FACULDADE DE TECNOLOGIA DE AMERICANA “MINISTRO RALPH BIASI”

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CAIO MATHEUS TELES

LUCAS ALVES DA SILVA

Jambo: coleta de dados com Web scraping

AMERICANA

2021

CAIO MATHEUS TELES

LUCAS ALVES DA SILVA

JAMBO: COLETA DE DADOS COM WEB SCRAPING

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana sob a orientação do Prof. Esp. Antônio Alfredo Lacerda.

Área de concentração: Tecnologia da Informação.

AMERICANA

2021

Fatec Americana Ministro Ralph Biasi

Centro Paula Souza

SÃO PAULO GOVERNO DO ESTADO

Secretaria de Desenvolvimento Econômico

Faculdade de Tecnologia de Americana

CAIO MATHEUS TELES

LUCAS ALVES DA SILVA

JAMBO: COLETA DE DADOS COM WEB SCRAPING

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana sob a orientação do Prof. Esp. Antônio Alfredo Lacerda.

Área de concentração: Tecnologia da Informação.

Americana, 8 de Dez de 2021.

Banca Examinadora:

Prof. Esp. Antônio Alfredo Lacerda

Fatec Americana – Ministro Ralph Biasi

Prof. Me. Clerivaldo Jose Roccia

Fatec Americana – Ministro Ralph Biasi

Prof. Dr. Renato Kraide Soffner

Fatec Americana – Ministro Ralph Biasi

Dedicamos esse trabalho as nossas famílias e amigos que acreditaram em nós e nos deram forças para chegar até aqui. A jornada foi longa nesses 3 anos de faculdade e graças ao apoio de todos, tivemos o privilégio de aproveitar nossos estudos.

AGRADECIMENTOS

Ao Prof. Esp. Antônio Alfredo Lacerda, pela orientação, pelo tempo e apoio. Também agradecemos a todos os professores do curso de Análise e Desenvolvimento de Sistemas com quem tivemos a oportunidade e a honra de aprender e agregar valores em nossas vidas, pelo tempo, pela paciência, pela experiência e por todos seus ensinamentos durantes esses três longos anos.

RESUMO

Esse Projeto propõe uma ferramenta para auxiliar em pesquisas para fins estudantis, a ideia é retirar informações da internet de forma automática e trazer para o usuário através de uma tela amigável de maneira simples, direta e sem distrações. Foram utilizadas as práticas de engenharia de software além de frameworks e serviços integrados que foram inseridos no desenvolvimento e agilizaram a criação da ferramenta. O programa foi escrito na linguagem de programação Python utilizando uma técnica conhecida como *Web Scraping* e estabelece um algoritmo capaz de extrair dados e salvar informações em arquivos, além do mais, o programa possibilita o uso de um navegador integrado para pesquisas auxiliares. Para além da criação de um protótipo, é importante ressaltar questões éticas e legais sobre o impacto de uma ferramenta automatizada em servidores de terceiros, portanto, no final do trabalho foi descrito algumas questões importantes que foram plenamente seguidas a fim de evitar problemas relacionados a dados ou infraestrutura alheia, uma vez que, um script que coleta dados da web pode causar danos se criado de maneira incorreta.

Palavras-chave: Web Scraping; Python; Automática**.**

ABSTRACT

This Project proposes a tool to assist in research for student purposes, the idea is to automatically retrieve information from the internet and bring it to the user through a friendly screen in a simple, direct and without distractions. Software engineering practices were used, in addition to frameworks and integrated services that were included in the development and streamlined the creation of the tool. The program was written in the Python programming language using a technique known as Web Scraping and establishes an algorithm capable of extracting data and saving information in files, moreover, the program allows the use of an integrated browser for auxiliary searches. In addition for creating a prototype, it is important to highlight ethical and legal issues about the impact of an automated tool on third-party servers, so at the end of the work some important issues were described that were fully followed to avoid problems related to external data or infrastructure, as a script that collects data from the web can cause damage if created incorrectly.

Keywords: Web Scraping; Python; Automatic**.**

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

A proposta deste Trabalho de Conclusão de Curso tem como objetivo proporcionar uma visão de uma ferramenta capaz de auxiliar nas pesquisas estudantis do dia a dia. Vale ressaltar que em nenhum momento deste trabalho será apresentado um sistema com a finalidade de substituir, anular ou extinguir qualquer meio de pesquisa tradicional.

A ideia estabelece uma ferramenta simples em que um usuário possa inserir um tema em um campo de texto e o próprio software seja capaz de realizar buscas na internet, trazendo informações relevantes. Podemos imaginar neste contexto um robô coletando dados de maneira automática.

O Capítulo 2 começa abordando as práticas e técnicas relacionadas a engenharia de software. Nesta seção é possível ter a visão do que se trata o trabalho por meio dos diagramas e ideias estabelecidas conforme os requisitos desenvolvidos.

O Capítulo 3 fornece uma visão sobre o desenvolvimento do Projeto no qual é de extrema importância conhecermos os recursos necessários que foram a base para a criação de um protótipo funcional.

O Capítulo 4 aborda algumas questões essenciais de planejamento e gerenciamento de Projeto. Começa descrevendo a estrutura analítica do Projeto (EAP), o planejamento das funcionalidades com a metodologia Kanban e mostrará as interfaces propostas para o trabalho desenvolvido.

O Capítulo 5 tenta proporcionar uma reflexão sobre algumas questões éticas em relação a empregabilidade de uma ferramenta ou qualquer software que trabalha com um algoritmo automatizado extraindo informações da internet.

O Capítulo 6 trata-se de uma conclusão abstraída do trabalho que estabelece alguns pensamentos e informações finais decorrentes do Projeto como um todo.

2 ENGENHARIA DE SOFTWARE

Assim como outras engenharias, na área de TI a engenharia de software é responsável pela especificação, desenvolvimento e todo a etapa relacionada a criação de software, utilizando por exemplo práticas de gerenciamento e organização, visando a produtividade e qualidade de um sistema (WIKIPEDIA, 2021).

A engenharia de software envolve diversos processos que facilitam a criação de sistemas robustos e eficazes garantindo a utilização de boas práticas e direcionamento de um projeto de tal forma que possibilita que a entrega do produto seja exatamente aquilo que foi pensado inicialmente. Nas próximas seções, serão apresentadas as etapas de engenharia de software que foram utilizadas no trabalho.

2.1 LEVANTAMENTO DE REQUISITOS

É uma das partes que engloba o entendimento sobre aquilo que um sistema pretende realizar. De um modo geral, podemos dizer que é um processo de compreensão e identificação das necessidades de um usuário para com o software proposto ou em outras palavras, fornece as etapas necessárias para definir o que será desenvolvido.

2.1.1 Requisitos funcionais

Os requisitos funcionais são os requisitos que descrevem o comportamento de uma aplicação, ou seja, define o que o sistema deve fazer. Este tipo de requisito estabelece explicitamente as funcionalidades de um software e deve ser o mais objetivo possível para garantir que o desenvolvimento esteja de acordo com as necessidades do usuário. Dessa forma, tendo como base a definição apresentada, o Quadro 1 indicará os requisitos funcionais para o Projeto.

Quadro 1 - Requisitos funcionais do projeto

Identificação	Requisito Funcional	Prioridade
RF01	Fazer busca de dados	Essencial
RF02	Filtrar dados	Essencial
RF03	Carregar elementos e exibi-los previamente	Essencial
RF04	Deve fornecer links de pesquisas	Importante
RF05	Deve possibilitar salvar os dados pesquisados	Importante
RF06	Deve possibilitar salvar links	Importante
RF07	Deve fornecer uma alternativa de navegador	Importante

Fonte: Elaborado pelo autor.

2.1.2 Requisitos não funcionais

Os requisitos não funcionais são requisitos que não interferem nas regras de negócio e não influenciam diretamente nas funcionalidades específicas de um sistema, ou seja, basicamente não descreve o que o sistema fará, mas sim, como fará. Podem ser chamados de atributos de qualidade pois servem de escolha para alternativas ao projeto em relação a arquitetura do sistema e formas de implementação. Dessa forma, tendo como base a definição descrita, o Quadro 2 apresentará os requisitos não funcionais para o Projeto.

Quadro 2 - Requisitos não funcionais do projeto

Identificação	Requisito Não Funcional	Prioridade
RNF01	O sistema deverá ser desenvolvido na linguagem de programação Python	Essencial
RNF02	Todo o processamento e pesquisas não devem ser superiores a 40 segundos	Essencial
RNF03	O sistema somente deverá ser funcional quando houver conexão com a internet	Essencial
RNF04	Toda informação pesquisada deve ser armazenada em arquivos	Essencial
RNF05	Deve ser compatível com os sistemas operacionais Windows ou Linux	Desejável

Fonte: Elaborado pelo autor.

2.2 MODELAGEM

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language (*UML) para modelar o diagrama de caso de uso e o diagrama de estados, foi definido também um fluxograma básico na tentativa de demonstrar o mecanismo de pesquisa. Esses 2 diagramas assim como o fluxograma foram essenciais para criação e desenvolvimento do Projeto e sua lógica geral, uma vez que projetam a ideia e o funcionamento do sistema.

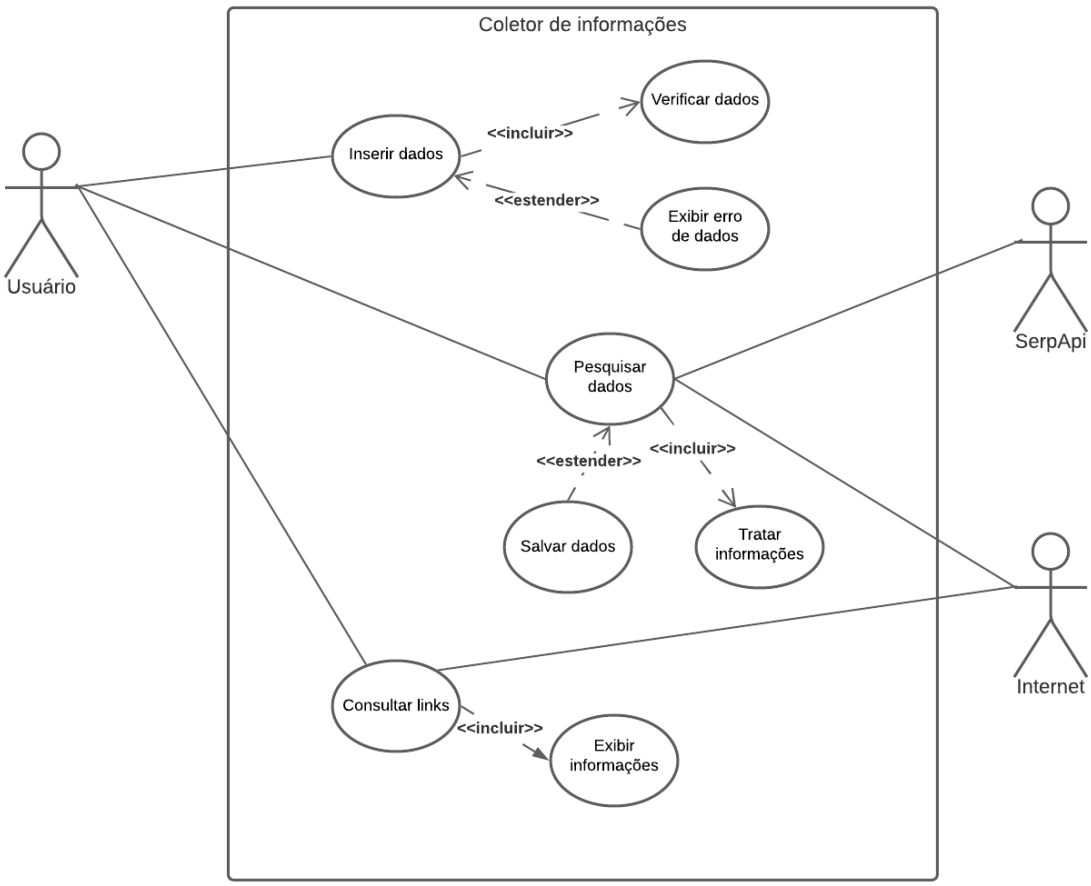
2.2.1 Diagrama de casos de uso

É um diagrama que descreve um cenário de funcionalidades do ponto de vista do usuário e serve para demonstrar as diferentes maneiras de interação com um

sistema. Esse diagrama faz o uso de bonecos e balões interligados para definir funcionalidades e as possibilidades ao usuário.

A Figura 1 mostrará o diagrama de caso de uso que proporciona uma visão geral das funcionalidades e as maneiras em que o usuário e serviços terceiros interagem com o sistema. É importante ressaltar que os atores que estão à esquerda do diagrama são atores primários, portanto são os interessados ou beneficiados diretamente pelo caso de uso e os atores que estão à direita do diagrama são os atores secundários, portanto prestam apenas um tipo de suporte oferecendo serviços ao sistema.

Figura 1 - Diagrama de caso de uso do sistema



Fonte: Elaborado pelo autor.

Conforme a Figura 1 apresentada, os atores que interagem com a ferramenta são: O usuário e do outro lado a própria internet e uma interface de programação de aplicações (API).

- **Usuário:** é o ator que representa os utilizadores da aplicação. Neste caso, o usuário será capaz de inserir palavras chaves ou um conteúdo específico em um campo de texto do programa para iniciar as pesquisas.
- **Internet:** representa a ligação entre o mecanismo de pesquisa desenvolvido e a própria internet. Permite a interação do programa para com os mais diversos serviços e informações na web.
- **SerpApi:** representa a API integrada ao sistema, possibilitando a busca de dados variados de forma genérica. Essa busca será apresentada ao usuário sempre com as informações mais relevantes.

2.2.1.1 Documentação do caso de uso

Comumente, um diagrama de caso de uso pode possuir diversas variações e alternativas para demonstrar a ideia básica de um sistema. A Figura 1 mostra um exemplo de uma variação no qual podemos perceber que alguns balões estão sendo indicados por uma seta com as seguintes nominações: incluir e estender. Essas nominações indicam a característica de uma funcionalidade para uma determinada condição estabelecida na aplicação, portanto por definição, o caso de uso com a seta indicada por incluir será sempre executado e o caso de uso com a seta indicada por estender pode ou não ser executado. Dessa forma, conforme os conceitos apresentados, cada funcionalidade do diagrama de caso de uso será descrita no Quadro 3 ao Quadro 5.

O Quadro 3 descreve o caso de uso inserir dados. Este caso de uso conta com outros dois sendo o caso de uso incluído verificar dados, portanto pode-se afirmar que sempre haverá uma verificação no ato da inserção de dados, e um caso de uso estendido exibir erro de dados, no qual indica que uma mensagem só será apresentada ao usuário caso haja erros encontrados.

Quadro 3 - Caso de uso para inserção de dados

Nome do caso de uso	Inserir dados
Caso de uso estendido	Exibir erro de dados
Caso de uso incluído	Verificar dados
Atores envolvidos	Usuário
Objetivo	Este caso de uso descreve as etapas percorridas pelo usuário para inserir corretamente uma informação no programa
Ações do ator primário	**Ações do sistema**
1. Insere dados em um campo de texto e clica em um botão para pesquisar	2. O sistema recebe os dados e verifica o conteúdo 3. Pode ou não exibir uma mensagem de erro
Restrições / Validações	1. Deve inserir ao menos um conteúdo não vazio e válido 2. Deve haver conexão com a internet

Fonte: Elaborado pelo autor.

O Quadro 4 descreve o caso de uso pesquisar dados. Este caso de uso está relacionado a toda etapa para que o usuário consiga fazer suas pesquisas. Possui o caso de uso incluído tratar Informações que indica que sempre ocorrerá a execução para o tratamento de conteúdos e o caso de uso estendido salvar dados que é acionado apenas se de fato houver informações a serem salvas.

Quadro 4 - Caso de uso para pesquisa de dados

Nome do caso de uso	Pesquisar dados
Caso de uso estendido	Salvar dados
Caso de uso incluído	Tratar informações

Atores envolvidos	Usuário, internet, SerpApi
Objetivo	Este caso de uso descreve as etapas percorridas pelo usuário, pela internet e pela API na pesquisa dos dados.
Ações do ator primário	**Ações do sistema**
1. Insere as informações a serem pesquisadas e clica no botão pesquisar 5. Poderá consultar, pesquisar e visualizar parte do conteúdo pelo próprio sistema e salvar um arquivo em PDF	2. O sistema utiliza a internet e a API e percorre sites em busca de conteúdos relacionados 3. O conteúdo é tratado e organizado de forma elegível para o usuário 4. Gera um arquivo contendo as informações pesquisadas
Restrições / Validações	1. Deve haver conexão com a internet 2. Site deve estar disponível 3. Usuário deve inserir informações corretas

Fonte: Elaborado pelo autor.

O Quadro 5 descreve o caso de uso consultar links. Trata-se de uma funcionalidade pertinente para consulta, desde que o link seja anteriormente adicionado ou apareça na tela de resultados após uma pesquisa. Possuí também o caso de uso incluído exibir informações que mostra a página do site por meio de um navegador integrado, sempre que todo e qualquer link for acessado.

Quadro 5 - Caso de uso para consulta de links

Nome do caso de uso	Consultar links
Caso de uso estendido	
Caso de uso incluído	Exibir informações
Atores envolvidos	Usuário, internet

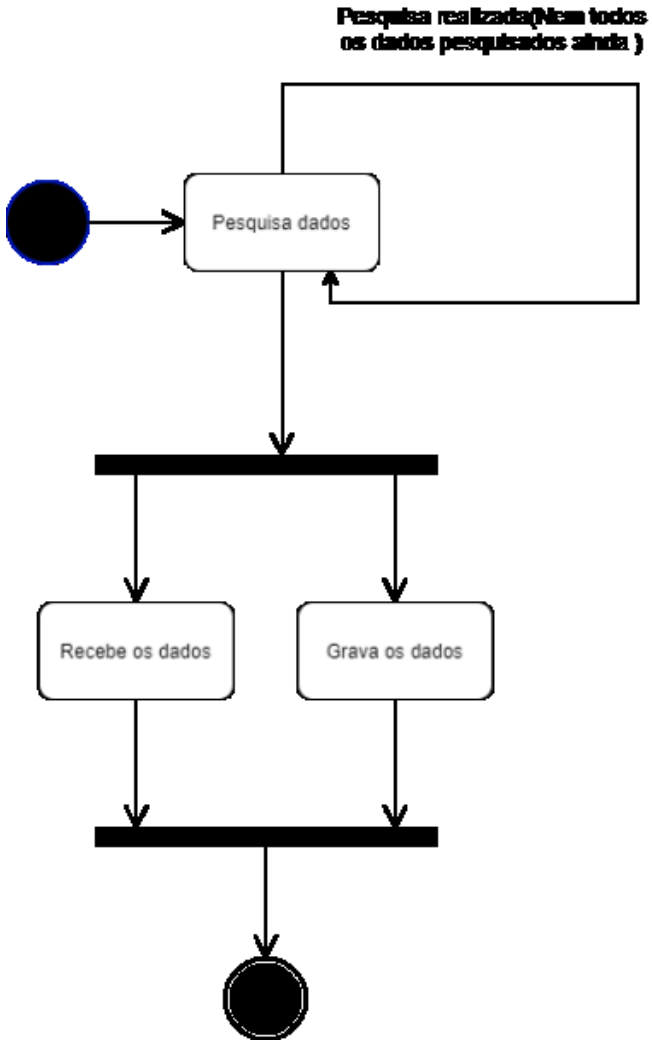
Objetivo	Este caso de uso descreve as etapas percorridas pelo usuário e pela internet na consulta de links
Ações do ator primário	**Ações do sistema**
1. Insere e salva links no programa 2. Abre um navegador na própria estrutura do programa e insere o link ou apenas clica duas vezes no endereço inserido 5. Pode consultar histórico de navegação.	3. Lê o endereço e utiliza a internet com um navegador para busca do site 4. Exibe o site com todas as informações
Restrições / Validações	1. Link inserido ou salvo deve ser válido 2. Deve haver conexão com a internet

Fonte: Elaborado pelo autor.

2.2.2 Diagrama de estados

"Um diagrama de estados, por vezes conhecido como diagrama de máquina de estados, é um tipo de diagrama comportamental na Linguagem de modelagem unificada (UML) que mostra transições entre vários objetos e estados." (LUCIDCHAT, 2021). Para identificar a situação ou estado mais comum no decorrer da execução do programa, foi criado um diagrama de estados conforme indicará a Figura 2.

Figura 2 - Representação do diagrama de estados



Fonte: Elaborado pelo autor.

2.2.2.1 Estados do diagrama

Os estados representados na Figura 2 nos mostram que:

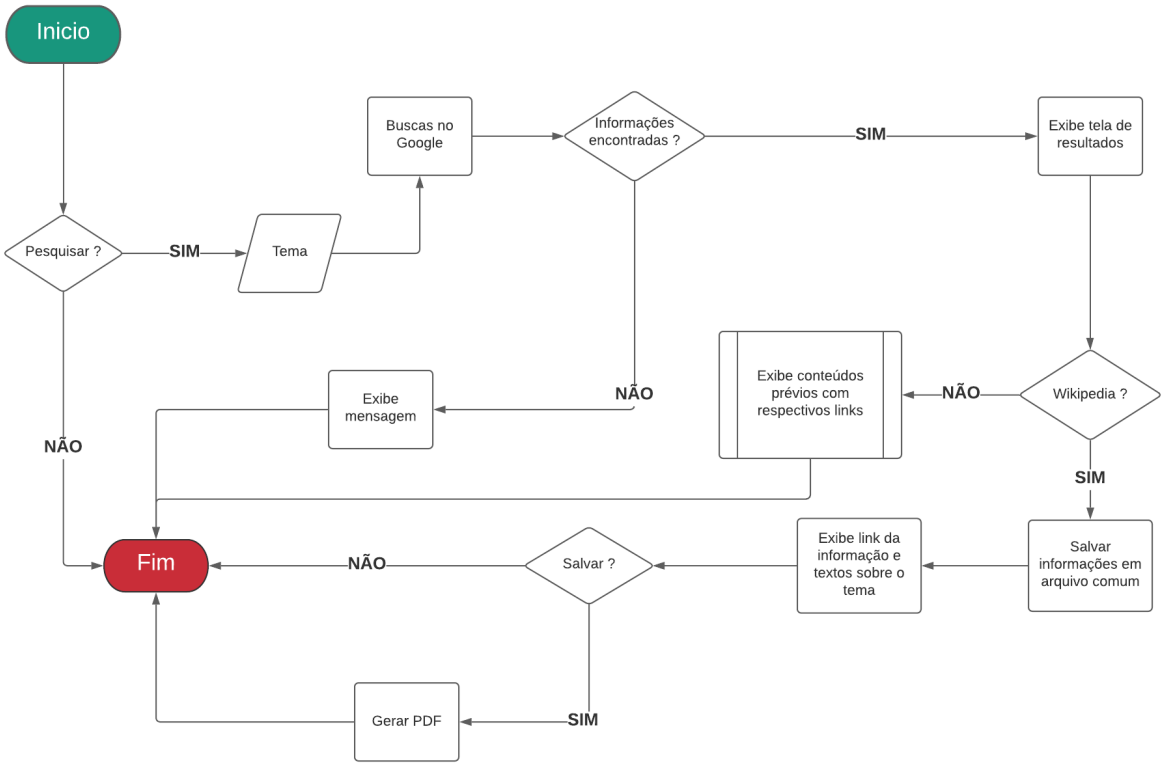
- **Pesquisa de dados:** a partir do ponto inicial o programa fará a pesquisa de dados conforme tema inserido pelo usuário e se manterá neste estado até que toda pesquisa seja realizada. Esse limite é pré-definido de maneira lógica com um total de 10 sites na própria aplicação.
- **Recebe os dados**: após a pesquisa ser realizada, o programa recebe os dados e os exibe para usuário, seja conteúdo ou apenas links com informações prévias.

- **Grava os dados**: após a pesquisa, caso haja conteúdo e textos com informações relevantes, o programa grava os dados em um arquivo de texto para que possa ser consultado por uma tela amigável e ser salvo em PDF.

2.2.3 Fluxograma

"Um fluxograma ilustra as etapas e decisões realizadas para concluir um processo." (LUCIDCHART, 2021). No fluxograma as etapas são organizadas de forma sequencial e facilitam a compreensão do projeto, processo, sistema e o fluxo de trabalho. Podemos afirmar que é uma ferramenta simples, mas poderosa e eficiente para entendermos neste caso, como a funcionalidade de pesquisa se encaixa no Projeto, conforme veremos na Figura 3.

Figura 3 - Fluxograma base para funcionalidade de pesquisa



Fonte: Elaborado pelo autor.

A Figura 3 representa um fluxograma e nos mostra o fluxo de processos para a funcionalidade básica do sistema que é a pesquisa de dados. Utilizamos neste exemplo o site Wikipédia para extração de informações, dada a sua simplicidade e boa diversidade de conteúdo. A seguir, será feita uma breve descrição do funcionamento do fluxo.

O usuário necessita pesquisar um tema de maneira automática, para isso, só precisa inserir alguma informação na aplicação. O algoritmo, fará buscas no Google de duas maneiras distintas, sendo a primeiro utilizando uma API e a segunda por meio de uma biblioteca de código aberto conhecida como Googlesearch que foi desenvolvida na linguagem de programação Python. Enquanto a API busca por resultados, o *Scraper* desenvolvido entra em ação paralelamente utilizando a biblioteca e tenta localizar alguns links, caso um deles seja o Wikipédia, a ferramenta acessa o link e extrai os conteúdos com informações relevantes e as exibe ao usuário, caso não haja conteúdos relacionados pelo site pré-definido, o sistema apenas exibe os conteúdos prévios e links semelhante a uma busca rotineira. Por fim, tendo como base algumas informações e a ideia de que os campos de textos são editáveis, o usuário pode salvar as informações em formato PDF e se preferir, pode inserir ou deletar qualquer conteúdo exibido.

3 DESENVOLVIMENTO

3.1 TECNOLOGIA EMPREGADA

A seguir serão descritas as tecnologias que fizeram a integração e a disposição de práticas e recursos para que a criação de um protótipo fosse possível. Inicialmente, é importante definirmos e entendermos a diferença entre raspagem de dados da web e rastreamento de dados da web, uma vez que, são práticas e algoritmos semelhantes, mas para propósitos diferentes.

3.1.1 Raspagem da Web

Do termo *Web Scraping*, essa técnica permite a extração de dados de sites além de permitir a conversão de informações estruturadas para análises posteriores. Geralmente, utiliza algoritmos conhecidos como *Scrapers* que fazem todas as operações de busca de maneira automatizada, acelerando o acesso as informações de forma eficaz e prática principalmente pela possibilidade de salvar essas informações em arquivos ou então um banco de dados. Tendo como base a ideia, podemos afirmar que o Projeto proposto é um *Scraper*, uma vez que este tipo de algoritmo percorre páginas específicas para extração de informações.

3.1.2 Rastreamento na web

Uma possibilidade para rastreamento de dados na web pode ser criada por meio de um *Web Crawler*. Em um contexto geral, são algoritmos comumente utilizados para indexação na web e são conhecidos como *Crawlers* ou simplesmente um tipo de *Bot* capaz de extrair essencialmente links e informações genéricas. Um *Web Crawler* é muito utilizado dada sua versatilidade que permite explorar diversos links relacionados de forma simultânea, extraindo tudo o que aparece pela frente. O fato é que até buscadores como o gigante Google também utilizam *Crawlers* em sua estrutura, uma vez que esses scripts permitem encontrar, ler e indexar páginas de um site de forma prática e eficiente.

3.1.3 Beautiful Soup

Beautiful Soup é uma biblioteca escrita e desenvolvida em Python que possibilita análises de estruturas encontradas na web como *HyperText Markup Language* (HTML) e *Extensible Markup Language* (XML). Em um contexto em que se existe a necessidade da extração de dados, essa biblioteca fornece métodos simples e poderosos que podem garantir a navegação, pesquisa e extração da estrutura de um site para análises posteriores. Para além das funcionalidades, a *Beautiful Soup* é flexível sobre o padrão Unicode e garante uma conversão automática na saída dos códigos e informações retiradas da web, portanto não existe muita preocupação sobre a codificação dos caracteres extraídos.

3.1.4 Api

Do termo *Application Programming Interface* ou Interface de Programação de Aplicações, é um conjunto de padrões e protocolos que permitem a integração de softwares e aplicações (REDHAT, 2021). Uma API possibilita uma comunicação entre sistemas sem a necessidade de saber como foram implementados. De um modo geral, facilitam o desenvolvimento pois contam com uma boa infraestrutura pré-definida e permitem a comunicação de aplicações afim de agregar valor as funcionalidades. Neste trabalho foi utilizado uma API para auxiliar nas buscas de informações.

3.1.5 PyCharm

É um ambiente de desenvolvimento integrado (IDE) para desenvolvimento especificamente em Python. Oferece atualmente várias ferramentas úteis para ciência de dados, desenvolvimento web e programação no geral. A IDE facilita o desenvolvimento, debugs, testes e ainda ajuda o programador com correções de códigos de forma rápida e prática, além de possuir uma boa interatividade e fornecer recursos para linguagens modernas da web (JETBRAINS, 2021).

3.1.6 Python

É uma linguagem de programação mais especificamente para tratamento de dados, *Big Data*, *Data Science e* inteligência artificial entre muitas outras aplicações que demandam um grande tratamento e garantia da integridade dos dados. Com a tecnologia Python é possível desenvolver jogos, aplicações web com *Django*, *Flask*, aplicações mobile, desktop etc. É uma linguagem de código aberto e prática e possuí uma vasta comunidade de programadores que desenvolvem bibliotecas a todo momento.

3.1.7 QtCreator

É uma IDE para criação de interfaces gráficas de programas que podem funcionar em multiplataforma. Conta com uma biblioteca chamada *PySide* que é escrita em Python e suas aplicações podem rodar desde sistemas operacionais como Windows, Linux e Mac até sistemas operacionais mobile como Android e IOS. Pode ser considerado um framework bem documentado que ajuda os programadores nas mais diversas aplicações (QT, 2021).

3.1.8 LucidChart

É um website e uma ferramenta online para criação de basicamente qualquer tipo de diagrama e gráficos para gerenciamento e desenvolvimento de atividades. Permite criar uma variedade de desenhos de forma intuitiva e possibilita a exportação em diversos formatos o que torna a ferramenta muito versátil. (LUCIDCHART, 2021).

3.1.9 SerpApi

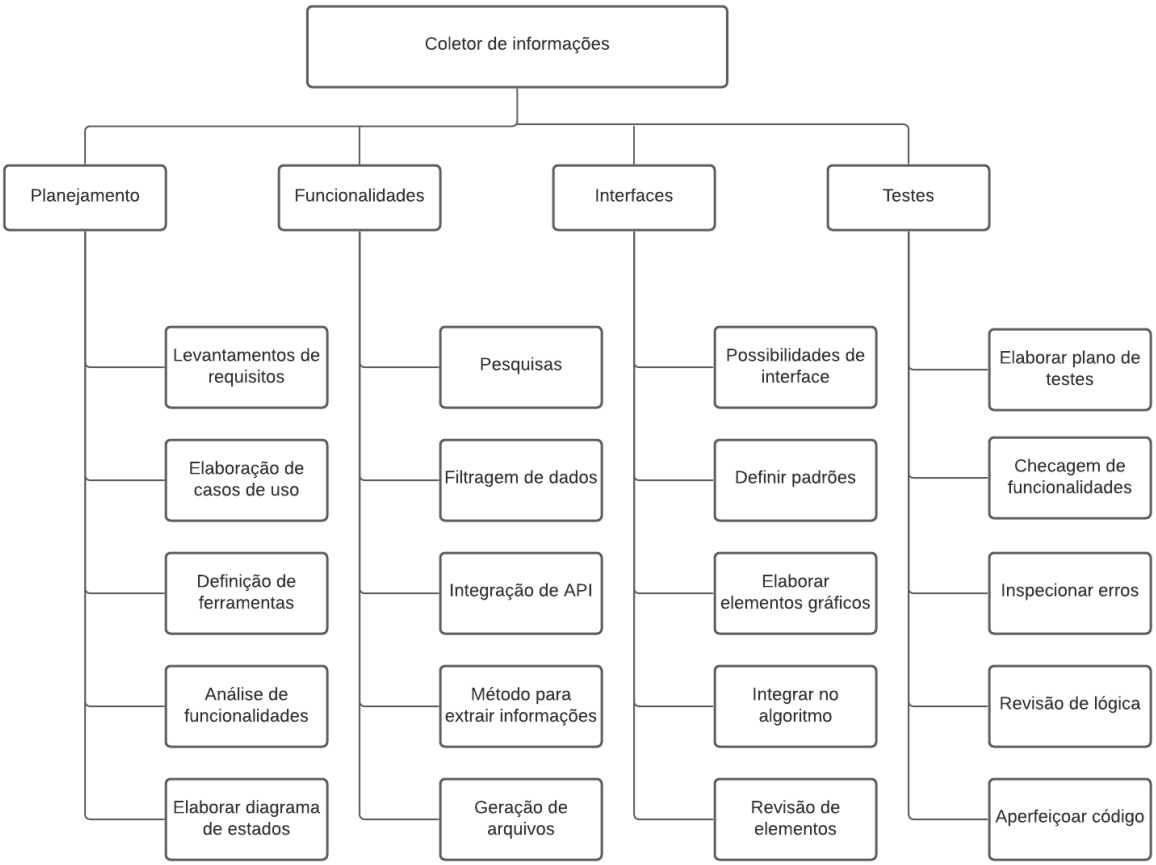
É uma API, que engloba os serviços do *Google Search* e em sua maioria, oferece serviços e padrões que possibilitam o acesso de forma ágil e eficiente ao motor de pesquisa do Google. Em geral, essa API possibilita o acesso as informações de pesquisas e fornece os resultados exibidos pelo próprio Google.

4 PROJETO

4.1 ESTRUTURA ANALÍTICA DO PROJETO

Uma Estrutura Analítica de Projeto (EAP) serve como um diagrama visual para organizar o escopo do projeto e facilitar o gerenciamento das entregas. É útil para organizar o trabalho e decompor de forma hierárquica cada grupo de atividades que deverá ser seguido. De forma geral, a EAP oferece uma visão estratégica sobre os grupos de tarefas e uma boa visualização sobre o desenvolvimento como um todo. A Figura 4 representará uma estrutura analítica e proporcionará a visualização da base do desenvolvimento do Projeto conforme as entregas.

Figura 4 - Representação de estrutura analítica do projeto



Fonte: Elaborado pelo autor.

Conforme a representação da Figura 4 que se refere a uma EAP, veremos alguns apontamentos gerais para as atividades apresentadas.

- **Planejamento:** é o ato de preparar e avaliar as maneiras necessárias para criar, otimizar ou até concluir alguma tarefa. Nesta etapa foram seguidas as práticas e diretrizes também relacionadas a engenharia de software. Este é um ponto essencial porque estabelece claramente os caminhos e a percepção do sistema, desde o levantamento de requisitos até a elaboração dos diagramas nos quais nos fazem compreender como o sistema funcionará. As entregas ocorreram de forma regular e graças as atividades desenvolvidas, foi possível identificar o plano ideal para desenvolver o trabalho.
- **Funcionalidades:** esta etapa é baseada nos requisitos desenvolvidos conforme o planejamento. Foram feitos aprimoramentos e a tentativa de desenvolvimento de um código coerente sobre cada funcionalidade. Este ponto, envolve os fatores mais críticos porque conforme as entregas, conseguimos visualizar se aquilo que era planejado de fato foi realizado.
- **Interfaces:** pode-se dizer que são as portas de entrada para qualquer sistema ou ferramenta. Para este projeto, as interfaces passaram por boas modificações e revisões antes de serem implementadas. Cada entrega permitiu a integração e referenciamento para com as funcionalidades em desenvolvimento.
- **Testes:** é um dos processos mais importantes porque propõe as verificações e validações do sistema e em geral, é a última etapa antes da proposta chegar ao usuário. As entregas permitiram o desenvolvimento de um código conciso pelo fato de os testes ocorrerem durante e depois da criação. Cada entrega permitiu o gerenciamento e validação das ideias propostas incialmente.

4.2 PLANEJAMENTO COM KANBAN

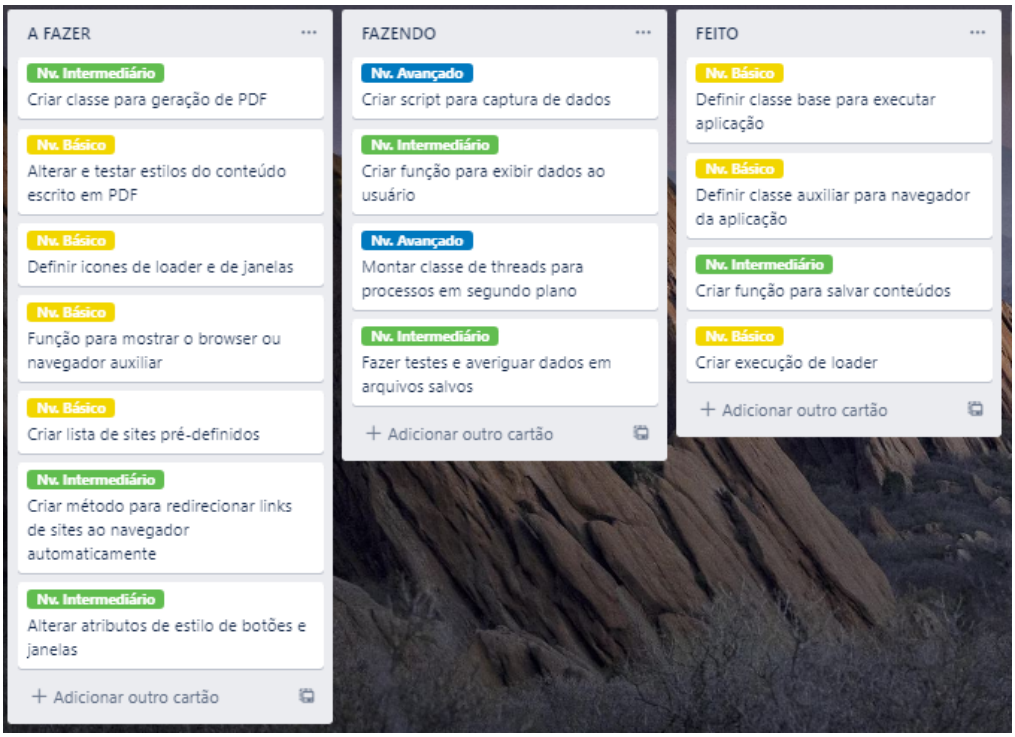
O Kanban é um método visual para gerenciar e conduzir o trabalho. (MESH, 2020). É uma alternativa de metodologia ágil que ajuda a controlar a gestão e fluxo de produção ou projetos e utiliza cartões coloridos para sinalizar prioridades e tarefas a

serem cumpridas. De modo geral, é bastante visual e prático envolvendo um grupo ou equipe de desenvolvimento e facilita na entrega rápida evitando desperdícios de tempo assim como desvio de objetivos.

Neste trabalho, o Kanban foi utilizado apenas nas etapas de desenvolvimento de código para otimizar as entregas de funcionalidades e alcançar objetivos de funcionamento básico do sistema. A metodologia se mostrou eficiente pois definia exatamente aquilo que deveria ser feito e organizava o processo por meio dos cartões virtuais conforme exemplo desenvolvido na Figura 5. Para este tipo de organização, foi utilizada a ferramenta online Trello que facilitou o gerenciamento de projetos por meio de quadros automatizados.

A Figura 5 mostra algumas das atividades de desenvolvimento no Projeto e possuí algumas etiquetas coloridas que diferenciam os níveis de dificuldade para cada atividade, sendo a etiqueta azul uma atividade mais avançada, a etiqueta verde uma atividade intermediária e a etiqueta amarela uma atividade básica. De modo geral, além dos níveis de dificuldade as 3 colunas facilitaram a visualização das tarefas necessárias para avançar e alcançar o desenvolvimento desejado.

Figura 5 - Cartões de atividades por meio do Kanban



Fonte: Elaborado pelo autor.

4.3 SOFTWARE SIMILARES

Atualmente existem várias aplicações direcionadas para busca e pesquisa de conteúdo, muitas delas proporcionam o acesso a uma infinidade de links para que usuários possam acessar e visualizar os mais diversos tipos de informações. Dessa forma, foram selecionadas três aplicações conhecidas quando o assunto é raspagem de dados. Vale ressaltar que o Projeto proposto, segue um caminho um pouco diferente, mas com a mesma ideia de trazer informações relevantes ao usuário conforme sua necessidade.

- **Octoparse:** É um software que possuí recursos para extração de dados da internet, geralmente pode ser utilizado para extração de dados em massa e não é necessário uma codificação previa, sendo relativamente simples até para usuários menos experientes. O software permite extrair conteúdos de diversos sites e permite salvar os elementos de uma maneira estruturada e legível.
- **ParseHub:** Pode-se dizer que é uma alternativa relevante para raspagem de dados da web, possuí uma aplicação tanto para desktop quanto extensões para navegadores, facilitando o rastreamento e também é muito utilizado para análise de dados. Permite extrair elementos automaticamente de múltiplas páginas e interage também com formulários.
- **Helium Scraper:** É uma ferramenta para raspagem de dados que propõe uma interface intuitiva e simples de usar, basicamente possibilita a extração de dados apenas ao apontar o elemento correto na própria página do site e conta com algumas alternativas para extração de dados complexos. Pode ser utilizado para análise de dados de forma geral e possibilita meios de salvar as informações em arquivos.

Conforme o Quadro 6, será apresentado uma breve descrição em relação ao comparativo de funcionalidades. É uma tentativa de indicar apenas as diferenças entre as funcionalidades e recursos básicos das ferramentas, ressaltando que todas elas têm as próprias peculiaridades e funcionam muito bem para seus propósitos.

Quadro 6 - Comparativo de funcionalidades

Descrição	Jambo	Octoparse	ParseHub	Helium Scraper
Extração de qualquer site por seleção de itens no navegador	-	X	X	X
Filtrar elementos	X	X	X	X
Extração de dados automatica	X	X	X	X
Extração de informações com tema estabelecido	X	-	-	-
Salvar em PDF	X	-	-	-
Navegador Integrado	X	-	-	-

Fonte: Elaborado pelo autor.

Todas essas aplicações possuem a mesma ideia em comum, ou seja, coletar informações relevantes da web e salvar os dados para análises. A proposta desse trabalho segue um caminho um tanto diferente no qual o único objetivo é extrairmos informações referentes a um tema indicado pelo próprio usuário. A ferramenta não busca conteúdos aleatórios e segue apenas um caminho direto ao tentar extrair informações relacionadas ao tema proposto. Nesse aspecto, como a ideia é apenas proporcionar uma visão de um sistema capaz de auxiliar e facilitar pesquisas, portanto um tipo de protótipo, o algoritmo retira informações apenas de um único domínio.

O Wikipédia é um website que possui uma imensa quantidade de informações e a cada dia aumenta sua base de dados, embora haja controvérsias, a cada ano é registrado bilhões de visualizações e acessos. Podemos afirmar que é uma excelente ferramenta para pesquisas de artigos e informações variadas, além do mais, é repleto de conteúdos relevantes que também contam com suas respectivas referências. Visando isso, o algoritmo proposto para este trabalho se baseou na retirada de informações exclusivamente desde website.

4.4 TESTES

É uma das atividades em um desenvolvimento de software que visa encontrar falhas e qualquer problema que não deva aparecer ao usuário ou que prejudique a utilização do sistema. Para as próximas seções, foi elaborado um plano de testes no qual contém as etapas básicas relacionadas aos requisitos e entrada e saída de dados.

4.4.1 Plano de testes

“Normalmente, um plano de teste identifica requisitos, riscos, casos de teste, os ambientes de teste a serem testados, objetivos de qualidade e negócios, planejamentos de teste e outros itens” (IBM, 2021). É necessário validações, verificações e controle de atividades para identificar e resolver os problemas do sistema. Portanto, foram desenvolvidos alguns testes visando encontrar falhas e erros na aplicação.

O Quadro 7, representa testes baseados na pesquisa de dados que está relacionado ao campo de busca da aplicação.

Quadro 7 - Teste de pesquisa

Objetivo do teste	Testar busca de dados e verificar informações disponíveis
Requisitos que motivaram o teste	**RF01** - Fazer busca de dados
Procedimento	**Resultado esperado**
1. Inserir alguma informação ou tema no campo de busca	O campo deve receber os dados digitados
2. Pressionar o botão Pesquisar	Uma animação de carregamento deve iniciar
3. Verificar o tempo de resposta da aplicação	As informações devem ser apresentadas o mais breve possível

4. Checar disponibilidade de conteúdos	Uma nova tela deve abrir com os resultados

Fonte: Elaborado pelo autor.

O Quadro 8 indica os testes realizados na lógica interna da aplicação que é referente a filtragem e separação das informações relevantes.

Quadro 8 - Teste de filtragem de informação

Objetivo do teste	Filtrar dados a serem pesquisados, trazendo apenas informações pertinentes ao tema
Requisitos que motivaram o teste	**RF02** - Filtrar dados
Procedimento	**Resultado esperado**
1. Checar informações apresentadas após a pesquisa	Os conteúdos devem estar relacionados ao tema inserido no ato da pesquisa
2. Comparar informações prévias trazidas pela API com as informações capturadas pelo algoritmo	As informações embora distintas devem propiciar dados referentes ao mesmo tema
3. Checar arquivo de armazenamento	O arquivo deve conter algumas informações apenas do site Wikipedia e o texto armazenado deve estar em conformidade com a página

Fonte: Elaborado pelo autor.

O Quadro 9 indica um teste essencial no qual foi analisado a condição de abertura de uma segunda tela que sempre deve ser apresentada após a execução inicial do algoritmo e após a extração de informações pesquisadas.

Quadro 9 - Teste de abertura da tela de resultados

Objetivo do teste	Checar funcionamento de algoritmos paralelos e abertura da página de resultados.
Requisitos que motivaram o teste	**RF03** - Carregar elementos e exibi-los previamente
Procedimento	**Resultado esperado**
1. Após a inserção do tema e início da pesquisa, verificar se a tela está travada	O algoritmo deve exibir uma animação de carregamento e não deve travar nenhuma funcionalidade
2. Testar botões para abrir outras páginas	O algoritmo de busca deve rodar em paralelo a execução da interface e outras páginas devem abrir normalmente
3. Checar abertura de tela de resultados	A tela pode ser exibida com dois campos, sendo um informações prévias e o outro com conteúdo de textos extraídos do Wikipédia
4. Verificar se a tela pode ser expandida	A tela pode ser expandida ou contraída de acordo com o desejo do usuário, todas as informações da tela devem se alocar uniformemente

Fonte: Elaborado pelo autor.

Conforme o Quadro 10, foram feitos alguns testes relacionados a veracidade dos links fornecidos após o ato de uma pesquisa, portanto, este teste visa estabelecer se de fato o link que está sendo apresentado ao usuário é o que foi encontrado na execução do algoritmo.

Quadro 10 - Teste relacionado aos links das pesquisas

Objetivo do teste	Verificar a exibição dos links das pesquisas realizadas
Requisitos que motivaram o teste	**RF04** - Deve fornecer links de pesquisas
Procedimento	**Resultado esperado**
1. Inserir informações na aplicação e pesquisar qualquer tema	Na tela de resultados, deve apresentar todos os links que foram encontrados tanto pela API quanto pelo *Scraper*

Fonte: Elaborado pelo autor.

O Quadro 11 indica um teste importante e essencial que está relacionado a tela de resultados e a um botão para que o usuário possa salvar os conteúdos pesquisados.

Quadro 11 - Teste de salvar informações

Objetivo do teste	Verificar disponibilidade e funcionalidade do botão para salvar as informações
Requisitos que motivaram o teste	**RF05** - Deve possibilitar salvar os dados pesquisados
Procedimento	**Resultado esperado**
1. Com a tela de resultados aberta e qualquer texto inserido no segundo campo de textos, clicar no botão Salvar.	Deve abrir uma janela nativa do sistema operacional do usuário
2. Verificar se a opção para salvar está ativa	A opção não deve ficar ativa enquanto não for inserido um nome para o arquivo
3. Clicar para salvar o documento	Deve gerar um arquivo em PDF com todas as informações de texto no local especificado pelo usuário

Fonte: Elaborado pelo autor.

O Quadro 12 se refere a um teste que está relacionado a uma página que possibilita exclusivamente o armazenamento de links, portanto, este teste foi criado para garantir a disponibilidade de links salvos na aplicação.

Quadro 12 - Teste na tela de links

Objetivo do teste	Verificar se a aplicação está salvando links inseridos na tela de sites e checar a disponibilidade desses links.
Requisitos que motivaram o teste	**RF06** - Deve possibilitar salvar links
Procedimento	**Resultado esperado**
1. Ir na tela inicial da aplicação e pressionar o botão Sites	Deve abrir uma nova tela exclusivamente para salvar links
2. Inserir no campo de texto algum link e pressionar o botão representado por um ícone	Deve inserir o link na aplicação e armazená-lo em um arquivo
3. Observar a lista logo abaixo do campo de texto e dar dois clicks sobre qualquer link armazenado	Deve abrir o navegador da aplicação inserindo automaticamente o link no campo de endereço do navegador.

Fonte: Elaborado pelo autor.

O Quadro 13 indica um teste realizado no navegador da aplicação, sendo uma funcionalidade tanto para consulta de links salvos pelo próprio usuário quanto para informações ou pesquisas adicionais.

Quadro 13 - Teste no navegador

Objetivo do teste	Verificar a disponibilidade de um navegador para consultas alternativas, testar todo o motor de navegação

Requisitos que motivaram o teste	**RF07** - Deve fornecer uma alternativa de navegador
Procedimento	**Resultado esperado**
1. Pressionar o botão Browser na tela inicial	Deve abrir uma nova tela correspondente ao navegador da aplicação
2. Inserir algum site no campo de endereços do navegador	Deve receber o endereço do site digitado
3. Pressionar Enter ao inserir algum endereço	Redireciona para o site escolhido
4. Utilizar os botões representados por setas apontadas para esquerda e direita respectivamente	Fornece a navegação adicional de voltar ou seguir adiante para a próxima página se houver
5. Pressionar o botão representado por um ícone de recarregar	Deve fazer o recarregamento da página semelhante aos atalhos CTRL + F5
6. Utilizar o botão mais à direita com o ícone de uma lista	É exibido o histórico de navegação
7. Fazer qualquer pesquisa pela página inicial	Deve propiciar o acesso as informações e navegação pela internet

Fonte: Elaborado pelo autor.

O Quadro 14 começa abordando alguns testes praticados sobre a ferramenta no quesito desempenho e disponibilidade. Vale ressaltar que é importante garantir um bom funcionamento e a compatibilidade, uma vez que a ideia proposta tende a funcionar tanto no sistema operacional Windows como em Linux.

Quadro 14 - Teste de desempenho

Objetivo do teste	Analisar tempo das pesquisas contando com as pausas de requisições

Requisitos que motivaram o teste	**RNF02** - Todo o processamento e pesquisas não devem ser superiores a 40 segundos
Procedimento	**Resultado esperado**
1. Inserir qualquer tema na tela inicial e clicar no botão Pesquisar	O tempo para exibição da tela de resultados com as informações, deve ser inferior a 40 segundos.

Fonte: Elaborado pelo autor.

Conforme o Quadro 15, foram feitos testes em relação ao comportamento da aplicação para com a disponibilidade de internet, vale ressaltar que a aplicação não busca dados se não houver conexão.

Quadro 15 - Teste de conexão com internet

Objetivo do teste	Verificar disponibilidade e acesso à internet
Requisitos que motivaram o teste	**RNF03** - O sistema somente deverá ser funcional quando houver conexão com a internet
Procedimento	**Resultado esperado**
1. Inserir informações no campo de texto e clicar no botão Pesquisar	Deve exibir uma mensagem de erro caso não haja conexão com a internet
2. Clicar no botão Browser	Deve exibir a página inicial do Google, caso não haja conexão o próprio browser deve exibir uma mensagem indicando a indisponibilidade

Fonte: Elaborado pelo autor.

O Quadro 16 está relacionado a um teste importante de como a aplicação salvará os dados pesquisados, foi feito algumas análises para verificar se o arquivo está sendo gerado e contém todas as informações extraídas. Salientamos que foi de preferência do grupo optar por um algoritmo que salva as informações em arquivos comuns do tipo texto ao invés de um banco de dados, tendo em vista que os tipos de dados extraídos são únicos e apenas em formato de texto.

Quadro 16 - Teste no armazenamento de informações

Objetivo do teste	Analisar após as pesquisas se um arquivo está sendo gerado com as informações pertinentes
Requisitos que motivaram o teste	**RNF04** - Toda informação pesquisada deve ser armazenada em arquivos
Procedimento	**Resultado esperado**
1. Fazer qualquer pesquisa pela tela inicial	Deve aguardar a extração das informações e exibir a tela de resultados
2. Verificar a pasta do algoritmo	*Se* o algoritmo encontrar algum conteúdo no Wikipédia um arquivo de texto será gerado com algumas informações e parágrafos da página

Fonte: Elaborado pelo autor.

O Quadro 17 indica os testes relacionados a execução em sistemas operacionais distintos para analisar o comportamento da aplicação em ambientes alternativos.

Quadro 17 - Teste de execução em sistemas operacionais distintos

Objetivo do teste	Analisar o comportamento da aplicação em sistemas operacionais diferentes

Requisitos que motivaram o teste	**RNF05** - Deve ser compatível com os sistemas operacionais Windows ou Linux
Procedimento	**Resultado esperado**
1. Executar a aplicação e testar todas as funcionalidades e botões no Windows	Todas as funcionalidades devem funcionar normalmente para ambos os sistemas operacionais e qualquer erro será apresentado uma mensagem indicando a causa
2. Executar a aplicação e testar todas as funcionalidades e botões no Linux	

Fonte: Elaborado pelo autor.

4.5 INTERFACES DE USUÁRIO

A necessidade da construção de uma interface amigável ao usuário é fundamental em um sistema. A interface faz parte do sistema computacional e determina como as pessoas operam e controlam o software. Quando uma interface é bem projetada ela é compreensível, agradável e controlável. Neste contexto, essa seção tem como objetivo apresentar os recursos e a ideia de interface desenvolvida para o Projeto.

A Figura 6 apresentará a tela inicial da ferramenta. Nesta interface o usuário pode inserir um dado no campo de textos e fazer as pesquisas.

Figura 6 - Tela inicial da ferramenta

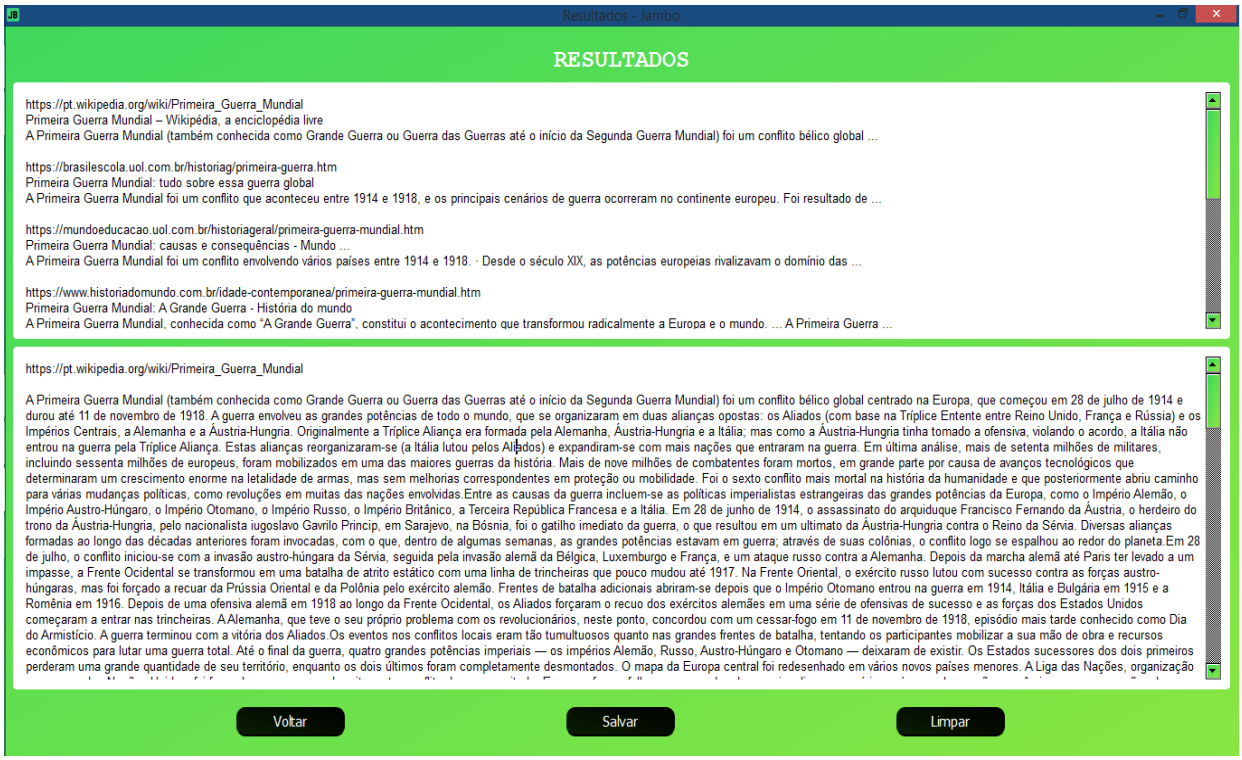
Fonte: Elaborado pelo autor.

A tela inicial conforme apresenta a Figura 6 é composta por:

- **Campo de texto:** Para colocar os dados a serem pesquisados.
- **Botão Pesquisar:** Para iniciar as pesquisas.
- **Botão Sites:** Direciona para uma outra página que possibilita o armazenamento de links de websites.
- **Botão Browser**: Direciona para o navegador auxiliar da ferramenta. O usuário pode navegar como desejar semelhante a navegadores comuns encontrados na web.

A Figura 7 mostrará a tela de resultados após uma pesquisa ser realizada, essa tela só será exibida quando de fato o sistema concluir uma busca de dados.

Figura 7 - Tela de resultados



Fonte: Elaborado pelo autor.

A tela de resultados conforme apresentada na Figura 7 é composta por:

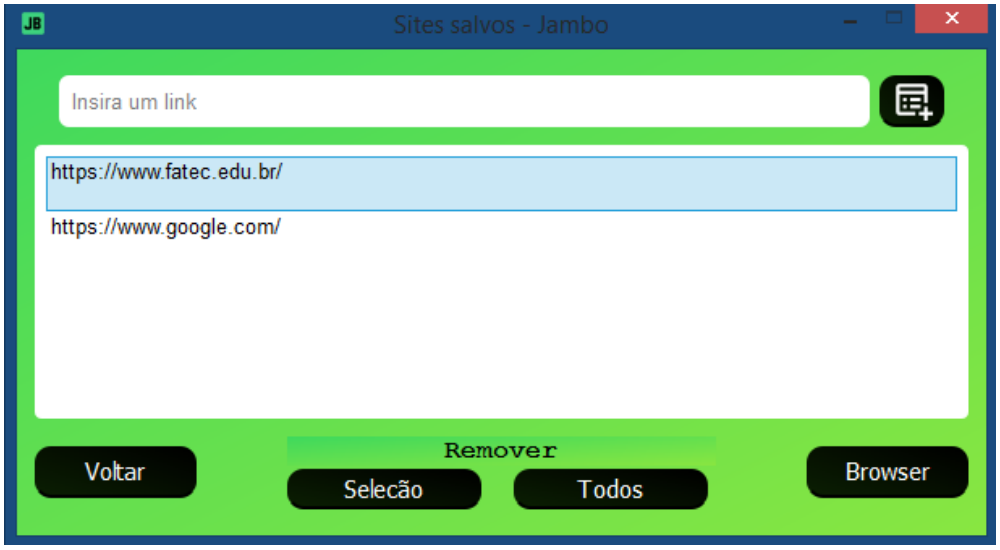
- **Campo de texto prévio:** Exibe dados prévios conforme busca da API, esse campo possibilita visualizar resultados como se fosse uma busca comum

no próprio Google, exibindo os resultados mais relevantes podendo dar uma prévia das informações ao usuário.

- **Campo de texto resultados:** Exibe um texto com informações relevantes e relacionadas ao tema. Neste campo, será exibido parágrafos extraídos do site. Vale ressaltar, que caso não haja resultado encontrado no Wikipédia, a aplicação exibe apenas o campo de texto prévio. Ambos os campos de textos são editáveis e podem ser alterados a qualquer momento.
- **Botão Voltar:** Permite retornar para a página inicial.
- **Botão Salvar:** Permite salvar os dados exibidos no campo de texto resultados. O arquivo sempre será em formato PDF e pode ser salvo em qualquer lugar.
- **Botão Limpar**: Limpa todos os resultados da pesquisa. Neste caso, os arquivos de armazenamento também são limpos.

A Figura 8 apresentará a tela que oferece a possibilidade de armazenar os links de websites.

Figura 8 - Tela para armazenamento de links



Fonte: Elaborado pelo ator.

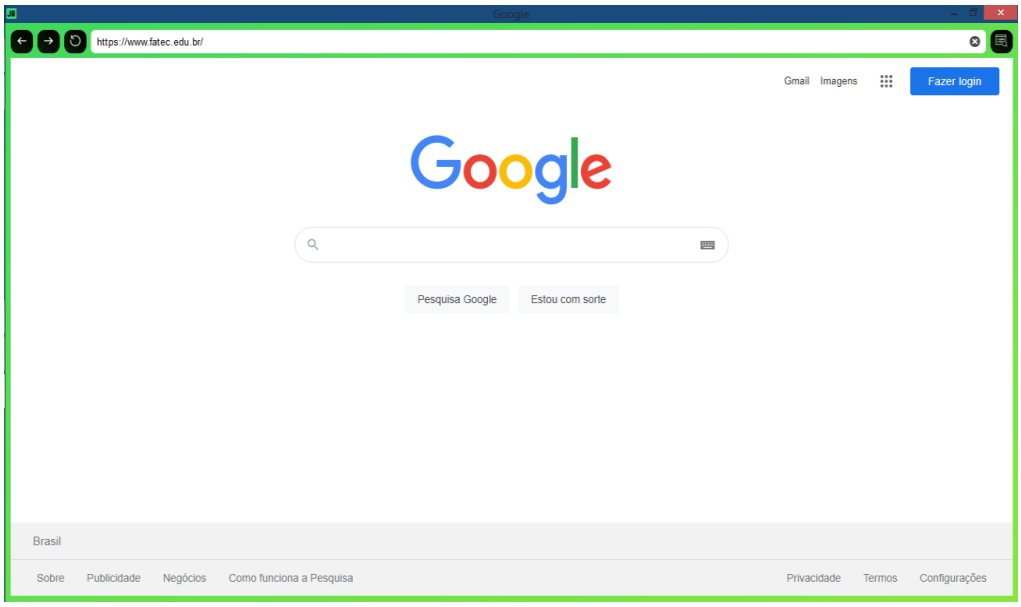
A tela para armazenamento de links conforme apresentada na Figura 8 é composta por:

- **Campo de texto:** Para inserir os links a serem armazenados.

- **Botão Adicionar:** Representado por um ícone de lista, serve para executar a ação de inserir os links.
- **Lista de textos:** Exibe todos os links armazenados, é possível clicar duas vezes sobre o link, que a ferramenta direciona automaticamente para o navegador, o usuário pode também copiar o link pelo atalho CRTL+C.
- **Botão Voltar:** Permite retornar para a página inicial.
- **Botão Remover/Seleção:** Ao selecionar um item na lista de textos, permite remover o item selecionado.
- **Botão Remover/Todos:** Limpa todos os dados da tela e do arquivo de armazenamento.
- **Botão Browser:** Possibilita a abertura do navegador.

A Figura 9 mostrará o navegador da aplicação no qual é possível acessar qualquer website semelhante aos navegadores convencionais. A ideia desta funcionalidade é justamente facilitar pesquisas adicionais sem a necessidade de abrir outros programas. É importante ressaltar que o navegador não foi criado do zero e possuí linhas de código e ferramentas pré-programadas do próprio framework QtCreator e apontamos que utiliza um projeto de código aberto com o mesmo motor do Google, no qual é conhecido como Chromium.

Figura 9 - Navegador da ferramenta



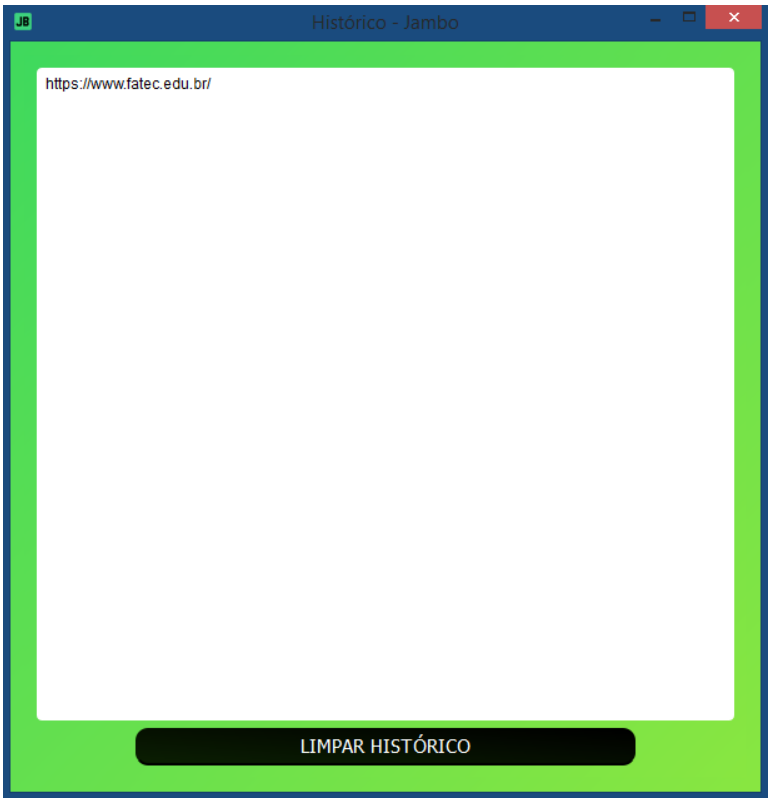
Fonte: Elaborado pelo autor.

A tela do navegador conforme apresentada na Figura 9 é composta por:

- **Campo de texto/endereço:** Para inserir os links e possibilitar o redirecionamento para a página.
- **Botão Voltar:** É representado por uma seta apontando para a esquerda, serve para retornar as páginas.
- **Botão para Frente:** É representado por uma seta apontando para a direita, serve para seguir adiante para as páginas se houver.
- **Botão Histórico:** É representado por um ícone de pesquisa, serve para exibir o histórico do navegador, para isso é exibido uma nova tela.

Conforme representação da Figura 9, o botão histórico possibilita o acesso a uma outra tela no qual é possível visualizar o histórico da navegação. A Figura 10 mostrará a página de históricos no qual possuí apenas um único campo para exibição dos links acessados pelo usuário além de um botão para limpar o histórico de pesquisa.

Figura 10 - Tela de histórico de navegação



Fonte: Elaborado pelo autor.

É importante definir em relação ao navegador, que a primeira página a ser apresentada ao usuário será sempre a página inicial do próprio Google. Isso traz uma certa gama de possibilidades em que se pode inserir um link no campo de endereços ou então navegar normalmente através de informações no campo de pesquisa. A Figura 11 representa um exemplo de uma tela de navegação em que um link foi inserido no campo de textos e uma pesquisa foi realizada, sendo uma tentativa de demonstrar a funcionalidade.

Figura 11 - Pesquisa por link através do navegador

Fonte: Elaborado pelo autor.

5 QUESTÕES ÉTICAS E LEGAIS

Como abordado ao longo do projeto, todo o algoritmo trabalha retirando dados da internet e embora essa tarefa seja algo relativamente simples mesmo sem um programa, podemos dizer que não é correto fazer isso de maneira aleatória afim de desrespeitar servidores e termos de utilização de qualquer site que se encontra na internet. “Os servidores são caros. Além dos custos do servidor, se seus scrapers derrubarem um site ou limitarem sua capacidade de atender a outros usuários, isso pode ser incluído no dano causado por você” (MITCHELL, 2019, p. 310).

Ao longo da execução de um *Web Scraper*, podem ser realizadas diversas requisições a servidores solicitando o acesso aos conteúdos e informações, isso feito de maneira descontrolada pode causar infrações e danos ao próprio servidor ou mesmo a um provedor em que o site está hospedado.

É importante salientar que as práticas de *Web Scraping* e algoritmos automatizados não são ilegais desde que sejam criados de maneira consciente e utilizados em sites de domínio público, ou seja, sites ou páginas em que qualquer pessoa pode ter acesso respeitando também as políticas e diretrizes, por outro lado, é de se pensar o motivo para qual muitas plataformas constantemente desenvolvem ferramentas para inibir que robôs possam coletar dados automaticamente, afinal, qual é a vantagem para um provedor de informações no qual um robô consegue extrair dados facilmente para o usuário? Do ponto de vista do próprio usuário, podemos afirmar que se torna mais prático o acesso à informação, já do ponto de vista do provedor é algo complicado e discutível, pois mais vale um usuário navegando e interagindo do que uma máquina coletando dados.

Sabemos que todo e qualquer conteúdo criado mesmo na internet está sujeito automaticamente a pertencer a um determinado dono assim que estiver disponível, a lei dos direitos autorais garante a posse de toda e qualquer obra intelectual e em hipótese alguma esse trabalho tem como ideia ou objetivo infringir esses direitos. Felizmente nesta questão, a ferramenta tenta extrair fatos e informações de um portal muito conhecido e aberto com uma vasta quantidade de artigos e informações, ressaltando também que quando os conteúdos são exibidos, o usuário tem os dados necessários para saber de onde foram tiradas as informações.

6 CONCLUSÃO

Este trabalho tem como objetivo final transmitir a ideia de uma ferramenta de fácil manuseio para auxiliar em pesquisas e acesso à informação com fins estudantis. Houve inúmeras dificuldades, pois um algoritmo complexo para buscar dados pode envolver aprendizagem de máquina e inteligência artificial, no entanto, este não é o propósito deste trabalho. Após alguns estudos e pesquisas foi extraído uma base essencial para dar sequência no Projeto e viabilizar uma aplicação que extraísse conteúdos com uma tela amigável, é claro que nem todo esse processo foi tranquilo, de qualquer forma, a aplicação se mostrou funcional e versátil para o dia a dia mesmo sendo apenas demonstrativa.

Em relação aos problemas que podemos nos deparar no decorrer da utilização da ferramenta, o *Web Scraper* utilizado é irrelevante em relação a sobrecarga de qualquer servidor, é fato que os *Scrapers* são scripts automatizados capazes de percorrer e extrair milhares de dados simultaneamente, no entanto, foi feita uma tentativa de trazer uma abordagem diferente para o tema, além do que, foi inserido limitações e pausas consideráveis entre as requisições e isto torna a ferramenta praticamente imperceptível mas não menos funcional. Portanto, o algoritmo foi desenvolvido de forma satisfatória empregando as práticas de engenharia de software com o levantamento de requisitos, diagramas e os testes que ocorreram durante e após o desenvolvimento, sendo toda a etapa de planejamento essencial para garantir a viabilidade dos trabalhos.

REFERÊNCIAS

IBM. Desenvolvendo Planos de Teste. IBM, 2021. Disponível em: https://www.ibm.com/docs/pt-br/elm/6.0.5?topic=testing-developing-test-plans. Acesso em: 25 de jun. 2021.

JETBRAINS. O IDE Python para desenvolvedores profissionais. JetBrains, 2021. Disponível em: https://www.jetbrains.com/pt-br/pycharm/. Acesso em: 17 de jun. 2021.

LUCIDCHART: O que é um diagrama de máquina de estados?. Lucid Software Inc, 2021. Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml. Acesso em: 22 de jun. 2021.

LUCIDCHART. Todas as formas de que você precisa para criar fluxogramas profissionais. Lucid Software Inc, 2021. Disponível em: https://www.lucidchart.com/pages/pt/exemplos/fluxograma-online. Acesso em: 25 de jun. 2021.

MESH, Janet. Método Kanban. Trello, 2020: Disponível em: https://blog.trello.com/br/metodo-kanban. Acesso em: 18 de jun. 2021.

MITCHELL, Ryan. Web Scraping com Python: Coletando mais dados na web moderna. 2. ed. São Paulo: Novatec, 2019.

QT. A Cross-Platform IDE for Application Development. Qt, 2021. Disponível em: https://www.qt.io/product/development-tools. Acesso em: 19 de jun. 2021.

REDHAT. Interface de Programação de Aplicações. Red Hat, 2021. Disponível em: https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces. Acesso em: 26 de out. 2021.

WIKIPEDIA. Engenharia de Software. Wikipedia, 2021. Disponível em: https://pt.wikipedia.org/wiki/Engenharia_de_software. Acesso em: 19 de mai. 2021.

APÊNDICE A – DONWLOAD DO PROJETO

Todo o trabalho relacionado ao código e scripts podem ser encontrados nas plataformas GitHub e Google Drive. Salientamos que todos os arquivos podem ser baixados e utilizados livremente desde que não seja para fins comerciais.

GitHub é uma plataforma de hospedagem de código fonte e arquivos variados. A plataforma permite um grande controle de versionamento de código e é muito útil para programadores compartilhar e desenvolver códigos abertos.

- **Link**: https://github.com/caioTeless/final_jambo

Google Drive é um serviço do google para armazenamento e sincronização de dados em nuvem, muito útil para guardar arquivos sem exigir muito do hardware ou dispositivos físicos.

- **Link:** https://drive.google.com/drive/folders/1olhc9SH-qCsPa5LdhQ8tpRzrP0Szv1ZX?usp=sharing