



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LUIZ CARLOS DE LEMOS JÚNIOR

**GERAÇÃO AUTOMÁTICA DE SLIDES UTILIZANDO PYTHON
E INTELIGÊNCIA ARTIFICIAL GENERATIVA**

**CAMPINA GRANDE
2024**

LUIZ CARLOS DE LEMOS JÚNIOR

**GERAÇÃO AUTOMÁTICA DE SLIDES UTILIZANDO PYTHON
E INTELIGÊNCIA ARTIFICIAL GENERATIVA**

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Orientador: Profa. Dra. Kézia Vasconcelos de Oliveira Dantas.

**CAMPINA GRANDE
2024**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

L557g Lemos Junior, Luiz Carlos de.

Geração automática de slides utilizando Python e Inteligência Artificial generativa [manuscrito] / Luiz Carlos de Lemos Junior. - 2024.

63 p. : il. colorido.

Digitado. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024. "Orientação : Profa. Dra. Kézia Vasconcelos de Oliveira Dantas, Coordenação do Curso de Computação - CCT. "

1. Automação de slides. 2. Apresentações com Python. 3. IA generativa. I. Título

21. ed. CDD 028.5

LUIZ CARLOS DE LEMOS JÚNIOR

GERAÇÃO AUTOMÁTICA DE SLIDES UTILIZANDO PYTHON E INTELIGÊNCIA ARTIFICIAL GENERATIVA

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Aprovada em: 27 / Agosto / 2024.

BANCA EXAMINADORA

Kézia de Vasconcelos Oliveira Dantas

Profa. Dra. Kézia Vasconcelos de Oliveira Dantas (Orientadora)
Universidade Estadual da Paraíba (UEPB)

Kátia Elizabete Galdino

Profa. Dra. Kátia Elizabete Galdino
Universidade Estadual da Paraíba (UEPB)

Daniella Lima de Carvalho

Profa. Me. Daniella Lima de Carvalho
Universidade Estadual da Paraíba (UEPB)

Ao meu filho, pelo propósito de me tornar um homem melhor, uma inspiração e um pai, DEDICO.

AGRADECIMENTOS

A minha esposa Haysa, por sua presença ao meu lado, dando-me força.

Ao meu filho Leonardo, por sua alegria e inspiração.

Ao meu pai Luiz, a minha mãe Adjane, aos meus irmãos Luiziane e João, pelo apoio, motivação e solidez familiar.

Ao meu primo Gabriel, por sua ajuda em momentos difíceis nessa jornada.

À professora Kézia pelas leituras sugeridas ao longo dessa orientação, pela dedicação e pela confiança de ter me aceitado como seu aluno.

Aos meus amigos Hélder e Marcus, pela ajuda e motivação para seguir no curso.

Aos funcionários da UEPB pela presteza e atendimento quando nos foi necessário.

Aos colegas de classe pelos momentos de descontração e apoio.

“Quando algo é importante o suficiente, você faz, mesmo que as chances não estejam a seu favor.”

Elon Musk

RESUMO

Visando otimizar o processo de criação de apresentações, reduzindo o tempo e o esforço necessários para produzir conteúdo estruturado e informativo, este trabalho utiliza a aplicação de Python e inteligência artificial (IA) na automação de apresentações de slides. Embora existam algumas soluções desenvolvidas, elas se limitam à construção de uma apresentação a partir de um artigo estruturado, ou apresentam restrições de uso e limitação na quantidade de slides. Através da implementação de bibliotecas Python para a manipulação de slides e modelos de IA generativa para a criação de conteúdo textual e visual, este trabalho apresenta um sistema capaz de gerar apresentações de maneira eficiente. A IA generativa, em particular, é utilizada para criar textos e imagens, proporcionando uma experiência de criação mais dinâmica e adaptável. Por fim, temos um sistema de automação com Python combinado com IA generativa que acelera o processo de criação de slides, e garante consistência e qualidade no produto final.

Palavras-Chave: automação de slides; apresentações com Python; IA generativa.

ABSTRACT

Aiming to optimize the presentation creation process, reducing the time and effort required to produce structured and informative content, this work uses Python and artificial intelligence (AI) to automate slide presentations. Although there are some solutions developed, they are limited to building a presentation from a structured article, or have usage restrictions and limitations on the number of slides. By implementing Python libraries for manipulating slides and generative AI models for creating textual and visual content, this work presents a system capable of generating presentations efficiently. Generative AI, in particular, is used to create texts and images, providing a more dynamic and adaptable creation experience. Finally, we have an automation system with Python combined with generative AI that accelerates the slide creation process, and ensures consistency and quality in the final product.

Keywords: slide automation; Python presentations; generative AI.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Layouts para slides da biblioteca pptx	32
Figura 2 –	Exemplo de criação do primeiro slide	33
Figura 3 –	Princípio de funcionamento	36
Figura 4 –	Ambiente do código principal: passagem dos parâmetros de entrada; retorno da API do ChatGPT; criação da apresentação; adição dos elementos dos slides; finalização da apresentação	38
Figura 5 –	Apresentação com mais recursos do pptx	39
Figura 6 –	Ilustração da entrada de texto do automatizador	40
Figura 7 –	Apresentação seguindo arquivo de texto como entrada	41
Figura 8 –	Apresentação usando IA generativa de texto e imagem	42

LISTA DE QUADROS

Quadro 1 –	Ferramentas de extração de dados e criação de slides	26
Quadro 2 –	IAs que auxiliam na criação de textos e imagens	28
Quadro 3 –	Algoritmo do princípio de funcionamento do automatizador de slides com IA	37

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Justificativa	14
1.2	Objetivos	15
1.2.1	<i>Geral</i>	15
1.2.2	<i>Específicos</i>	15
1.3	Trabalhos Relacionados	15
1.4	Estrutura do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Automação de Slides	18
2.1.1	<i>Automação e Automatização de Processos</i>	20
2.2	IA Generativa	20
2.3	Tecnologias	22
2.3.1	<i>Python</i>	22
2.3.1.1	<i>Biblioteca python-pptx</i>	23
2.3.2	<i>ChatGPT</i>	24
2.3.2.1	<i>API do ChatGPT</i>	24
2.3.2.2	<i>API do DALL-E</i>	25
2.4	Ferramentas de Extração de Dados	25
3	METODOLOGIA	31
3.1	Levantamento de referências	31
3.2	Definição das Tecnologias	31
3.3	Implementação	33
3.3.1	<i>Geração Automática da Apresentação</i>	34
3.3.2	<i>Incorporação da API de IA</i>	34
3.3.2.1	<i>Adição da API do ChatGPT</i>	34
3.3.2.2	<i>Adição da API do DALL-E</i>	35
3.3.3	<i>Princípio de Funcionamento</i>	36
4	RESULTADOS E DISCUSSÕES	38
4.1	Apresentação com roteiro de entrada	39
4.2	Usando a IA para gerar o arquivo de entrada	41
4.3	Limitações	44

4.4	Comparativo com o Gamma App	44
5	CONCLUSÃO	46
	REFERÊNCIAS	47
	APÊNDICE A – PRIMEIRA VERSÃO DO AUTOMATIZADOR	49
	APÊNDICE B – ARQUIVO DE ENTRADA DO AUTOMATIZADOR	53
	APÊNDICE C – SEGUNDA VERSÃO DO AUTOMATIZADOR	55
	APÊNDICE D – FUNCOES.PY	59
	APÊNDICE E – CHAT.PY	63

1 INTRODUÇÃO

Atualmente, temos acesso a uma variedade de tecnologias que facilitam o acesso à informação em várias áreas da vida. Elas facilitam a descoberta de informações na internet e nos permitem acessar uma vasta quantidade de conteúdo, incluindo sites, blogs, fóruns, redes sociais, vídeos, e-books, cursos online entre outros. Tecnologias como Internet, motores de busca, redes sociais, aplicativos móveis, Inteligência Artificial e análise de dados estão moldando o nosso acesso à informação (TECH RADAR, 2023).

É verdade que, embora as tecnologias tenham facilitado o acesso à informação, elas também podem contribuir para tornar nosso tempo mais escasso. Fatores como sobrecarga de informação, distrações digitais, excesso de tarefas e alto consumo de entretenimento são armadilhas tecnológicas que podem nos levar a uma vasta quantidade de informações que se tornem difíceis de filtrar, de se concentrar e evitar a perda do nosso bem maior atualmente, o tempo (BRISTOW, 2024).

Para otimizar nossa produção e ajudar-nos a alcançar um nível de rendimento elevado, existem diferentes tipos de plataformas no mercado, as chamadas ferramentas de produtividade. São recursos tecnológicos voltados para priorizar tarefas importantes, evitar retrabalho e facilitar a comunicação (COUTO, 2024). Entretanto, a maioria dessas ferramentas são pagas, ou apresentam alta complexidade no seu aprendizado. Além de que, algumas dessas ferramentas ajudam apenas na formatação de conteúdo, como alinhamento, marcadores, estilo e tamanho da fonte, etc., mas não na preparação do conteúdo em si (APPUKUTTAN, 2016).

Para se tornar um pesquisador bem-sucedido, é preciso disseminar as principais descobertas da pesquisa. E para isso, um dos requisitos é a apresentação de slides, a maneira mais comum de apresentar o conteúdo do artigo (CAGLIERO, 2021). Seguindo essa mesma linha de raciocínio, Shreewastav *et al.* (2024), afirma que a comunicação em ambientes profissionais e educacionais depende de apresentações claras e bem estruturadas, e criá-las manualmente pode ser demorado, exigindo muitos recursos, dificultando a disseminação do conhecimento.

A integração da Inteligência Artificial na automação de tarefas oferece não apenas ganhos significativos em termos de eficiência e economia de tempo, mas também abre novas possibilidades para a inovação e a criatividade, especialmente quando se utiliza a IA generativa (AWARI, 2023). Ao permitir que as máquinas lidem

com tarefas repetitivas e intensivas em dados, profissionais e empresas podem se concentrar em atividades mais estratégicas, criativas e de alto valor, impulsionando a produtividade e melhorando a qualidade dos resultados finais. Dentre as diversas áreas existentes para otimização do trabalho, este trabalho tem por foco o desenvolvimento de um método que usa as tecnologias de IA generativa e automação de slides para nos auxiliar na padronização e otimização do tempo. Isto nos permite gerar uma apresentação mesmo que não saibamos nada sobre o tema, poupando-nos tempo de busca, diminuindo esforço e evitando retrabalho (AWARI, 2023).

Atualmente, existem ferramentas baseadas em IA que ajudam na produção de textos e imagens como:

- Jasper.ai - utilizando tecnologia de processamento de linguagem natural, é capaz de produzir textos em diversos formatos, como artigos, posts de blog, anúncios, descrições de produtos, e até scripts para vídeos, de maneira rápida e eficiente;
- Textio - voltada para a melhoria e otimização de linguagem em textos, especialmente em comunicações empresariais, como descrições de vagas de emprego, e-mails corporativos, anúncios e outros tipos de conteúdo escrito. Utiliza técnicas de processamento de linguagem natural e aprendizado de máquina para analisar, sugerir melhorias e prever o impacto que certas palavras ou frases podem ter no público-alvo;
- ChatGPT - projetado para gerar texto de maneira conversacional e interativa. Baseado na arquitetura GPT (*Generative Pre-trained Transformer*), utiliza técnicas de processamento de linguagem natural para entender e responder a perguntas, conduzir diálogos, fornecer informações, e até realizar tarefas criativas como escrever histórias ou gerar código;
- DALL-E - capaz de gerar imagens a partir de descrições textuais. O nome "DALL-E" é uma combinação de "Dalí" (em referência ao artista surrealista Salvador Dalí) e "WALL-E" (o robô do filme da Pixar), refletindo sua capacidade de criar imagens de maneira criativa e imaginativa com base em comandos de texto;
- Midjourney - criação de imagens a partir de descrições textuais, semelhante ao DALL-E. Tem ganhado popularidade entre artistas, designers e entusiastas criativos por sua capacidade de gerar arte digital de alta qualidade e estilo único com base em prompts fornecidos pelos usuários;

- Gamma App - permite a criação de apresentações e outros tipos de documentos visuais com a ajuda de inteligência artificial. Desenvolvido para simplificar e agilizar o processo de design de apresentações, o Gamma App oferece uma série de funcionalidades que visam melhorar a qualidade e a eficiência da criação de slides e documentos.

Essas ferramentas trabalham de modo isolado e não geram a apresentação em si. Entretanto, a Gamma App permite gerar uma apresentação baseada no tema proposto e com imagens, limitando a quantidade de uso diária e o número máximo de 10 slides. Nossa geração automática de slides com IA generativa apresenta comportamento similar, porém permite uma maior quantidade de slides, e possui estrutura baseada numa apresentação clássica organizada em títulos, tópicos, subtópicos e imagens.

1.1 Justificativa

A geração automática de slides é essencial para otimizar tempo e recursos, uma vez que elimina a necessidade de criar e formatar manualmente cada slide, permitindo que apresentações completas sejam desenvolvidas em uma fração do tempo. Buscamos não só acelerar o processo de criação, mas também reduzir os custos ao minimizar a demanda por trabalho manual, permitindo que equipes e indivíduos foquem em tarefas mais estratégicas.

Ao utilizar tecnologias como Inteligência Artificial associada à biblioteca de criação de slides do Python, podemos reduzir significativamente o tempo gasto na criação e formatação de apresentações, permitindo nos concentrar em tarefas mais estratégicas e de alto valor agregado.

A aplicação de recursos de IA generativa na automação de atividades tem ganhado destaque pela sua capacidade de transformar diversas áreas e processos. Isto é crucial para promover eficiência, inovação e personalização em massa, além de reduzir custos e melhorar a qualidade dos produtos e serviços oferecidos, e posicionar empresas e organizações à frente num mercado cada vez mais competitivo e dinâmico.

Com o uso de IA generativa, a automação não só otimiza o processo, mas também amplia o potencial criativo, permitindo a criação de apresentações

visualmente atraentes e dinâmicas, que atendem às necessidades específicas de cada contexto. Assim, podemos economizar tempo e recursos ao automatizar tarefas repetitivas, garantindo consistência visual em cada slide.

1.2 Objetivos

Utilizando as tecnologias atuais ao nosso favor, Inteligência Artificial (ChatGPT e DALL-E) e automação de slides, buscamos desenvolver uma maneira de criar e formatar apresentações de modo automático com o mínimo esforço e tempo gasto para o usuário.

1.2.1 Geral

- Desenvolver um método que associe o uso da IA generativa na criação de conteúdo, textos e imagens, com a geração automática de slides, permitindo o usuário construir uma apresentação padronizada de modo rápido.

1.2.2 Específicos

- Desenvolver uma aplicação inicial para montar rapidamente uma apresentação introduzindo apenas uma estrutura de texto baseada em tópicos;
- Modificar a aplicação desenvolvida para que a apresentação seja montada a partir do uso de IA geradora de textos;
- Incorporar a introdução de figuras na apresentação criada a partir da IA geradora de imagens.

1.3 Trabalhos Relacionados

A geração automática de slides está cada vez mais inteligente, já que muitas apresentações podem ser construídas com base em IA, por meio de aprendizado de máquina e mineração de dados. Embora a geração automática de slides para artigos acadêmicos venha crescendo nos últimos anos. Alguns estudos pesquisam diretamente o tópico de geração automática de slides, uns utilizam IA e outros não. Dos modelos estudados sem uso de IA temos:

Sravanthi *et al.* (2009) reforça a importância de se criar uma estrutura padrão para apresentação de slides, e para isso, usa artigos acadêmicos no formato LATEX como entrada estruturada para gerar uma saída intermediária em XML antes de gerar a apresentação final.

Hu e Wan (2015) não abordam uma geração automática de slides em si, mas ajudam no preparo da apresentação. A extração do conteúdo de artigos acadêmicos, funciona como um rascunho ou uma versão anterior a construção de uma apresentação final. Os slides gerados podem ser usados como rascunhos para ajudar os apresentadores a preparar seus slides formais de forma mais rápida.

Já para os artigos que usam IA na geração de apresentações e extrações de dados:

Kevin Shaj (SHAJ, 2020) aborda a importância de gerar automaticamente slides para ajudar na construção de apresentações mais interessantes para os espectadores, baseado na conversão de palavras chave em imagens a partir de artigos acadêmicos como entrada, usando o modelo modelo BERT (Bidirectional Encoder Representations from Transformers).

Tsu-Jui Fu, William Wang, Daniel McDuff e Yale & Song (FU *et al.*, 2022) apresentam um gerador de slides baseado num documento de entrada. Eles tratam textos e imagens, reconhecendo-os e diferenciando-os, depois realizando a estruturação dos slides da apresentação. O DOC2PPT converte um arquivo de texto no formato doc para uma saída de apresentação no formato ppt.

Shreewastav *et al.* (2024) apresenta uma nova abordagem para automatizar a geração de slides a partir de artigos acadêmicos. Após extraírem os dados textuais, utilizam a IA Gemini para estruturá-los, seguindo o uso do modelo T5 para sumarizar a apresentação e então gerar os slides por meio da biblioteca python-pptx em Python.

Observamos que a maioria dos trabalhos relacionados tem como foco artigos acadêmicos, e usam sua estruturação já padronizada para extrair informações e então compor os slides. A nossa proposta difere destes artigos por não usar estes padrões, mas criar uma apresentação a partir de um tema qualquer gerado por uma IA. Isto nos permite uma aplicação mais ampla fora do escopo da academia. Entretanto, os trabalhos se assemelham por apresentarem um arquivo de entrada para gerar apresentação.

Outro ponto em que diferimos é o uso das imagens: enquanto os demais trabalhos extraem imagens dos documentos a serem usadas nas apresentações, nós geramos imagens com uso de IA, levando a uma criação completa do documento final.

1.4 Estrutura do Trabalho

O restante deste trabalho é organizado da seguinte forma:

- Capítulo 2 – são apresentados o conceito de automação de processos, inteligência artificial generativa e as tecnologias utilizadas para o desenvolvimento de nossa ferramenta que possibilitou a aplicação de nosso método de associar tecnologias atuais para geração automática de slides.
- Capítulo 3 – é apresentada a metodologia utilizada para desenvolvimento do gerador automático de slides baseado em IA.
- Capítulo 4 – são apresentados os resultados das atividades desenvolvidas e discutidos seus significados e limitações.
- Capítulo 5 – são apresentadas as considerações finais sobre o trabalho desenvolvido e sugerido atividades futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Apresentamos neste capítulo os conceitos que serviram como base para desenvolvimento do método de geração automática de slides, bem como as tecnologias utilizadas para que esta aplicação se tornasse possível.

2.1 Automação de Slides

Segundo Souza (2023) criar um programa de automação de slides pode ser uma ótima maneira de economizar tempo e padronizar apresentações. Porém, precisamos seguir algumas etapas para desenvolver um programa como este, tais como:

1. Definir os requisitos: Antes de começar a escrever código, é importante entendermos as funcionalidades que vamos incluir no programa, como adicionar slides, texto e imagens e aplicar formatação.
2. Escolher a linguagem de programação: dentre várias linguagens de programação, devemos escolher a que nos ofereça melhor suporte às bibliotecas necessárias para interagir com o PowerPoint. No nosso caso, a linguagem Python.
3. Usar biblioteca apropriada: Para interagir com a apresentação, mais especificamente com o PowerPoint, precisamos de uma biblioteca que permita fazer isso. No nosso caso, a biblioteca `python-pptx` é uma escolha comum para uso no Python por fornecer uma vasta documentação de uso.
4. Desenvolver funcionalidades: Com a biblioteca escolhida, desenvolvemos as funcionalidades do programa como: criação de funções para adicionar slides, inserir texto e imagens e aplicar formatação.
5. Testar: À medida que desenvolvemos o programa, é importante testar cada funcionalidade para garantir que esteja funcionando conforme o esperado.

Ao seguir essas etapas, podemos criar um programa de automatização de slides em PowerPoint utilizando Python, que atenda às nossas necessidades e nos ajude a agilizar o processo de criação de apresentações. Dentre esses benefícios adquiridos, podemos citar:

1. **Eficiência e Economia de Tempo:** Automatizar a criação de slides nos permite economizar tempo significativo. Em vez de criarmos manualmente cada slide, o programa pode gerar slides com base em um conjunto de regras ou dados de entrada, acelerando o processo de criação.
2. **Padronização:** Um programa automatizado garante consistência e padronização nos slides. Isso é útil especialmente em organizações onde há necessidade de manter uma identidade visual consistente em todas as apresentações.
3. **Redução de Erros:** A automatização reduz o risco de erros humanos durante o processo de criação dos slides. Isso ajuda a garantir que as informações sejam precisas e completas.
4. **Escalabilidade:** Com um programa automatizado, podemos lidar com grandes volumes de slides de forma mais eficiente. Caso precisemos gerar muitas apresentações ou slides regularmente, a automatização pode lidar com isso de maneira mais escalável do que fazê-lo manualmente.
5. **Integração com Dados Externos:** Um programa automatizado pode se integrar a fontes de dados externas, como planilhas ou bancos de dados, para gerar slides dinâmicos e atualizados automaticamente.
6. **Facilidade de Atualização:** Se houver mudanças nos dados ou nas informações que precisam ser refletidas nos slides, um programa automatizado pode facilitar a atualização rápida e em grande escala.
7. **Otimização de Recursos Humanos:** Ao automatizar tarefas repetitivas, liberamos a equipe para se concentrar em atividades mais estratégicas e criativas.

Assim, um programa de automação de criação de slides pode ser uma ferramenta valiosa para aumentar a eficiência, reduzir erros e lidar com volume e complexidade em contextos onde a produção de apresentações é uma parte importante do trabalho.

Mas o termo correto seria automação ou automatização? Essa discussão é abordada a seguir.

2.1.1 Automação e Automatização de Processos

Automatização e automação são dois termos cada vez mais usados, principalmente no mundo da tecnologia voltada para negócios, contribuindo para que empresas possam executar seu trabalho com a ajuda dessas soluções. Ainda que esses dois termos sejam bem parecidos, o significado deles difere e deve ser entendido claramente pensarmos em usar essas ferramentas, já que a escolha incorreta pelo termo usado, não resultará no que precisamos (Souza, 2023).

A automatização é um termo relacionado a sistemas ou tecnologias que realizam ações específicas pré-programadas, mas que precisam da interferência de uma pessoa para que objetivos sejam alcançados completamente. É muito comum de ser encontrada em trabalhos que são normalmente muito repetitivos e podem ter a ajuda da tecnologia em sua execução, mas precisam de monitoramento, já que não conseguem analisar o trabalho que realizam.

A automação, por outro lado, se relaciona com tipos de inteligências que atuam em tarefas automatizadas, sendo uma tecnologia mais complexa e que consegue, por conta própria e sem a interação de humanos, tomar decisões e aprimorar seu trabalho com base no que ela mesma sabe fazer. Assim, a automação é capaz de identificar erros e corrigi-los por conta própria dentro do que se propõe a fazer.

Como nosso sistema possui um funcionamento independente, baseado em inteligência artificial e geração automática de slides, e não interferimos durante o processo para a criação dos slides, ele é automático. E para desenvolvimento do nosso sistema de geração automática de slides, a linguagem escolhida foi Python associada a textos e imagens gerados por IA generativa.

2.2 IA Generativa

Inteligência Artificial generativa se refere ao uso de IA para criar novos conteúdos, como texto, imagens, música, áudio e vídeos. Segundo o Google Cloud e AWS, IA generativa pode ser definida como:

Ela usa modelos de fundação (modelos de IA grandes) capazes de realizar várias tarefas ao mesmo tempo, além de resumos, perguntas e respostas, classificações e muito mais. Com o mínimo de treinamento necessário, os

modelos de fundação podem ser adaptados para casos de uso específicos com poucos dados de exemplo (Google Cloud, 2024).

As tecnologias de IA tentam imitar a inteligência humana em tarefas de computação não tradicionais, como reconhecimento de imagem, processamento de linguagem natural (PLN) e tradução. A IA generativa é a próxima etapa da inteligência artificial. É possível treiná-la para aprender linguagem humana, linguagens de programação, arte, química, biologia ou qualquer assunto complexo (AWS, 2024).

A IA generativa usa um modelo de aprendizado de máquina para reconhecer padrões e as relações em um conjunto de dados de conteúdo criado por humanos. Em seguida, ele usa os padrões aprendidos para gerar novo conteúdo. A maneira mais comum de treinar um modelo generativo de IA é usar o aprendizado supervisionado: o modelo recebe um conjunto de conteúdo criado por humanos e rótulos correspondentes. Em seguida, ele aprende a gerar conteúdo semelhante ao criado por humanos e rotulado com os mesmos rótulos (Google Cloud, 2024).

Aplicações de IA generativa, como o ChatGPT, chamaram a atenção e a imaginação de todos. Elas podem ajudar a reinventar a maioria das experiências e aplicações dos clientes, criar novas aplicações nunca antes vistas e ajudar os clientes a alcançar novos níveis de produtividade. De acordo com a Goldman Sachs, a IA generativa poderia impulsionar um aumento de 7% (ou quase 7 trilhões de dólares) no produto interno bruto (PIB) global. Também é previsto que isso poderia elevar o crescimento da produtividade em 1,5 pontos percentuais em 10 anos (AWS, 2024). A seguir, apresentamos mais alguns benefícios da IA generativa:

- Acelera a pesquisa - os algoritmos de IA generativa podem explorar e analisar dados complexos de novas maneiras. Assim, os pesquisadores podem descobrir novas tendências e padrões que, de outra forma, não seriam aparentes. Esses algoritmos podem resumir o conteúdo, delinear vários caminhos de solução, debater ideias e criar documentação detalhada a partir de notas de pesquisa. É por isso que a IA generativa aprimora drasticamente a pesquisa e a inovação;
- Melhora a experiência dos clientes - pode responder naturalmente à conversa humana e servir como uma ferramenta para atendimento ao cliente e personalização dos fluxos de trabalho do cliente;

- Otimiza os processos de negócios - a empresa pode otimizar os processos de negócios utilizando aplicações de machine learning (ML) e IA em todas as linhas de negócios. É possível aplicar a tecnologia em todas as linhas de negócios, incluindo engenharia, marketing, atendimento ao cliente, finanças e vendas;
- Aumenta a produtividade dos funcionários - podem aumentar os fluxos de trabalho dos funcionários e atuar como assistentes eficientes para todos em sua organização. Eles podem fazer tudo, desde a pesquisa até a criação, de forma humana.

2.3 Tecnologias

Nesta seção, apresentamos as tecnologias utilizadas para desenvolvimento da ferramenta geradora automática de slides. Nelas se enquadram a linguagem de programação escolhida, biblioteca utilizada e IAs utilizadas.

2.3.1 Python

Python é uma linguagem de programação de alto nível, interpretada, de propósito geral e de código aberto. Criada por Guido van Rossum e lançada pela primeira vez em 1991, Python é conhecida por sua sintaxe simples e legibilidade, o que a torna uma excelente escolha para iniciantes e programadores experientes (Python, 2024). Como principais características, podemos citar:

1. Legibilidade: Python enfatiza a legibilidade do código, com uma sintaxe que se assemelha ao pseudocódigo, o que facilita a compreensão do código por parte dos programadores;
2. Linguagem interpretada: Python é uma linguagem interpretada, o que significa que o código-fonte é executado linha por linha por um interpretador, sem a necessidade de compilação prévia. Isso torna o desenvolvimento em Python rápido e interativo;
3. Multiplataforma: Python é executado em várias plataformas, incluindo Windows, macOS e Linux, tornando-o uma escolha versátil para desenvolvimento de software;

4. Ampla comunidade e bibliotecas: Python possui uma comunidade ativa de desenvolvedores e uma vasta coleção de bibliotecas e frameworks disponíveis para uma variedade de fins, incluindo desenvolvimento web, ciência de dados, automação, inteligência artificial, entre outros. Dentre elas, destacamos a biblioteca `python-pptx` que utilizamos no desenvolvimento do nosso sistema;
5. Tipagem dinâmica: significa que você não precisa declarar explicitamente os tipos de variáveis ao escrever código. Isso torna Python flexível e permite uma escrita de código mais rápida.

Python é amplamente utilizado numa variedade de campos, incluindo desenvolvimento web (usando frameworks como Django e Flask), ciência de dados (com bibliotecas como NumPy, pandas e scikit-learn), automação de tarefas, desenvolvimento de jogos, desenvolvimento de aplicações desktop e muito mais. Sua popularidade continua a crescer devido à sua facilidade de aprendizado, poder e versatilidade. E devido a sua comunidade ativa, suporte e existência da biblioteca `python-pptx`, esta foi a nossa escolha para criação do sistema de automatização de slides.

2.3.1.1 *Biblioteca python-pptx*

A biblioteca `python-pptx` é uma ferramenta poderosa para criar e manipular apresentações do PowerPoint (.pptx) usando Python. Ela nos permite criar novas apresentações, modificar apresentações existentes e automatizar tarefas relacionadas à criação de slides (Python-PPTX, 2013). As principais funcionalidades oferecidas pela biblioteca `python-pptx` são:

1. Criação de apresentações: podemos criar uma nova apresentação ou modificar uma existente. Isso inclui adicionar e remover slides, alterar o layout dos slides, definir o título da apresentação e muito mais;
2. Manipulação de slides: podemos adicionar texto, imagens, gráficos, formas e outros elementos aos slides utilizando a biblioteca. Além disso, é possível aplicarmos formatação, como cores, fontes, tamanhos e alinhamentos de texto;
3. Controle fino de elementos: A `python-pptx` oferece controle detalhado sobre os elementos individuais dos slides, permitindo posicionar, redimensionar e estilizar cada elemento conforme necessário;

4. Manipulação de gráficos: A biblioteca permite a criação e edição de gráficos diretamente nos slides do PowerPoint, facilitando a visualização de dados complexos;
5. Geração dinâmica de apresentações: Combinando a `python-pptx` com outras bibliotecas Python para processamento de dados, como `pandas` ou `matplotlib`, é possível criar apresentações dinâmicas que se atualizam automaticamente com novos dados;
6. Exportação e salvamento: É possível exportar as apresentações como arquivos `.pptx` para uso posterior ou compartilhamento com outras pessoas.

Com a compreensão das funcionalidades da biblioteca, implementa-se o sistema de acordo com as ações desejadas para nossa aplicação. Associada a ela, ainda incorporamos uma aplicação de IA, no caso, o ChatGPT.

2.3.2 ChatGPT

O ChatGPT é um modelo de linguagem desenvolvido pela OpenAI, baseado na arquitetura GPT (*Generative Pre-trained Transformer*). Ele foi treinado em uma ampla variedade de textos coletados da internet, permitindo que ele gere respostas coerentes e contextuais com base nas entradas que recebe.

Dentre as características e capacidades do ChatGPT, as que nos interessa são a Geração de Texto Coerente, permitindo gerar respostas que fazem sentido; e Ampla Gama de Tópicos, já que foi treinado em uma grande quantidade de dados da internet, o ChatGPT possui conhecimento sobre uma ampla gama de tópicos, fornecendo informações sobre ciência, tecnologia, cultura, história, entre outros assuntos (OpenAI, 2024).

No geral, o ChatGPT representa um avanço significativo na capacidade das máquinas de entender e gerar texto de forma inteligente, e seu potencial de aplicação é vasto e diversificado. Aqui, vale destacar o funcionamento de sua API.

2.3.2.1 API do ChatGPT

A API do ChatGPT é projetada para ser fácil de usar e integrar em diferentes ambientes de desenvolvimento. Ela fornece documentação abrangente, exemplos de

código e SDKs (kits de desenvolvimento de software) em várias linguagens de programação para facilitar a integração (OpenAI API, 2023). Além disso, ela implementa medidas de segurança e privacidade para proteger os usuários e os dados fornecidos, isso inclui práticas de criptografia, controle de acesso e conformidade com regulamentações de proteção de dados, permitindo aos desenvolvedores personalizar o comportamento do modelo GPT para atender às necessidades específicas de seus aplicativos. Isso inclui ajustes na personalidade do modelo, tom da conversa, sensibilidade a certos tópicos, entre outros (OpenAI API, 2023).

Ainda que seja uma API paga, a API do ChatGPT da OpenAI foi selecionada como IA do sistema automatizado devido ao suporte fornecido para uso da aplicação, segurança e facilidade de integração. Embora não seja necessário utilizá-la, o usuário pode descrever sua entrada no sistema sem auxílio da API.

2.3.2.2 API do DALL-E

A API do DALL-E, fornecida pela OpenAI, permite que desenvolvedores integrem a geração de imagens com IA em seus aplicativos, websites e outros serviços. DALL-E é um modelo de inteligência artificial que gera imagens a partir de descrições textuais (OpenAI API, 2023).

No modelo de entrada de texto do usuário, ela pode ser utilizada ao utilizar a palavra imagem, levando o algoritmo a gerar uma imagem utilizando a API. No modelo automático, a inserção dessa imagem ficará a cargo da saída gerada pela API do ChatGPT.

2.4 Ferramentas de Extração de Dados

Foram pesquisadas algumas ferramentas que trabalham com extração de dados de slides e criação de apresentação e classificá-las quanto às vantagens e desvantagens. A mesma análise também foi feita para algumas IAs de interesse. As fontes usadas para este levantamento seguem nas referências deste documento, e o resultado está apresentado nos Quadros 1 e 2 que seguem:

Quadro 1 – Ferramentas de extração de dados e criação de slides

Ferramenta	Descrição	Vantagens	Desvantagens
Klipa DocHorizon	Uma plataforma para automatizar fluxos de trabalho relacionados com documentos.	<ul style="list-style-type: none"> - Digitalização móvel, OCR, extração de dados, classificação, conversão de documentos. - Forma automática. - API implementada em aplicação web ou móvel. 	<ul style="list-style-type: none"> - Pago. - Custo depende do volume de documentos. - Muitos recursos para apenas ser usado em extração de slides.
PDFelement	Conversor de imagem para palavras que você pode utilizar para extrair dados em PDF e imagens.	<ul style="list-style-type: none"> - Fácil utilização, e alta velocidade. - OCR e assinaturas digitais. - Total manipulação de PDF. 	<ul style="list-style-type: none"> - Apenas versões teste e paga. - Trabalha apenas com PDF.
PDFtables	Extração de dados de PDF converte com precisão para vários formatos como Excel, CSV, XML, e HTML.	<ul style="list-style-type: none"> - Fácil de usar e eficiente. - Transações são criptografadas de ponta a ponta. 	<ul style="list-style-type: none"> - Gratuito apenas para 50 conversões de páginas.
Adobe Acrobat	Pode converter, editar, comprimir, executar OCR, assinatura digital, e imprimir os seus arquivos PDF.	<ul style="list-style-type: none"> - Permite processo de conversão em série. - É rápido e confiável. 	<ul style="list-style-type: none"> - Programa de custo elevado. - Versão gratuita para apenas 1 teste. - Demora para abrir arquivos grandes (50 MB acima).
Foxit PhantomPDF	Oferece uma boa forma de gerar e gerir arquivos PDF.	<ul style="list-style-type: none"> - Criar, classificar, converter, compartilhar, proteger, e colaborar com arquivos PDF. 	<ul style="list-style-type: none"> - Versões teste e pagas. - A versão de teste tem funcionalidades

		<ul style="list-style-type: none"> - Tem várias funcionalidades avançadas, incluindo OCR. 	<ul style="list-style-type: none"> limitadas, como editar o PDF.
Nitro	<p>Ferramenta de conversão de PDF para Word que lhe permite extrair dados do PDF sem esforço. Permite importar e exportar dados de formulários com facilidade.</p>	<ul style="list-style-type: none"> - Vasta gama de formatos populares de arquivos. - Várias funções, incluindo OCR. - Processamento em série. 	<ul style="list-style-type: none"> - Quando carregado com arquivos pesados, pode ficar lento (acima de 50 MB) - Atualizações do sistema pioraram a usabilidade segundo usuários.
Think-Cell	<p>Software de criação de gráficos e layouts que automatiza seu trabalho no PowerPoint e melhora a eficiência e a qualidade na criação de slides.</p>	<ul style="list-style-type: none"> - Em minutos você terá slides com excelente aparência e layout correto. - Economia de tempo na criação de gráficos. - Interface simples. 	<ul style="list-style-type: none"> - Voltado para gráficos e layouts apenas. - Não contempla slides tradicionais. - Para apresentações simples (poucos gráficos, imagens e número de slides), se torna mais lento.
Aspose.Slides Parser	<p>Aplicativo online para extrair texto e imagens de documentos de apresentação.</p>	<ul style="list-style-type: none"> - Gratuito. - Trabalha com formatos de apresentação PPT, PPTX e ODP. - Baixar, salvar na nuvem ou enviar por e-mail. 	<ul style="list-style-type: none"> - Requer acesso à Internet. - Segurança: em nenhum momento é citada privacidade e não vazamento de dados.
Extractor App	<p>Extrair páginas do documento PowerPoint rápido e fácil.</p>	<ul style="list-style-type: none"> - Gratuito. - Rápido. - Seleciona quais páginas deseja extrair. 	<ul style="list-style-type: none"> - Requer conexão com Internet. - Não extrai os dados da apresentação.

		- Salva ou envia por e-mail.	- Segurança, não vaza os dados convertidos. - Apenas Splitter.
StudioX	Automatiza as tarefas mais comuns utilizadas em apresentações do PowerPoint.	- Não é intuitivo. - Acelera o processo de criação de apresentações longas. - Salva em diversos formatos como: PDF, CSV, PPT, PPTX e ODT.	- Depende do PowerPoint. - Um gerenciador na criação de apresentações, o usuário indica a sequência de execução de tarefas - Versões gratuita de 60 dias, após só assinando o serviço.

Fonte: Elaborada pelo autor, 2023.

Quadro 2 – IAs que auxiliam na criação de textos e imagens.

IA	Descrição	Vantagens	Desvantagens
Chat GPT	Avançada ferramenta de IA para geração de texto. Pode ser utilizada para criar textos, traduzir documentos e criar respostas para perguntas específicas.	- Fácil uso. - Documentação bem difundida.	- Recursos limitados na versão gratuita. - API versão paga.
Copy.ai	Criar títulos, descrições de produtos, slogans e outros tipos de conteúdo de marketing em segundos.	- Criatividade na criação de propagandas	- Voltado para o marketing. - API paga.
MarketMuse	Ajuda a criar conteúdo de alta qualidade e relevante para seus leitores. Pode descobrir quais tópicos devem ser abordados em seus artigos e como otimizar-los para mecanismos de busca.	- Facilita criação de artigos. - Indica temas relevantes.	- Busca de conteúdos existentes num site dada uma palavra-chave. - API paga.
Textio	Melhorar a redação de e-mails, anúncios e outros tipos de conteúdo. Ele analisa o	- Verificador de ortografia, coesão e coerência;	- Não cria texto. - API paga.

	tom e o estilo de escrita e sugere melhorias para torná-los mais eficazes.	- Melhora a escrita dos textos.	
Jasper.ai	Criar blogs, artigos, livros, roteiros e qualquer outro conteúdo. Automação de atendimento ao cliente e oferece soluções para responder perguntas frequentes, gerenciar tickets de suporte e até mesmo realizar vendas.	- Versátil na criação de textos.	- API paga.
Cloudinary	Plataforma de gerenciamento de imagens que utiliza tecnologias de IA para otimizar as suas imagens. Inclui compressão, corte, redimensionamento, detecção de faces e etiquetagem de imagens.	- Criação de imagens. - Manipulação de imagens.	- Não busca imagens. - API paga.
Imagga	Utiliza tecnologias de IA para detectar objetos, cores e emoções em imagens. É capaz de analisar imagens em tempo real e fornecer informações sobre ela, tais como sua qualidade, estética e conteúdo.	- Detecção de objetos e emoções. - Bom treinamento de reconhecimento de imagem.	- Não busca imagens. - API paga.
DeepArt	Utiliza redes neurais para transformar suas fotos em obras de arte. Pode criar pinturas que se assemelham a estilos de artistas famosos, como Van Gogh, Picasso e Monet.	- Designers na criação de imagens.	- Não busca imagens. - API paga.
Tineye	Busca de imagens que utiliza IA para encontrar outras semelhantes. Pode encontrar imagens de alta qualidade.	- Busca imagens semelhantes. - Achar imagens com melhor resolução.	- Busca imagens a partir de outras imagens. - API paga.
OpenCV	Biblioteca de código aberto que fornece recursos para	- Gratuito.	- Não busca imagens.

	processamento de imagens, incluindo detecção de bordas, detecção de objetos, segmentação de imagem, reconhecimento de padrões e reconhecimento facial.	- Processamento de imagens.	- Voltado apenas ao processamento de imagem.
DALL-E	Gera imagens a partir de descrições de texto. Capacidade de criar imagens realistas e criativas, abrindo novas perspectivas nos campos da arte, publicidade e design.	- Criatividade ilimitada. - Economia de tempo e esforço. - Gera imagens que não existem no mundo real.	- Inconsistência e interpretação, podendo retornar imagens diferentes das solicitadas. - Requer processadores velozes e consome muita memória. - Ética e propriedade intelectual.
Midjourney	Uma das IAs mais desenvolvidas na transformação da descrição de texto em imagens. Conhecido por suas imagens esteticamente agradáveis.	- Foco na estética. - Visualizar o processo de criação. - Qualidade da imagem.	- Versão de teste caótica, usada no chat do discord, deixando a leitura confusa - Acesso complicado. - Sem edição. - API paga.
Gamma App	Plataforma que oferece ferramentas inovadoras para criar apresentações e documentos interativos.	- Uso simples e intuitivo - Apresenta versão gratuita	- Versão gratuita limitada a 10 slides - Versão completa é paga

Fonte: Elaborada pelo autor, 2024.

A seguir, detalharemos a metodologia utilizada no processo de desenvolvimento do sistema.

3 METODOLOGIA

Para desenvolvermos nosso sistema de automatização de slides, seguimos algumas etapas: levantamento das referências; definição das tecnologias e implementação. Ao final destas etapas, obtivemos um programa funcional para um sistema de automatização de slides.

3.1 Levantamento de referências

Para estimar o comportamento do automatizador de apresentações, foi necessário avaliar algumas ferramentas existentes no mercado. Primeiramente, foram feitas pesquisas sobre essas aplicações, e uma vez entendido seu funcionamento, partimos para um tratamento dos dados dessas apresentações: extraindo e inserindo-os em slides.

Nesta etapa do projeto, foram pesquisadas ferramentas capazes de extrair dados de apresentações, criar apresentações e APIs de IA capazes de auxiliar o usuário nas criações de texto e imagens.

O objetivo neste momento foi observar a forma como essas ferramentas se comportavam com os dados dos slides das apresentações, e estimar como as IAs poderiam ser utilizadas para melhorar esse funcionamento.

3.2 Definição das tecnologias

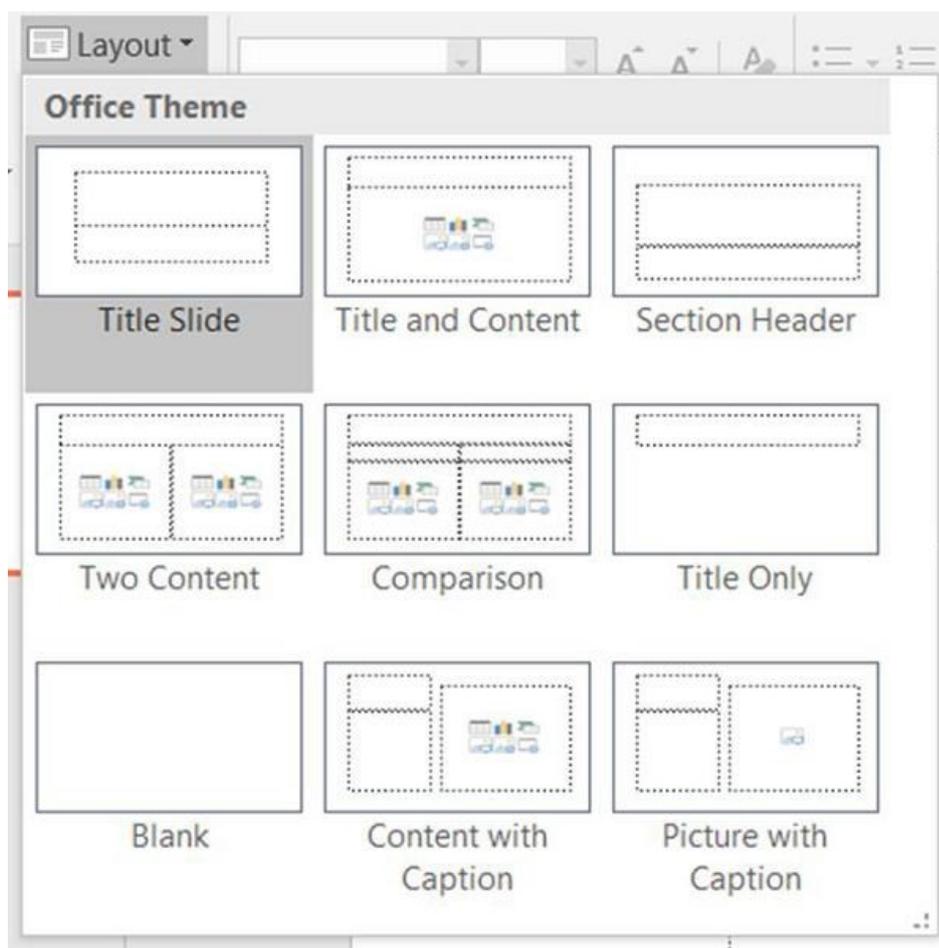
Por ser Python uma linguagem de programação amplamente utilizada por sua simplicidade, versatilidade e vasta comunidade de desenvolvedores, ela foi escolhida por ser conhecida por sua legibilidade e facilidade de aprendizado, o que a torna uma ótima escolha para uma variedade de aplicações, desde scripts simples até projetos complexos de desenvolvimento de software. Já o python-pptx é uma biblioteca Python que permite criar, manipular e automatizar apresentações PowerPoint (.pptx) de forma programática.

Nesta etapa do projeto, testou-se aplicações e replicando todo o processo de criação de slide passo a passo, desde a criação dos arquivos até a adição dos slides e tópicos. O objetivo, neste instante, era se familiarizar com a linguagem e a biblioteca, entendendo seu funcionamento.

As fases de construção de uma apresentação utilizando a biblioteca python-pptx consistem em:

1. Instalar o pacote com a biblioteca pptx caso o usuário ainda não a tenha;
2. Importar a classe Presentation a ser utilizada no programa com o comando da biblioteca pptx;
3. Criar o objeto da apresentação;
4. Criar e selecionar o layout do slide (a biblioteca apresenta 9 layouts predefinidos como pode ser visto na Figura 1);
5. Criar um slide objeto a ser adicionado o primeiro slide;
6. Nesse objeto adicionado, selecionar seus campos (baseado nos layouts) e adicionar o textos;
7. Salvar a apresentação.

Figura 1 – Layouts para slides da biblioteca pptx.



Fonte: BALAJI (2021)

Seguindo este passo a passo, é possível criar uma apresentação em PowerPoint. Porém, deve-se destacar que esse é um trabalho demorado, pois para cada slide tem-se inúmeras linhas de código, tal qual o exemplo que se pode observar na Figura 2.

Figura 2 – Exemplo de criação do primeiro slide.

```
# import Presentation class from pptx library
from pptx import Presentation

# Creating presentation object
root = Presentation()

# Creating slide layout
first_slide_layout = root.slide_layouts[0]

# Creating slide object to add in ppt i.e. Attaching slides with Presentation i.e. ppt
slide = root.slides.add_slide(first_slide_layout)

# Adding title and subtitle in slide i.e. first page of slide
slide.shapes.title.text = " Created By python-pptx"

# We have different formats of subtitles in ppts, for simple subtitle this method should
# implemented, you can change 0 to 1 for different design
slide.placeholders[1].text = " This is 2nd way"

# Saving file
root.save("Output.pptx")

print("done")
```

Fonte: Geek For Geeks (2020, com modificações).

A depender das funcionalidades que se queira introduzir nos slides, essas linhas de comando serão variadas, isso apenas para textos. Mas se quisermos introduzir gráficos, tabelas, imagens, outros comandos podem ser adicionados. Para maiores detalhes e outros exemplos, acessar o site *Geek For Geeks* (2020).

3.3 Implementação

Nesta etapa, montou-se um escopo do funcionamento do nosso sistema automatizador de apresentação de slides. Seguindo os passos apresentados no item 3.2, foi elaborada a primeira apresentação seguindo o exemplo de como criar um arquivo PowerPoint usando Python por BALAJI (2021).

Esta implementação se deu em três fases: criar uma apresentação completa utilizando comandos sequenciais; identificar os padrões de criação de slides e criar toda uma apresentação automaticamente; incorporar as IAs generativas ao processo

de criação automática. Esta versão final do automatizador é a que será tratada a seguir.

3.3.1 Geração Automática da Apresentação

Na primeira versão do automatizador, havia um script com algumas linhas de código mas que geravam apenas dois simples slides, a capa e um slide de texto. Por isso, o próximo passo foi incorporar comandos ao programa para que obtivéssemos uma apresentação com mais slides e maior conteúdo. O objetivo, nesta etapa, era mostrar que é possível construir toda uma apresentação utilizando a linguagem de programação, e em seguida, identificar os padrões para automatizar o processo de geração. Esse código está apresentado no APÊNDICE A no final deste documento.

3.3.2 Incorporação da API de IA

Uma vez que a versão mais recente do automatizador consegue criar a apresentação dada uma entrada padrão, foi adicionada a ele a API do ChatGPT. E então, a partir da geração automática dessa apresentação, adicionou-se a API do DALL-E para gerar imagens a serem utilizadas nos slides.

3.3.2.1 Adição da API do ChatGPT

No início do programa, serão coletados do usuário, por meio de pergunta e resposta via chat, o tema da apresentação e a quantidade de slides. Esses parâmetros são então passados via post usando a API, e a resposta obtida será o nosso roteiro de entrada para criação da apresentação. Esse arquivo usado como entrada apresenta o padrão de resposta do ChatGPT, e por isso o retorno da resposta da API coincide com o mesmo formato do arquivo.

Para usar a API, enviamos no cabeçalho da requisição a chave de autenticação (API KEY), e definimos o tipo de conteúdo, no nosso caso, uma aplicação json. No corpo da requisição, definimos a versão do modelo, gpt-3.5-turbo, a versão mais indicada para respostas mais rápidas utilizando chats via API na época de nosso desenvolvimento, e o link para o qual esta requisição deve ser encaminhada (<https://api.openai.com/v1/chat/completions>). E por fim, a mensagem que contém a

pergunta passada por parâmetro. A resposta da requisição obtida é então filtrada e salva num arquivo de formato.txt que será a entrada da formação automática dos slides. Esta rotina pode ser vista por completo no APÊNDICE E.

3.3.2.2 Adição da API do DALL-E

Com a API da IA implementada junto ao automatizador, tem-se a versão final do back-end de nosso sistema, que nos permite gerar uma apresentação automatizada em que podemos fornecer o roteiro, ou usar a IA para nos auxiliar. Neste caso, fornecemos o tema e ela retorna o arquivo de texto que será a entrada do nosso sistema. Entretanto, ainda não temos imagens na apresentação gerada, visto que a IA usada até o momento produz apenas textos.

O uso desta API é similar ao anterior, envia-se no cabeçalho da requisição a chave de autenticação (API KEY), e define-se o tipo de conteúdo, no nosso caso, uma aplicação json. Entretanto, no corpo da requisição, há alterações: definimos a versão do modelo, dall-e-3, o link para o qual esta requisição deve ser encaminhada (<https://api.openai.com/v1/images/generations>), a mensagem que contém o tema da imagem a ser gerada, o número de imagens como retorno (sempre usamos n = 1 na nossa aplicação) e sua resolução (1024x1024).

A resposta obtida é então filtrada para que se obtenha o link da imagem gerada, e esta é salva num arquivo .jpeg, e então convertida para um tamanho menor e adicionada ao slide. Este processo é demorado, e por isso não permite a presença de muitas imagens na apresentação.

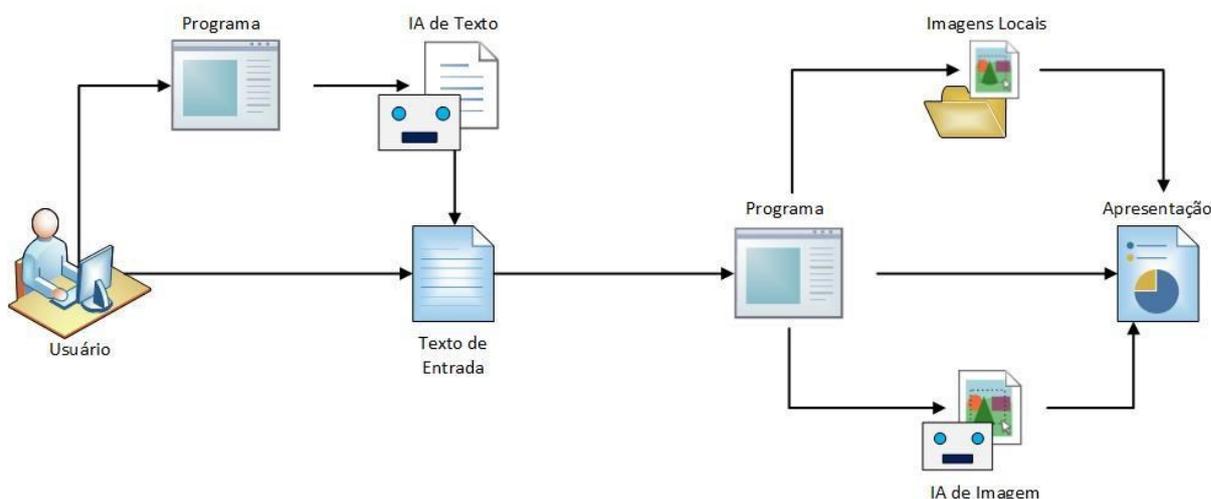
Incorporando a inteligência artificial geradora de imagens, DALL-E, solicita-se dentro da rotina de criação de slides a geração de figuras a serem adicionadas naquele ponto, fechando assim o ciclo para uma apresentação completa: tema, tópicos e figuras. Esta rotina pode ser vista por completo no APÊNDICE D.

3.3.3 Princípio de Funcionamento

Ao iniciarmos o programa, informa-se o tema e a quantidade de slides, de posse de nossas respostas, o automatizador questiona o ChatGPT por meio de sua API, e a resposta, no formato de texto, será tratada e usada na construção da apresentação. A saída da API geradora de texto será a entrada da criação dos slides, e dela é

derivada a geração de imagens a serem utilizadas na apresentação. Na Figura 3, podemos observar a sequência de funcionamento do programa desenvolvido.

Figura 3 – Princípio de funcionamento.



Fonte: Elaborada pelo autor, 2024.

Como pode ser observado na Figura 3, a utilização do programa para gerar apresentação automaticamente pode ser dada de duas formas: utilizando IA para gerar o arquivo de texto de entrada; ou próprio usuário entrando com o arquivo de texto de entrada.

Ao iniciar o programa, o usuário informa seu nome, o tema e a quantidade de slides, esses dados são passados como parâmetros para que a API do ChatGPT requisite a resposta textual. Nesse ponto, a resposta é salva como um arquivo txt, uma estrutura de apresentação resumida em tópicos.

Este arquivo é então percorrido pela rotina em Python para identificar a quantidade de slides, a criação de um vetor com os títulos e outro com os tópicos a serem usados. Após extração desses elementos, cria-se a apresentação, usando a biblioteca python-pptx, com a quantidade exata de slides, adicionando-se os títulos pela posição sequencial dos vetores, bem como os tópicos.

Caso haja a presença da palavra imagem, uma sob-rotina é chamada para buscar a imagem localmente, e se não existir, chama-se a API do DALL-E para criar a imagem dado o referido título do slide. Se não houver a presença desse marcador, todos os tópicos são tratados como texto.

Ao adicionar o último tópico no último slide, o laço se encerra e a apresentação é salva no formato .pptx. A implementação do sistema desenvolvido está representada no algoritmo a seguir:

Quadro 3 – Algoritmo do princípio de funcionamento do gerador de slides com IA

Entrada: autor, tema e quantidade de slides.	
Saída: apresentação em formato pptx.	
Processo:	
1.	Solicitar autor, tema e quantidade de slides.
2.	Requisitar ao ChatGPT uma apresentação passando os parâmetros de 1.
3.	Salvar a resposta num arquivo de texto.
4.	Percorrer o arquivo: <ol style="list-style-type: none"> a. identificando a quantidade de slides criada; b. construindo o vetor de títulos; c. construindo o vetor de tópicos.
5.	Criar o arquivo de apresentação.
6.	Criar e adicionar o slide.
7.	Montar o slide com seu título e seus tópicos.
8.	Se surgir a palavra imagem: <ol style="list-style-type: none"> a. buscar pela imagem localmente; b. se não existir localmente, solicitar ao DALL-E geração de imagem; c. adicionar imagem ao slide.
9.	Repetir os itens 6, 7 e 8 até o final dos vetores.
10.	Salvar a apresentação no formato .pptx.

Entretanto, há uma variação da execução padrão do sistema. O usuário pode criar o arquivo de texto, sem solicitar ao Chat GPT, ou mesmo alterar a saída pela IA informada, e assim, gerar uma apresentação personalizada, alterando títulos, tópicos e presença de imagens.

No capítulo a seguir, traremos alguns resultados e discussões do sistema desenvolvido.

4 RESULTADOS E DISCUSSÕES

A versão final do automatizador nos permite utilizar a IA ChatGPT e DALL-E para construir uma apresentação em alguns segundos. Na Figura 4, podemos observar o código final do automatizador, que faz uso dos arquivos de funções `funções.py` e `chat.py` (responsável pela chamada da API) localizada no APÊNDICE E.

Figura 4 – Ambiente do código principal: passagem dos parâmetros de entrada; retorno da API do ChatGPT; criação da apresentação; adição dos elementos dos slides; finalização da apresentação.

```
In [1]: import chat as gpt

In [2]: tema = input("Qual o tema da apresentação?")

In [3]: quantidade = input("Quantos slides deseja ter?")

In [4]: autor = input("Qual o nome do autor?")

In [5]: gpt.askGPT(tema, quantidade, autor)
<Response [200]>

In [6]: nome_arquivo = "info_presentation.txt"

In [7]: import funcoes as fun

In [8]: num_slides = fun.contar_slides(nome_arquivo)

In [9]: texto = fun.ler_arquivo(nome_arquivo)

In [10]: from pptx import Presentation
auto_apresentacao = Presentation()

In [11]: fun.criar_capa(auto_apresentacao, texto)
titulos = fun.coletar_titulos(texto)
topicos = fun.coletar_topicos(texto)
slides = fun.adicionar_slides(auto_apresentacao, texto, titulos, topicos, num_slides)

In [12]: auto_apresentacao.save("auto_apresentacao_IA.pptx")
```

Fonte: Elaborada pelo autor, 2023.

Tomemos uma apresentação completa, ainda que seja um simples exemplo, já foram apresentados textos em diversos níveis, imagens, gráficos, figuras em formas geométricas, tabelas e hyperlinks. Isto nos mostra os recursos que temos para criar apresentações usando a biblioteca `pptx`. O seu resultado pode ser observado na Figura 5.

Figura 5 – Apresentação com mais recursos do pptx.



Fonte: Elaborada pelo autor, 2023.

Esta foi uma aplicação simples, mostrando a possibilidade e os recursos para se criar uma apresentação automatizada. Este exemplo foi uma variação da aplicação, na qual o usuário entra com o um roteiro de entrada definido.

4.1 Apresentação com Roteiro de Entrada

O funcionamento do automatizador de apresentação é dado por um roteiro de texto em formato .txt de entrada, que será a base da padronização de nossos slides.

Com os dados informados por esse roteiro, o programa lê as linhas, e gera a apresentação necessária. Na Figura 6, podemos ver um trecho do roteiro utilizado, sua versão completa se encontra no APÊNDICE B.

Figura 6 – Ilustração da entrada de texto do automatizador.

Slide 1: Capa

- Título: Introdução à Inteligência Artificial
- Autor: Luiz Carlos de Lemos IA

Slide 2:

- Título: O que é Inteligência Artificial?
- Breve descrição da IA
- Exemplos de aplicações da IA:
 - Assistente virtual
 - Carros autônomos
 - Reconhecimento facial
- Citação sobre a importância da IA na atualidade

Slide 3:

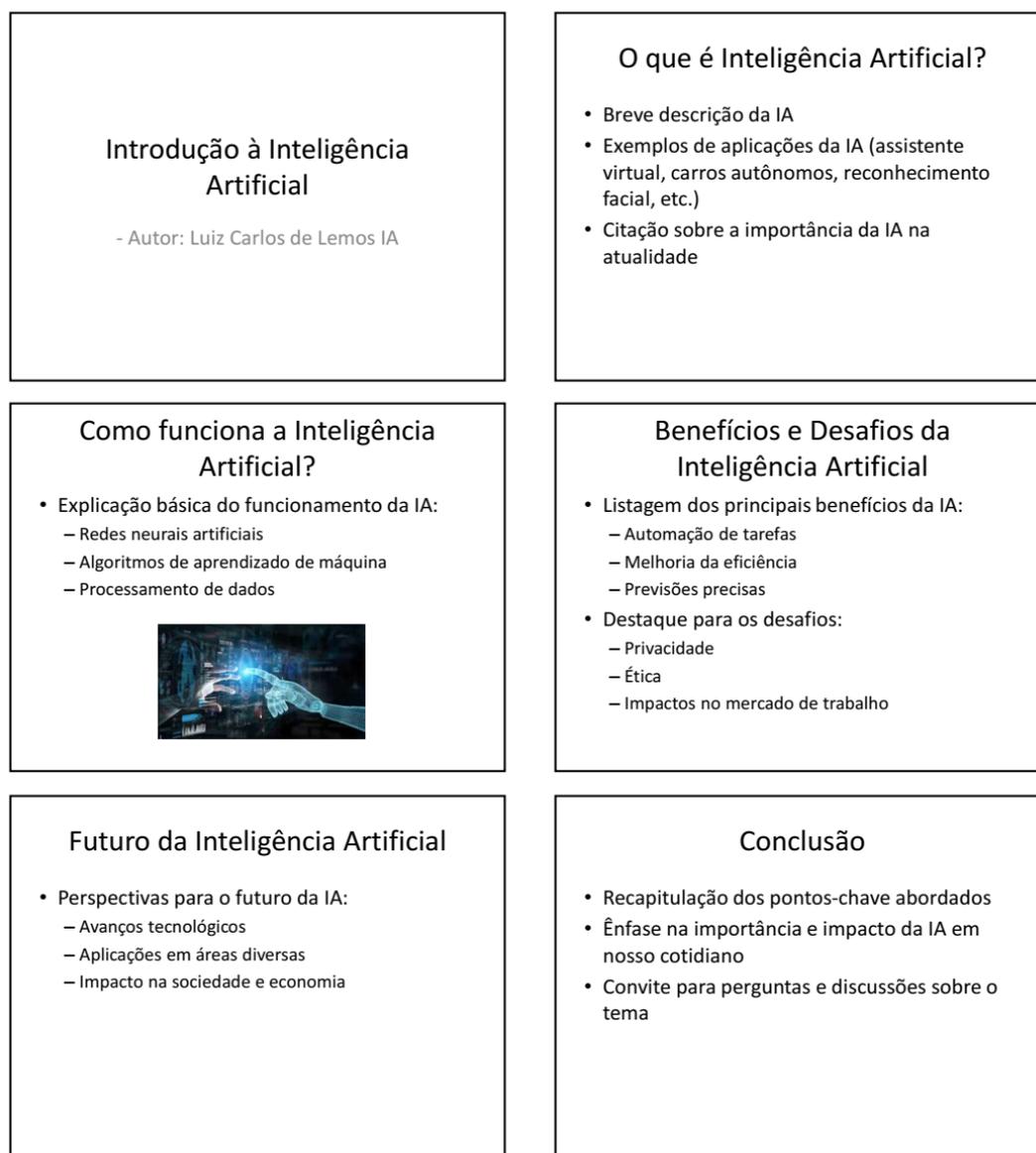
- Título: Como funciona a Inteligência Artificial?
- Explicação básica do funcionamento da IA:
 - Redes neurais artificiais
 - Algoritmos de aprendizado de máquina
 - Processamento de dados
- Imagem ilustrativa para auxiliar na explicação

Fonte: Elaborada pelo autor, 2023.

O automatizador conta a quantidade necessária de slides baseando-se no identificador de título de cada slide, a palavra “Slide”. Em seguida, preenche cada campo de texto inserindo linha a linha, tal qual o roteiro apresenta, e se houver indentação, ele reduz um nível no tópico do slide como podemos observar na Figura 7. O código completo utilizado para leitura do arquivo de texto e geração da apresentação, encontra-se no APÊNDICE C ao final deste relatório.

É possível ainda verificar na Figura 7, a presença de uma imagem nos slides. Isto foi possível porque o automatizador ao reconhecer a palavra no roteiro, chama uma rotina para inserção de imagem no slide. Esta rotina tem uma característica própria: primeiro, ela busca a imagem na pasta em que se encontra o roteiro, e caso não encontre, irá solicitar ao DALL-E a geração de uma imagem com o tema do slide em questão por meio do uso de sua API. E após introduzir a imagem, ajustamos sua posição no slide e tamanho.

Figura 7 – Apresentação seguindo arquivo de texto como entrada.



Fonte: Elaborada pelo autor, 2023.

Para os casos que não existam roteiros de entrada definidos, o usuário fará uso total das IA generativas, texto e imagem, nesse caso, ele apenas informa o autor, tema e a quantidade de slides desejada.

4.2 Usando a IA para gerar o arquivo de entrada

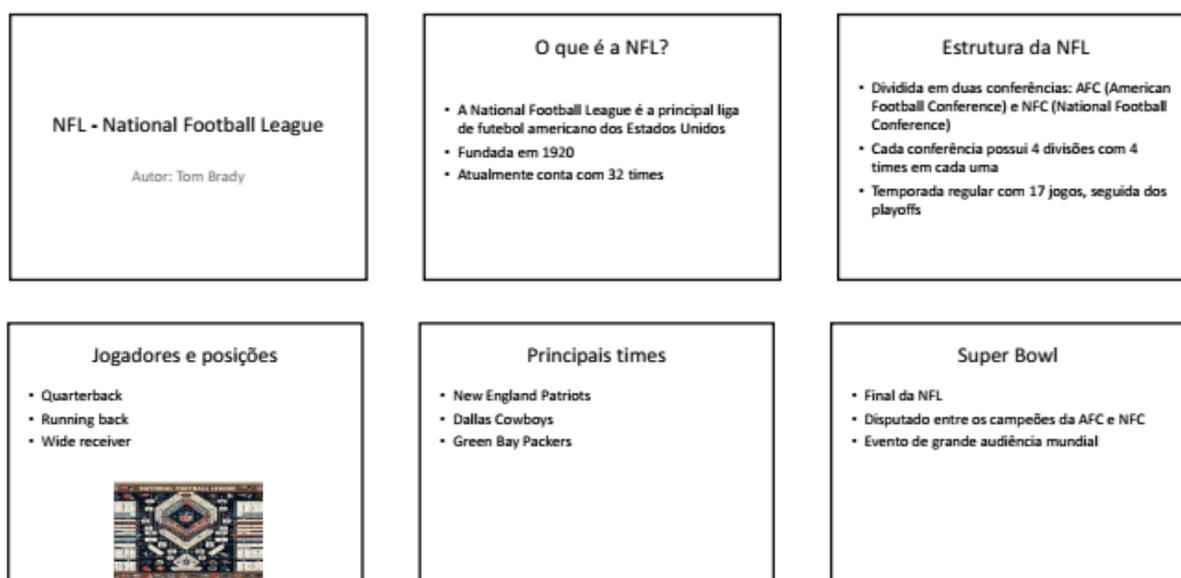
Nesse modo, o usuário ao executar o programa, apenas informa o tema da apresentação, o autor a ser inserido e a quantidade de slides desejada. Deste ponto em diante, tudo funciona de modo automático:

- O programa usará a API do ChatGPT para obter uma saída no formato de apresentação;
- Essa saída, no formato de texto, será tratada para usar a biblioteca do Python e então construir a apresentação com os dados obtidos;
- Se houver imagens na apresentação (derivadas da resposta da IA) a API do DALL-E será usada para gerar estas imagens e inseridas nos slides, caso não tenha, será gerado uma apresentação apenas com texto.

Por ser realizado em tempo real, há uma limitação das requisições para gerar imagens, a fim de que não trave o programa ou que exceda o tempo limite de execução devido a demora na geração das figuras.

Tomemos como exemplo uma aplicação sobre o tema NFL, com Tom Brady como autor e que contenha 19 slides. O automatizador além de gerar o slides automaticamente, também escolhe os títulos e tópicos. Também é gerado aleatoriamente, de acordo com a resposta do Chat GPT, a presença das imagens nos slides. Podemos observar o resultado dessa aplicação na Figura 8.

Figura 8 – Apresentação usando IA generativa de texto e imagem.



<p>Estádios icônicos</p> <ul style="list-style-type: none"> • Lambeau Field • Soldier Field 	<p>Recordes na NFL</p> <ul style="list-style-type: none"> • Peyton Manning - maior número de touchdowns • Jerry Rice - maior número de recepções • Adam Vinatieri - maior número de field goals 	<p>Pro Bowl</p> <ul style="list-style-type: none"> • Jogo das estrelas da NFL • Realizado após o fim da temporada regular • Confronto entre os melhores jogadores da AFC e NFC
<p>Draft da NFL</p> <ul style="list-style-type: none"> • Seleção de novos jogadores para os times • Realizado anualmente • Momento crucial para a renovação das equipes 	<p>Rivalidades históricas</p> <ul style="list-style-type: none"> • Cowboys vs. Redskins • Packers vs. Bears • Steelers vs. Ravens 	<p>Cheerleaders</p> <ul style="list-style-type: none"> • Presentes em muitos jogos da NFL • Animam a torcida • Parte tradicional das equipes
<p>Hall da Fama</p> <ul style="list-style-type: none"> • Os grandes ícones da NFL • Jogadores, treinadores e contribuintes • Local de homenagem aos melhores da história 	<p>Regras do jogo</p> <ul style="list-style-type: none"> • Touchdown • Field goal • Quarto down 	<p>Lesões na NFL</p> <ul style="list-style-type: none"> • Parte inevitável do esporte • Cuidados médicos para os jogadores • Preocupação constante da liga
<p>Celebridades na NFL</p> <ul style="list-style-type: none"> • Jay-Z • Beyoncé • Justin Timberlake 	<p>NFL e a cultura americana</p> <ul style="list-style-type: none"> • Evento esportivo de grande importância • Comerciais no Super Bowl • Influência na moda e música 	<p>Futuro da NFL</p> <ul style="list-style-type: none"> • Expansão para outros países • Novas tecnologias • Sustentabilidade e responsabilidade social
<p>Obrigado!</p> <ul style="list-style-type: none"> • Obrigado pela atenção • Qualquer dúvida, estamos à disposição • Tom Brady - autor da apresentação 		

Fonte: Elaborada pelo autor, 2023.

Caso o resultado da apresentação gerada pelo gerador automático não seja satisfatório ao usuário, este pode refazer sua solicitação, levando a uma nova geração total, ou apenas editando o arquivo de entrada, realizando pequenos ajustes nos slides antes de sua criação.

4.3 Limitações

Embora o automatizador de apresentações funcione normalmente para qualquer tema que a IA possa responder, o mesmo apresentou algumas falhas durante sua execução. Em geral, estas falhas ocorreram devido a interação com a IA do ChatGPT ao repetir a mesma solicitação.

O erro é decorrente da programação da IA do Chat, já que ele tenta conversar com o usuário, a cada vez que é repetida a mesma pergunta, ele responde diferente, tentando se adaptar ao diálogo. Então, ao questionar a IA sobre o tema da apresentação, a mesma pode retornar apresentações diferentes, e até mesmo gerar uma resposta fora do padrão previsto, levando a mal funcionamento do código, ainda que haja um tratamento de sua resposta para gerar a apresentação. Em algum momento a IA pode gerar uma resposta fora dos padrões.

Outra limitação é a quantidade de requisições de imagens geradas pelo DALL-E. As respostas das requisições, por vezes, demoravam bastante, e a depender do tema, esse tempo poderia ainda ser maior. Isto levou a alguns estouros de tempo limite na execução, ou a um mau funcionamento do algoritmo na composição dos slides. No algoritmo do programa, limitamos a 1/5 da quantidade de slides para que não haja travamentos. Chegamos a este valor após algumas tentativas na geração de apresentações em variados temas, se tornando um valor seguro.

4.4 Comparativo com Gamma App

O Gamma App é uma ferramenta baseada na web que permite aos usuários criar apresentações, documentos e páginas interativas de forma simples e intuitiva. Ele se destaca por utilizar a inteligência artificial para facilitar a criação de conteúdos visuais dinâmicos (Gamma App, 2024).

Tendo como vantagens sua facilidade do uso, eficiência e flexibilidade, todas essas características em conjunto permitem ao usuário montar uma apresentação sem muito conhecimento sobre design ou sobre o tema, e no final ainda ajustar conforme desejado.

O Gamma App ganhou um destaque dentre nossa pesquisa por ser a ferramenta que mais se aproxima da nossa aplicação desenvolvida: a criação de uma

apresentação utilizando ChatGPT para gerar texto, montar slides e adicionar figuras geradas por IA.

Há algumas semelhanças no nosso programa desenvolvido com o Gamma App, ambos trabalham utilizando API do ChatGPT para gerar o texto das apresentações segundo o tema fornecido pelo usuário, e adicionam imagens a estas. Entretanto, o Gamma App é um produto finalizado, portanto, é mais rebuscado e dispõe de recursos além do nosso automatizador, como seleção de temas e personalização de slides.

Porém, o nosso programa permite uma quantidade maior que 10 slides (quantidade máxima permitida pelo Gamma App gratuitamente), e focamos na apresentação clássica do PowerPoint com slides baseados em tópicos, estilo este não coberto pelo Gamma App.

Mas o ponto principal aqui, é mostrar que é possível construir uma ferramenta geradora de apresentação de slides, incorporando o uso de IA's generativas de texto e imagens, assim, podemos alterar as API's em questão e obter outras formas de gerar o arquivos final.

5 CONCLUSÃO

Geração automática de slides pode ser um método muito útil, especialmente para pessoas que fazem apresentações regularmente. Ele ajuda a economizar tempo, poupando-nos trabalho de pesquisa e confecção dos slides, permitindo que o apresentador se concentre mais no conteúdo e na interação com o público.

No entanto, é importante que a geração seja confiável e fácil de usar, para evitar interrupções ou problemas técnicos durante a apresentação.

Como os trabalhos observados tinham foco em artigos acadêmicos com estrutura padronizada para extração de dados e composição de slides, o nosso trabalho teve como objetivo criar uma apresentação a partir de um tema qualquer gerado por uma IA. Isto nos permitiu implementar uma aplicação mais ampla, não restrita a artigos acadêmicos.

Visando trazer facilidade de uso e eficiência ao usuário, implementamos um gerador automático capaz de produzir apresentações utilizando IA generativa de texto e imagem, permitindo não apenas criar toda uma apresentação instantaneamente, mas economizando tempos de busca sobre o tema e construção de tópicos. Definindo um tema e a quantidade de slides, podemos construir apresentações com conteúdo que não estavam previstos, mas foram trazidos devido a colaboração da IA, não ficando restritos ao conteúdo fornecido como entrada dos trabalhos relacionados.

O sistema apresenta algumas limitações, como a qualidade dos slides gerados que ainda pode ser aprimorada, especialmente em relação ao design dos slides. Para trabalhos futuros, é preciso melhorar a geração de imagens para que esta etapa seja mais rápida, evitando estouro de tempo do programa e permitindo adicionar mais imagens aos slides.

Com resultados promissores, o gerador automático demonstrou ser capaz de gerar slides coerentes com resultados que mostraram ser viável a abordagem proposta e pode ser aplicado em diferentes contextos, desde apresentações acadêmicas até corporativas, reduzindo significativamente o tempo gasto na criação de slides.

REFERÊNCIAS

APPUKUTTAN, Manju. (2016). **Automatic Slide Generation of Academic Paper**. Engineering, Technology and Applied Science Research. 4.

AWARI, 2023. Inteligência Artificial: Crie Slides de Forma Automática. Disponível em: <https://awari.com.br/inteligencia-artificial-crie-slides-de-forma-automatica/>. Acesso em: 10 jun 2024.

AWS, 2024. **O que é IA generativa?** Disponível em: <https://aws.amazon.com/pt/what-is/generative-ai/>. Acesso em 10 ago 2024.

BALAJI, Vijay 2021. **Tutorials Point**: How to create powerpoint files using Python. Disponível em: <https://www.tutorialspoint.com/how-to-create-powerpoint-files-using-python/>. Acesso em: 18 out 2023.

BRISTOW, Emily. RPA & IA: the automation tools reshaping business – What is RPA and why is it necessary for organizations? **Tech Radar**, 2024. Disponível em: <https://www.techradar.com/pro/rpa-and-ia-the-automation-tools-reshaping-business/>. Acesso em: 25 mai 2024.

CAGLIERO, Luca & LA QUATRA, Moreno. (2021). **Automatic slides generation in the absence of training data**. 103-108. 10.1109/COMPSAC51774.2021.00025.

COUTO, Marcela. 13 melhores ferramentas de produtividade do mercado. **Nuvemshop**, 2024. Disponível em: <https://www.nuvemshop.com.br/blog/ferramentas-de-produtividade/>. Acesso em: 10 jun 2024.

FU, Tsu-Jui & WANG, William & MCDUFF, Daniel & SONG, Yale. (2022). **DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents**. Proceedings of the AAAI Conference on Artificial Intelligence. 36. 634-642. 10.1609/aaai.v36i1.19943.

GAMMA APP, 2024. Gamma App Overview. Disponível em: <https://gamma.app/>. Acesso em: 09 jun 2024.

GEEKS FOR GEEKS, 2020. Creating and updating PowerPoint Presentations in Python using python-pptx. Disponível em: <https://www.geeksforgeeks.org/creating-and-updating-powerpoint-presentations-in-python-using-python-pptx/>. Acesso em: 12 out 2023.

GOOGLE CLOUD, 2024. **Exemplos de IA generativa**. Disponível em: <https://cloud.google.com/use-cases/generative-ai/>. Acesso em: 10 ago 2024.

HU, Yue & WAN, Xiaojun. (2015). **PPSGen: Learning-Based Presentation Slides Generation for Academic Papers**. Knowledge and Data Engineering, IEEE Transactions on. 27. 1085-1097. 10.1109/TKDE.2014.2359652.

OPENAI API, 2023. API Overview. Disponível em: <https://openai.com/product/>. Acesso em: 05 out 2023.

OPENAI. ChatGPT, 2024. ChatGPT Overview. Disponível em: <https://openai.com/chatgpt/>. Acesso em: 24 abr 2024.

PYTHON. **Python**, 2024. About Python. Disponível em: <https://www.python.org/about/>. Acesso em: 24 abr 2024.

PYTHON-PPTX, 2013. Python-pptx Release v0.6.22. Disponível em: <https://python-pptx.readthedocs.io/en/latest/>. Acesso em: 10 out 2023.

SHAJ, Kevin. (2020). **Learning Based Slide Generator**. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS070415.

SHREEWASTAV, Atul & ACHARYA, Bidhan & PAUDEL, Nischal & HUMAGAIN, Yugratna. (2024). **Presentify: Automated Presentation Slide Generation from Research Papers using NLP and Deep Learning** (May 2024). 10.36227/techrxiv.171561007.70987286/v1.

SOUZA, Kymberli. Qual a diferença entre automatização e automação? **Zeev**, 2023. Disponível em: <https://zeev.it/blog/qual-a-diferenca-entre-automatizacao-e-automacao/>. Acesso em: 24 abr 2024.

SRAVANTHI, M. & CHOWDARY, Ravindranath & KUMAR, P.. (2009). **SlidesGen: Automatic Generation of Presentation Slides for a Technical Paper Using Summarization**. Conference: Proceedings of the Twenty-Second International Florida Artificial Intelligence Research Society Conference, May 19-21, 2009, Sanibel Island, Florida, USA.

TECH RADAR, 2023. 6 ways to boost your career with smart tech: From clever timers to EE's new Smart Hub Plus, there are plenty of slick tricks to help supercharge your work. Disponível em: <https://www.techradar.com/computing/internet/6-ways-to-boost-your-career-with-smart-tech/>. Acesso em 25 mai 2024.

APÊNDICE A – PRIMEIRA VERSÃO DO AUTOMATIZADOR DE SLIDES UTILIZANDO DIVERSOS RECURSOS DA BIBLIOTECA PPTX

```
In [1]: from pptx import Presentation

In [2]: # create ppt presentation to add slides
pr1 = Presentation()

In [3]: # referes to layouts of each slide
# 0 = title slide; 1 = title and content; 3 = section header...
slide1_register = pr1.slide_layouts[0]

In [4]: # add initial slide to presentation
slide1 = pr1.slides.add_slide(slide1_register)

In [5]: # main top placeholder
title1 = slide1.shapes.title
# placeholder = item in layout
subtitle1 = slide1.placeholders[1]

In [6]: # insert text
title1.text = "CRIAÇÃO DE SLIDES USANDO IA"
subtitle1.text = "Luiz Carlos de Lemos"

In [7]: # add more slides - bullet points and a picture

In [8]: # slide 2
# create bullet point slide
slide2_register = pr1.slide_layouts[1]
slide2 = pr1.slides.add_slide(slide2_register)

In [9]: # edit bullet points
title2 = slide2.shapes.title
title2.text = "Testar Alguns Tópicos"

In [10]: bullet_point_box = slide2.shapes

In [11]: bullet_points_lvl1 = bullet_point_box.placeholders[1]
bullet_points_lvl1.text = "Nível 0"

In [12]: bullet_points_lvl2 = bullet_points_lvl1.text_frame.add_paragraph()
bullet_points_lvl2.text = "Nível 1"
bullet_points_lvl2.level = 1

In [13]: bullet_points_lvl2 = bullet_points_lvl1.text_frame.add_paragraph()
bullet_points_lvl2.text = "Nível 2"
bullet_points_lvl2.level = 2

In [14]: bullet_points_lvl2 = bullet_points_lvl1.text_frame.add_paragraph()
bullet_points_lvl2.text = "Nível 3"
bullet_points_lvl2.level = 3
```

```
In [15]: # for pictures
from pptx.util import Inches
```

```
In [16]: # slide 3
# create bullet point and picture slide
slide3_register = pr1.slide_layouts[5]
slide3 = pr1.slides.add_slide(slide3_register)
```

```
In [17]: # edit bullet point and picture slide
title3 = slide3.shapes.title
title3.text = "Incluir Uma Figura"
```

```
In [18]: import os
arquivo = r"C:\Users\Luiz Carlos\Documents\UEPB\Estágio\Criação\presentation
if os.path.isfile(arquivo):
    # add image
    img1 = "presentation.jpg"

    from_left = Inches(3)
    from_top = Inches(2)
    add_picture = slide3.shapes.add_picture(img1, from_left, from_top)
else:
    # edit bullet point and picture slide
    title3 = slide3.shapes.title
    title3.text = "Não Há Nenhuma Figura! :(""
```

```
In [19]: # for autoshapes
from pptx.enum.shapes import MSO_SHAPE
from pptx.enum.dml import MSO_THEME_COLOR
```

```
In [20]: # slide 4
# register/create slide
slide4_register = pr1.slide_layouts[5]
slide4 = pr1.slides.add_slide(slide4_register)
```

```
In [21]: # slide 4 title
title4 = slide4.shapes.title
title4.text = "Formatos e Formas"
```

```
In [22]: # create shapes
# shape 1
left1 = top1 = width1 = height1 = Inches(2)
add_shape1 = slide4.shapes.add_shape(MSO_SHAPE.ROUNDED_RECTANGLE, left1, top1,
```

```
In [23]: # shape 2
left2 = Inches(6)
top2 = Inches(2)
width2 = height2 = Inches(2)
arrow1 = slide4.shapes.add_shape(MSO_SHAPE.DOWN_ARROW, left2, top2, width2,
```

```
In [24]: # change arrow main color
fill_arrow1 = arrow1.fill
fill_arrow1.solid()
fill_arrow1.fore_color.theme_color = MSO_THEME_COLOR.ACCEPT_2
```

```
In [25]: # rotate shape
arrow1.rotation = 90
```

```
In [26]: # for graphs
from pptx.chart.data import CategoryChartData
from pptx.enum.chart import XL_CHART_TYPE
from pptx.enum.chart import XL_TICK_MARK
from pptx.util import Pt
```

```
In [27]: # slide 5
# register/create slide
slide5_register = pr1.slide_layouts[5]
slide5 = pr1.slides.add_slide(slide5_register)
```

```
In [28]: # slide 4 title
title5 = slide5.shapes.title
title5.text = "Gráficos"
```

```
In [29]: # build graph
graph_info = CategoryChartData()
graph_info.categories = ["Banana", "Laranja", "Uva"]
graph_info.add_series("Produção (t)", (15, 11, 18))
```

```
Out[29]: <pptx.chart.data.CategorySeriesData at 0x243a0a18730>
```

```
In [30]: # add graph to slide with positioning
left_graph = Inches(2)
top_graph = Inches(2)
width_graph = Inches(6)
height_graph = Inches(4)
graph1_frame = slide5.shapes.add_chart(XL_CHART_TYPE.COLUMN_CLUSTERED, left_
```

```
In [31]: graph1 = graph1_frame.chart
```

```
In [32]: # edit graph
category_axis = graph1.category_axis
category_axis.has_major_gridlines = True
category_axis.minor_tick_mark = XL_TICK_MARK.OUTSIDE
category_axis.tick_labels.font.italic = True
category_axis.tick_labels.font.italic = True
```

```
In [33]: # parte 5 - table
# slide 6
# register/create slide
slide6_register = pr1.slide_layouts[5]
slide6 = pr1.slides.add_slide(slide6_register)
```

```
In [34]: # slide 4 title
title6 = slide6.shapes.title
title6.text = "Tabela"
```

```
In [35]: # add table to slide
left_table = Inches(1)
top_table = Inches(2)
width_table = Inches(8)
height_table = Inches(3)
row_table = 3
column_table = 4
table1_frame = slide6.shapes.add_table(row_table, column_table, left_table,
```

```
In [36]: # populating a table
table1 = table1_frame.table
cell = table1.cell(0,0)
cell.text = "Aluno"
```

```
In [37]: cell = table1.cell(1,0)
cell.text = "José Maria"
```

```
In [38]: cell = table1.cell(0,1)
cell.text = "Nota"
```

```
In [39]: cell = table1.cell(1,1)
cell.text = "8.7"
```

```
In [40]: # parte 6 - hyperlinks
# slide 7
# register/create slide
slide7_register = pr1.slide_layouts[0]
slide7 = pr1.slides.add_slide(slide7_register)
```

```
In [41]: # slide 7 title
title7 = slide7.shapes.title
title7.text = "Hyperlinks"
```

```
In [42]: # adding hyperlinks
para1 = slide7.placeholders[1].text_frame.paragraphs[0]
addrun1 = para1.add_run()
addrun1.text = "Google"
hlink1 = addrun1.hyperlink
hlink1.address = "http://www.google.com"
```

```
In [43]: pr1.save("newpresentation.pptx")
```

APÊNDICE B – ARQUIVO DE ENTRADA DO AUTOMATIZADOR

Slide 1: Capa

- Título: Introdução à Inteligência Artificial
- Autor: Luiz Carlos de Lemos IA

Slide 2:

- Título: O que é Inteligência Artificial?
- Breve descrição da IA
- Exemplos de aplicações da IA:
 - Assistente virtual
 - Carros autônomos
 - Reconhecimento facial
- Citação sobre a importância da IA na atualidade

Slide 3:

- Título: Como funciona a Inteligência Artificial?
- Explicação básica do funcionamento da IA:
 - Redes neurais artificiais
 - Algoritmos de aprendizado de máquina
 - Processamento de dados
- Imagem ilustrativa para auxiliar na explicação

Slide 4:

- Título: Benefícios e Desafios da Inteligência Artificial
- Listagem dos principais benefícios da IA:
 - Automação de tarefas
 - Melhoria da eficiência
 - Previsões precisas
- Destaque para os desafios:
 - Privacidade
 - Ética
 - Impactos no mercado de trabalho

Slide 5:

- Título: Futuro da Inteligência Artificial
- Perspectivas para o futuro da IA:
 - Avanços tecnológicos
 - Aplicações em áreas diversas
 - Impacto na sociedade e economia

Slide 6:

- Título: Conclusão
- Recapitulação dos pontos-chave abordados
- Ênfase na importância e impacto da IA em nosso cotidiano
- Convite para perguntas e discussões sobre o tema

APÊNDICE C – SEGUNDA VERSÃO DO AUTOMATIZADOR DE SLIDES USANDO ARQUIVO DE TEXTO COMO ENTRADA

Lendo o arquivo de texto com roteiro

```
In [1]: with open("info.txt", "r", encoding="utf-8") as arquivo:
        texto = arquivo.read()
        print(texto)
```

Contabilizando Número de Slides

```
In [2]: num_slides = texto.count("Slide")
        print("Total de slides: ", num_slides)
```

Total de slides: 6

Lendo o Arquivo Linha a Linha

```
In [3]: with open("info.txt", "r", encoding="utf-8") as file:
        texto = file.readlines()
        print(texto)
```

Importando a Biblioteca de Criação do PPTX

```
In [4]: from pptx import Presentation
```

Criando a Apresentação

```
In [5]: apresentacao = Presentation()
```

Criando o Slide 1 - Capa

```
In [6]: # Escolhendo Layout do Slide 1 - Modelo Capa
        modelo_slide1 = apresentacao.slide_layouts[0]
        # Registrando Slide 1 na Apresentação
        slide1 = apresentacao.slides.add_slide(modelo_slide1)
        # Definindo Título e Subtítulo
        titulo1 = slide1.shapes.title
        subtitulo1 = slide1.placeholders[1]
        # Inserindo Textos no Título e Subtítulo
        titulo1.text = texto[1].rstrip("- Título: ").rstrip("\n")
        subtitulo1.text = texto[2].strip("\n")
```

Criando o Slide 2 - Tópicos

```
In [7]: # Escolhendo Layout do Slide 2 - Modelo Tópicos
modelo_slide2 = apresentacao.slide_layouts[1]
slide2 = apresentacao.slides.add_slide(modelo_slide2)
# Título do Slide
titulo2 = slide2.shapes.title
titulo2.text = texto[5].lstrip("- Título: ").rstrip("\n")
# Definindo Caixa dos Tópicos
caixa_de_topicos = slide2.shapes
# Inserindo Tópicos
caixa_de_topicos_1 = caixa_de_topicos.placeholders[1]
caixa_de_topicos_1.text = texto[6].strip("- "\n")
# Adicionando Parágrafos ao Tópico
caixa_de_topicos_2 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_2.text = texto[7].strip("- "\n")
caixa_de_topicos_3 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_3.text = texto[8].strip("- "\n")
```

Criando o Slide 3 - Subtópicos e Imagem

```
In [8]: # Escolhendo Layout do Slide 3 - Modelo Tópicos
modelo_slide3 = apresentacao.slide_layouts[1]
slide3 = apresentacao.slides.add_slide(modelo_slide3)
# Título do Slide
titulo3 = slide3.shapes.title
titulo3.text = texto[11].lstrip("- Título: ").rstrip("\n")
# Definindo Caixa dos Tópicos
caixa_de_topicos = slide3.shapes
# Inserindo Tópicos
caixa_de_topicos_1 = caixa_de_topicos.placeholders[1]
caixa_de_topicos_1.text = texto[12].strip("- "\n")
# Adicionando Subtópicos
caixa_de_topicos_2 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_2.text = texto[13].strip("- "\n")
caixa_de_topicos_2.level = 1
caixa_de_topicos_3 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_3.text = texto[14].strip("- "\n")
caixa_de_topicos_3.level = 1
caixa_de_topicos_4 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_4.text = texto[15].strip("- "\n")
caixa_de_topicos_4.level = 1
```

Adicionando Imagem

```
In [9]: # Importando Biblioteca Para Posicionar Imagem Inserida
from pptx.util import Inches
import os
imagem = r"C:\Users\Luiz Carlos\Documents\UEPB\Estágio\Slides\ia.jpg"
if os.path.isfile(imagem):
    # Adiciona Imagem
    img1 = "ia.jpg"
    esquerda = Inches(3)
    topo = Inches(4.5)
    adiciona_imagem = slide3.shapes.add_picture(img1, esquerda, topo)
else:
    # Adiciona Tópico
    caixa_de_topicos_5 = caixa_de_topicos_1.text_frame.add_paragraph()
    caixa_de_topicos_5.text = "\n * Não Há Nenhuma Imagem Para Adicionar **"
```

Criando Slide 4 - Diversos Níveis de Tópicos

```
In [10]: # Escolhendo Layout do Slide 4 - Modelo Tópicos
modelo_slide4 = apresentacao.slide_layouts[1]
slide4 = apresentacao.slides.add_slide(modelo_slide4)
# Título do Slide
titulo4 = slide4.shapes.title
titulo4.text = texto[19].lstrip("- Título: ").rstrip("\n")
# Definindo Caixa dos Tópicos
caixa_de_topicos = slide4.shapes
# Inserindo Tópicos
caixa_de_topicos_1 = caixa_de_topicos.placeholders[1]
caixa_de_topicos_1.text = texto[20].strip("- ""\n")
# Adicionando Subtópicos
caixa_de_topicos_2 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_2.text = texto[21].strip("- ""\n")
caixa_de_topicos_2.level = 1
caixa_de_topicos_3 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_3.text = texto[22].strip("- ""\n")
caixa_de_topicos_3.level = 1
caixa_de_topicos_4 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_4.text = texto[23].strip("- ""\n")
caixa_de_topicos_4.level = 1
# Inserindo Novo Tópico
caixa_de_topicos_5 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_5.text = texto[24].strip("- ""\n")
caixa_de_topicos_6 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_6.text = texto[25].strip("- ""\n")
caixa_de_topicos_6.level = 1
caixa_de_topicos_7 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_7.text = texto[26].strip("- ""\n")
caixa_de_topicos_7.level = 1
caixa_de_topicos_8 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_8.text = texto[27].strip("- ""\n")
caixa_de_topicos_8.level = 1
```

Criando Slide 5

```
In [11]: # Escolhendo Layout do Slide 5 - Modelo Tópicos
modelo_slide5 = apresentacao.slide_layouts[1]
slide5 = apresentacao.slides.add_slide(modelo_slide5)
# Título do Slide
titulo5 = slide5.shapes.title
titulo5.text = texto[30].lstrip("- Título: ").rstrip("\n")
# Definindo Caixa dos Tópicos
caixa_de_topicos = slide5.shapes
# Inserindo Tópicos
caixa_de_topicos_1 = caixa_de_topicos.placeholders[1]
caixa_de_topicos_1.text = texto[31].strip("- ""\n")
# Adicionando Subtópicos
caixa_de_topicos_2 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_2.text = texto[32].strip("- ""\n")
caixa_de_topicos_2.level = 1
caixa_de_topicos_3 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_3.text = texto[33].strip("- ""\n")
caixa_de_topicos_3.level = 1
caixa_de_topicos_4 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_4.text = texto[34].strip("- ""\n")
caixa_de_topicos_4.level = 1
```

```
In [12]: ## Criando Slide 6 - Conclusão
```

```
In [13]: # Escolhendo Layout do Slide 6 - Modelo Tópicos
modelo_slide6 = apresentacao.slide_layouts[1]
slide6 = apresentacao.slides.add_slide(modelo_slide6)
# Título do Slide
titulo6 = slide6.shapes.title
titulo6.text = texto[37].lstrip("- Título: ").rstrip("\n")
# Definindo Caixa dos Tópicos
caixa_de_topicos = slide6.shapes
# Inserindo Tópicos
caixa_de_topicos_1 = caixa_de_topicos.placeholders[1]
caixa_de_topicos_1.text = texto[38].strip("- ""\n")
# Adicionando Parágrafos ao Tópico
caixa_de_topicos_2 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_2.text = texto[39].strip("- ""\n")
caixa_de_topicos_3 = caixa_de_topicos_1.text_frame.add_paragraph()
caixa_de_topicos_3.text = texto[40].strip("- ""\n")
```

Salvando a Apresentação

```
In [14]: apresentacao.save("roteiro_apresentacao.pptx")
```

APÊNDICE D – FUNCOES.PY

```
from pptx.util import Inches
import os
import requests
import json
from idkey import id_key
import urllib.request
from PIL import Image
from senha import API_KEY

def contar_slides(nome_arquivo):
    with open(nome_arquivo, "r", encoding="utf-8") as arquivo:
        texto = arquivo.read()
        num_slides = texto.count("Slide")
    return num_slides

def ler_arquivo(nome_arquivo):
    with open(nome_arquivo, "r", encoding="utf-8") as arquivo:
        texto = arquivo.readlines()
    return texto

def criar_capa(auto_apresentacao, texto):
    modelo_slide = auto_apresentacao.slide_layouts[0]
    slide = auto_apresentacao.slides.add_slide(modelo_slide)
    titulo = slide.shapes.title
    subtitulo = slide.placeholders[1]
    if "Título" in texto[1]:
        titulo.text = texto[1].lstrip("- Título: ").rstrip("\n")
        subtitulo.text = texto[2].strip("\n")
    elif "Tema" in texto[1]:
        titulo.text = texto[1].lstrip("Tema: ").rstrip("\n")
        subtitulo.text = texto[2].strip("\n")
    else:
        titulo.text = texto[0].lstrip("Slide 0: ").rstrip("\n")
        subtitulo.text = texto[1].strip("\n")
    return slide

def coletar_titulos(texto):
    titulos = []
    for linha in texto:
```

```

        if "Título" in linha:
            titulos.append(linha.lstrip("- Título: ").rstrip("\n"))
    if titulos == []:
        for linha in texto:
            if "Slide" in linha:
                titulos.append(linha.lstrip("Slide ").rstrip("\n"))
    return titulos

def coletar_topicos(texto):
    topicos = []
    for topico in texto:
        if "- " in topico or "Imagem" in topico and "Título" not in topico
and "Autor:" not in topico or "\n" == topico:
            topicos.append(topico)
    topicos.append("\n")
    return topicos

def adicionar_slides(auto_apresentacao, texto, titulos, topicos,
num_slides):
    slides = []
    j = 1
    for i in range(num_slides - 1):
        modelo_slide = auto_apresentacao.slide_layouts[1]
        slide = auto_apresentacao.slides.add_slide(modelo_slide)
        titulo = slide.shapes.title
        titulo.text = titulos[i+1].lstrip("- Título: ").rstrip("\n")
        caixa_topicos = slide.shapes
        caixa_topico = caixa_topicos.placeholders[1]
        caixa_topico.text = topicos[j].strip("- ""\n")
        j += 1
        if topicos[0] == "\n":
            topicos.pop(0)
        while topicos[j] != "\n":
            if "  -" in topicos[j]:
                topico = caixa_topico.text_frame.add_paragraph()
                topico.text = topicos[j].strip("  -""\n")
                topico.level = 1
            elif "Imagem" in topicos[j]:
                imagem = r"C:\Users\Luiz
Carlos\Documents\UEPB\Estágio\Slides\ia.jpg"
                if os.path.isfile(imagem):

```

```

        img = "ia.jpg"
        esquerda = Inches(3)
        topo = Inches(4.5)
        adiciona_imagem = slide.shapes.add_picture(img,
esquerda, topo)
    else:
        #BUSCA NO UNSPLASH
        #busca = "beach"
        #requisicao =
requests.get(f"https://api.unsplash.com/search/photos?client_id={id_key}&qu
ery={busca}")

        headers = {"Authorization": f"Bearer {API_KEY}",
"Content-Type": "application/json"}
        id_modelo = "dall-e-3"
        link = "https://api.openai.com/v1/images/generations"

        body_message = {
            "model": id_modelo,
            "prompt": titulos[i],
            "n": 1,
            "size": "1024x1024",
        }

        body_message = json.dumps(body_message)
        requisicao = requests.post(link, headers=headers,
data=body_message)

        resposta = requisicao.json()
        #url_image = resposta['results'][0]['urls']['small']
        url_image = resposta['data'][0]['url']
        urllib.request.urlretrieve(url_image,
f"./imagem_url{i}.jpeg")

        ima = Image.open(f"imagem_url{i}.jpeg")
        ima = ima.resize((300, 200))
        ima.save(f"imagem_url{i}.jpeg")
        img = f"imagem_url{i}.jpeg"
        esquerda = Inches(3)
        topo = Inches(4.5)

```

```
        adiciona_imagem = slide.shapes.add_picture(img,
esquerda, topo)
    else:
        topico = caixa_topico.text_frame.add_paragraph()
        topico.text = topicos[j].strip("- ""\n")
        j += 1
    j += 1
    slides.append(slide)
return slides
```

APÊNDICE E – CHAT.PY

```

from senha import API_KEY
import requests
import json

def askGPT(tema, quantidade, autor):
    pergunta = f"Faça uma apresentação sobre {tema} com {quantidade} slides
incluindo capa. Na capa, identifique o título e autor com nome de {autor}.
Nos demais slides, além dos tópicos, identifique o título."

    headers = {"Authorization": f"Bearer {API_KEY}", "Content-Type":
"application/json"}
    id_modelo = "gpt-3.5-turbo"
    link = "https://api.openai.com/v1/chat/completions"

    body_message = {
        "model": id_modelo,
        "messages": [{"role": "user", "content": pergunta}],}

    # Transforma dicionário Python em Json
    body_message = json.dumps(body_message)

    requisicao = requests.post(link, headers=headers, data=body_message)
    print(requisicao)
    # print(requisicao.text)

    resposta = requisicao.json()
    # print(resposta)
    # print(resposta['choices'][0]['message']['content'])

    file = open("info_presentation.txt", "w", encoding = "utf-8")
    file.write(resposta['choices'][0]['message']['content'])
    file.close()

    with open("info_presentation.txt", "r", encoding="utf-8") as arquivo:
        texto = arquivo.readlines()

    texto.pop()
    # print(texto)

```