise com foco no desempenho da conversão de arquivos, isto é, foi verificado a eficácia das funcionalidades para cada tipo de conversão disponível. Esta análise foi limitada pela falta da presença de usuários dispostos a testar o sistema e a sua eficiência. Porém, com base nas revisões de metodologias usadas no desenvolvimento, verificou-se que cada função responsável pela transformação de arquivos, de forma que possibilite concluir cada proposta do trabalho satisfatoriamente, possuem uma boa performance e cumprem com seus objetivos de converter arquivos e fixação de novos conhecimentos ao desenvolvedor enquanto produzia o projeto.

6. CONCLUSÃO

O projeto atingiu todos os objetivos propostos no projeto em questão de programação e desenvolvimento. Ao longo do projeto foi estudado e pesquisado como ocorre a mesclagem de algumas áreas, neste caso, a educação a
tecnologia. O que se fez a partir de estudos focados em prática de conhecimento, que foram aplicados no sistema a partir da construção de questões, cumprindo com um dos principais objetivo deste projeto, que era estudar a união da educação com a tecnologia.

Foi concluído que as tecnologias usadas apresentaram um comportamento adequado, cada uma fazendo seu papel dentro do sistema. Tanto o design quanto a programação atenderam às expectativas e propostas iniciais do projeto, contando com uma programação repleta de bibliotecas auxiliares, inúmeras funcionalidades implementadas, tanto complexas quanto simples. Nos quesitos que tocam ao design do projeto, desde o início foi proposto um design minimalista, com poucos detalhes, porém atrativo e intuitivo, além de algumas funcionalidades de design como modo claro e escuro. Características que foram implementadas no sistema com o uso de bibliotecas auxiliares disponibilizadas por comunidades de programação na Internet.

O uso de bibliotecas auxiliares, principalmente as utilizadas no backend, foram de suma importância no projeto, o uso de bibliotecas foi muito grande, algo inesperado para desenvolvedores em processo de realização do Trabalho de Conclusão de Curso, porém positivo. O uso destas bibliotecas fez com que inúmeras funcionalidades fossem implementadas fácil e rapidamente tanto tarefas simples, quanto complexas, o que facilitou o desenvolvimento da aplicação.

O sistema é completamente dinâmico, responsivo e local, isto é, as funcionalidades do projeto não necessitam de conexão com a internet para uso do sistema. Todo o processo de desenvolvimento foi pensando na experiência do usuário no sistema. O usuário poderá converter tipos de arquivos mais

famosos para os mais usuais que necessitamos no cotidiano, isto cumpre um dos objetivos iniciais do sistema de atender o maior número possível de pessoas que necessitam converter um de seus arquivos de maneira simplificada, aumentando a eficiência em tal tarefa. E, como um “efeito colateral” positivo ocorrido no trabalho, houve a comprovação de pontos deveras atuais e debatidos quanto ao uso de IA para desenvolvimento e aprendizado de novos conhecimentos e a criação, concomitantemente, de um sistema útil para os mais diversos tipos de usos com dados e informações em formato de arquivos.

Para implementações futuras do projeto, a principal ideia é uma testagem mais ampla do sistema, disponibilizando uma pesquisa com fins de reconhecer e analisar as opiniões e experiências dos usuários com o sistema a fim de que possa ser aprimorado futuramente. Este formulário será simples e objetivo, com perguntas descritivas para que o usuário possa relatar sua experiência de forma resumida com o objetivo de verificar a qualidade e efetividade das conversões de arquivos realizados pelo sistema na máquina do usuário. Com o resultado adquirido nestas testagens irão ser feitas as melhorias sugeridas, sempre com foco de agregar mais para o sistema.

REFERÊNCIAS BIBLIOGRÀFICA

LUTZ, Mark. **Learning Python. 5. ed.** O'Reilly Media, 2013.

VANDERPLAS, Jake. **Python Data Science Handbook: Essential Tools for Working with Data.** O'Reilly Media, 2016.

BISHOP, Christopher M. **Pattern Recognition and Machine Learning.** Springer, 2006.

GRANGER, Ben E.; HUNTER, Jason. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.** O'Reilly Media, 2014.

CHOLLET, François. **Deep Learning with Python.** Manning Publications, 2018.

KELLEHER, John D.; TIERNEY, Brian. **Data Science: An Introduction.** Springer, 2018.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement Learning: An Introduction.** MIT Press, 2018.

RUSSEL, S., & NORVIG, P. **Inteligência Artificial: Estruturas e Estratégias para a Solução Complexa de Problemas (3ª ed.)**. Pearson Prentice Hall, 2010.

SIEMENS, G. **Connectivism: A learning theory for the digital age.** International Journal of Instructional Technology and Distance Learning, 2(1), 3-10, 2005.

MAYER, R. E. **The Cambridge Handbook of Multimedia Learning.** Cambridge University Press, 2019.

MINSKY, M. **The Society of Mind.** Simon and Schuster, 1986.

DEWEY, J. **Experience and Education.** Macmillan, 1938.

SUSSMAN, G., & Wisdom, M. **Structure and Interpretation of Computer Programs.** MIT Press, 2001.

BRIAND, L. C., et al. **Automated Detection of Software Quality Issues Using AI.** Journal of Software Engineering, 22(4), 455-473, 2016.

GRAHAM, S. L., KNUTH, D. E., & PATERSON, M.. ***Concepts of Programming Languages*.** Addison-Wesley, 2005.

MILLER, R. ***Batch Processing: A Key to Efficiency in Large-Scale Systems*.** Journal of Information Technology Management, 18(3), 156-162, 2007.

WHITE, C., & MCNABB, D. ***Optimizing File Management Systems Through Batch Processing*.** Computer Science and Technology Journal, 12(4), 123-134, 2011.