



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CARTOGRÁFICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS GEODÉSICAS E
TECNOLOGIA DA GEOINFORMAÇÃO

SABRINA DE OLIVEIRA GOMES

**ABORDAGEM PRÁTICA DA INTEGRAÇÃO DO GOOGLE EARTH ENGINE E
GOOGLE COLABORATORY NO PROCESSAMENTO DIGITAL DE IMAGENS**

Recife
2024

SABRINA DE OLIVEIRA GOMES

**ABORDAGEM PRÁTICA DA INTEGRAÇÃO DO GOOGLE EARTH ENGINE E
GOOGLE COLABORATORY NO PROCESSAMENTO DIGITAL DE IMAGENS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação da Universidade Federal de Pernambuco, como requisito para a obtenção do título de Mestre em Ciências Geodésicas e Tecnologias da Geoinformação.

Área de concentração: Ciências Geodésicas e Tecnologias da Geoinformação.

Orientador (a): Profa. Dra. Ana Lúcia Bezerra Candeias.

Recife

2024

. Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Gomes, Sabrina de Oliveira.

Abordagem prática da integração do Google earth engine e google colab no processamento digital de imagens / Sabrina de Oliveira Gomes. - Recife, 2024.

93f.: il.

Dissertação (Mestrado), Universidade Federal de Pernambuco, Centro de Tecnologia e Geoinformação, Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação, 2024.

Orientação: Ana Lúcia Bezerra Candeias.

1. Cartografia; 2. Processos automatizados; 3. Python. I. Candeias, Ana Lúcia Bezerra. II. Título.

UFPE-Biblioteca Central

CDD 526.1

SABRINA DE OLIVEIRA GOMES

**ABORDAGEM PRÁTICA DA INTEGRAÇÃO DO GOOGLE EARTH ENGINE E
GOOGLE COLABORATORY NO PROCESSAMENTO DIGITAL DE IMAGENS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação da Universidade Federal de Pernambuco, como requisito para a obtenção do título de Mestre em Ciências Geodésicas e Tecnologias da Geoinformação
Área de concentração: Ciências Geodésicas e Tecnologias da Geoinformação.

Aprovada em: 04/03/2024.

BANCA EXAMINADORA

Prof. Dra. Ana Lúcia Bezerra Candeias (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. João Rodrigues Tavares Junior (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Carlos Fabrício Assunção da Silva (Examinador Externo)
Universidade Federal de Pernambuco

SABRINA DE OLIVEIRA GOMES

**ABORDAGEM PRÁTICA DA INTEGRAÇÃO DO GOOGLE EARTH ENGINE E
GOOGLE COLABORATORY NO PROCESSAMENTO DIGITAL DE IMAGENS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação da Universidade Federal de Pernambuco, como requisito para a obtenção do título de Mestre em Ciências Geodésicas e Tecnologias da Geoinformação
Área de concentração: Ciências Geodésicas e Tecnologias da Geoinformação

Aprovada em: 04/03/2024.

BANCA EXAMINADORA

Prof. Dra. Ana Lúcia Bezerra Candeias (Orientadora)
Universidade Federal de Pernambuco

Prof. Dr. João Rodrigues Tavares Junior (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Carlos Fabrício Assunção da Silva (Examinador Externo)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Gostaria de dedicar este espaço para expressar minha sincera gratidão à Deus, à minha família, à minha orientadora e a todas as pessoas e à Universidade que contribuíram para a realização deste Trabalho.

Aos meus pais Adriana e Sérgio, meu mais profundo agradecimento, vocês foram os pilares essenciais neste caminho. Ao meu companheiro de vida, Thales, agradeço pelas palavras de incentivo, gestos de carinhos e paciência dedicada a mim durante esta jornada acadêmica. Todos vocês foram cruciais para minha perseverança na conclusão desta etapa, sendo um papel vital nesta jornada.

A minha querida orientadora, Dr^a Ana Lúcia, expresso minha sincera admiração e gratidão. Sua orientação foi mais que acadêmica, agradeço pela paciência, pela sabedoria compartilhada e pelo comprometimento que tornou este projeto possível.

Este trabalho não é apenas meu, mas uma conquista compartilhada com aqueles que estiveram ao meu lado. Agradeço a todos por fazerem parte dessa trajetória significativa.

RESUMO

A aplicação de técnicas de sensoriamento remoto é amplamente utilizada em diversos campos de estudo, como a cartografia, devido à riqueza de informações disponíveis nos dados coletados. A obtenção dessas informações é diretamente influenciada pelas abordagens e metodologias empregadas no processamento dos dados adquiridos. Atualmente, os sensores fornecem um volume significativo de dados que podem ser aplicados em uma variedade de contextos de pesquisa e aplicações. O presente trabalho visa desenvolver procedimentos automatizados para simplificar o processamento de imagens capturadas por sensores remotos, com foco nos dados provenientes do satélite Landsat-8. Para isso, foram criadas rotinas computacionais em linguagem Python, utilizando as plataformas do Google Earth Engine e Google Colaboratory. Essas rotinas abrangem a aplicação de índices, Índice de Vegetação por Diferença Normalizada (NDVI), Índice de Vegetação Ajustado ao Solo (SAVI), Índice de Água de Diferença Normalizada (NDWI), Índice de área construídas por diferença normalizada (NDBI). O processo consiste em várias etapas, iniciando pela delimitação da área de interesse, seguida pela preparação das imagens, como aplicação de máscaras para remover elementos indesejados como nuvens. Por fim, os processamentos serão convertidos em formato tabular, permitindo que sejam feitas análises estatísticas, gráficas e a representação cartográfica dos resultados. O desenvolvimento e aplicação dessas rotinas poderá colaborar para futuros trabalhos que desejem aprimorar esse tipo de rotina.

Palavras-chave: Cartografia; Processos automatizados; Python; Landsat-8; Google Earth Engine; Google Colab.

ABSTRACT

The application of remote sensing techniques is widely used in various fields of study, such as cartography, due to the wealth of information available in the collected data. Obtaining this information is directly influenced by the approaches and methodologies used in processing the acquired data. Currently, sensors provide a significant volume of data that can be applied in a variety of research contexts and applications. The present work aims to develop automated procedures to simplify the processing of images captured by remote sensors, focusing on data from the Landsat-8 satellite. For this, computational routines were created in Python language, using the Google Earth Engine and Google Colaboratory platforms. These routines cover the application of indices, Normalized Difference Vegetation Index (NDVI), Soil-Adjusted Vegetation Index (SAVI), Normalized Difference Water Index (NDWI), Normalized Difference Built Area Index (NDBI). The process consists of several steps, starting with delimiting the area of interest, followed by preparing the images, such as applying masks to remove unwanted elements such as clouds. Finally, the processes will be converted into tabular format, allowing statistical and graphic analyzes and cartographic representation of the results to be carried out. The development and application of these routines could contribute to future work that seeks to improve this type of routine.

Keywords: Cartography; Automated processes; Python; Landsat-8; Google Earth Engine; *Google Colab*.

LISTA DE FIGURAS

Figura 1 - Plataforma do GEE e seus catálogos de dados	18
Figura 2 - Ambiente de execução do Google Colaboratory	19
Figura 3 - Diagrama de blocos apresentando a forma de uso do Colaboratory .	20
Figura 4 - Mapa de localização da área de estudo	31
Figura 5 - Fluxograma da metodologia	33
Figura 6 - Delimitação da área de estudo	39
Figura 7 - Visualizações gráficas recomendadas	49
Figura 8 - Dados exportados	54
Figura 9 - Gráfico temporal NDVI	59
Figura 10 - Gráfico temporal SAVI	60
Figura 11 - Gráfico temporal NDWI	61
Figura 12 - Gráfico temporal NDBI	61
Figura 13 - Gráfico temporal com linha de tendência NDVI	62
Figura 14 - Gráfico temporal com linha de tendência SAVI	63
Figura 15 - Gráfico temporal com linha de tendência NDWI	63
Figura 16 - Gráfico temporal com linha de tendência NDBI	64
Figura 17 - NDVI (Período de chuva)	65
Figura 18 - SAVI (Período de chuva)	65
Figura 19 - NDWI (Período de chuva)	66
Figura 20 - NDBI (Período de chuva)	66
Figura 21 - NDVI (Período de seca)	67
Figura 22 - SAVI (Período de seca)	67
Figura 23 - NDWI (Período de seca)	68
Figura 24 - NDBI (Período de seca)	68

LISTA DE TABELAS

Tabela 1 - Características do Sensor OLI	26
Tabela 2 - Resultado obtido da análise de mínimo, máximo e média para cada data analisada NDVI	56
Tabela 3 – Resultado da análise de mínimo, máximo e média para cada data analisada SAVI	57
Tabela 4 - Resultado da análise de mínimo, máximo e média para cada data analisada NDWI	57
Tabela 5 - Resultado da análise de mínimo, máximo e média para cada data analisada NDBI	58

LISTA DE QUADROS

Quadro 1 - Código para instalação e acesso	34
Quadro 2 - Código para delimitação da área de estudo	36
Quadro 3 - Código para importar as imagens do Landsat-8	40
Quadro 4 - Código para remoção de nuvens	42
Quadro 5 - Função para calcular o NDVI	43
Quadro 6 - Função para calcular o SAVI	43
Quadro 7 - Função para calcular o NDWI	44
Quadro 8 - Função para calcular o NDBI	44
Quadro 9 - Código que converte as informações de imagem em tabela	45
Quadro 10 - Remoção de dados	47
Quadro 11 - Código para visualizar os dados a partir de gráficos	49
Quadro 12 - Código para gerar gráficos com linhas de tendência	50
Quadro 13 - Plotagem dos resultados	52
Quadro 14 - Exportação dos dados para o Google Drive	53
Quadro 15 - Código que retorna o tempo de execução total do programa	55

Sumário

1 INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO DO TEMA E PROBLEMÁTICA.....	14
1.2 OBJETIVOS DO TRABALHO	15
1.3 JUSTIFICATIVA.....	15
1.4 ESTRUTURA DO TRABALHO	16
2 REFERENCIAL TEÓRICO	16
2.1 PROCESSAMENTO DIGITAL TRADICIONAL	16
2.2 GOOGLE EARTH ENGINE E GOOGLE COLLABORATORY	17
2.3 LINGUAGEM PYTHON	21
2.4 SENSORIAMENTO REMOTO E PROCESSAMENTO DE IMAGENS	22
2.5 LANDSAT- 8 E SISTEMA OLI	25
2.6 ÍNDICE DE VEGETAÇÃO POR DIFERENÇA NORMALIZADA (NDVI)	27
2.7 ÍNDICE DE VEGETAÇÃO AJUSTADO AO SOLO (SAVI).....	27
2.8 ÍNDICE DE ÁGUA DE DIFERENÇA NORMALIZADA (NDWI).....	28
2.9 ÍNDICE DE ÁREA CONSTRUÍDA POR DIFERENÇA NORMALIZADA (NDBI).....	29
3 ÁREA DE ESTUDO	30
4 MATERIAIS E MÉTODOS UTILIZADOS	32
4.1 DESCRIÇÃO DOS MATERIAS UTILIZADOS	32
4.2 DESCRIÇÃO DA METODOLOGIA	32
4.2.1 Bibliotecas e acesso	34
4.2.2 Delimitação da área de estudo	36
4.2.3 Aquisição de imagens	39
4.2.4 Máscara de nuvens	40
4.2.5 NDVI, SAVI, NDWI E NDBI	42
4.2.6 Conversão de dados	45
4.2.7 Visualização dos dados	51
4.2.8 Exportação dos dados	53
4.2.9 Tempo de execução	54
5 RESULTADOS	56
5.1 Resultados obtidos a partir da programação	56
CONSIDERAÇÕES FINAIS	70
REFERÊNCIAS	73
APÊNDICE A – GOOGLE COLLABORATORY E O GOOGLE EARTH ENGINE (GEE) PARA A ANÁLISE MULTITEMPORAL DAS MUDANÇAS DE	

TEMPERATURA OCACIONADAS POR ILHAS DE CALOR A PARTIR DO TERRACLIMATE	75
APÊNDICE B – INTEGRAÇÃO DO GOOGLE EARTH ENGINE E GOOGLE COLLABORATORY NO PROCESSAMENTO DIGITAL DE IMAGENS: UMA ABORDAGEM PRÁTICA PARA ANÁLISE ESPACIAL COM GRANDE VOLUME DE DADOS	87

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO TEMA E PROBLEMÁTICA

Existem diversos softwares capazes de mapear, analisar e gerar resultados a partir de modelagem matemática usando algoritmos complexos. No entanto, muitos desses softwares exigem licenças caras e não disponibilizam seus algoritmos, dificultando a compreensão por parte dos usuários.

Para estudos que necessitem utilizar múltiplas imagens, vários sensores, análises temporais e múltiplas épocas, o sensoriamento remoto, processamento digital de imagens e geoprocessamento, são técnicas que auxiliam diversos trabalhos. Assim, o uso de ferramentas em nuvem com banco de dados de imagens e programação em código aberto são facilitadores para a geração de resultados que precisam ser desenvolvidos a partir desses materiais.

Yang et. al. (2017) afirmam que para trabalhar com análises de imagens orbitais de fluxo contínuo demanda grande capacidade de processamento de dados, o que é impraticável com computadores e programa de geoprocessamento comuns.

Nesse contexto, existem opções gratuitas para desenvolver códigos e compartilhá-los com a comunidade acadêmica e profissional, como o Google Earth Engine (GEE). A Google desenvolveu uma plataforma que disponibiliza uma coleção de imagens orbitais e de dados geoespaciais em uma escala planetária e com recursos para análise contidos na nuvem, chamada Google Earth Engine (GEE) (KINAKH; BUN; DANYLO, 2017))

Segundo Almeida (2023), o GEE possui um amplo banco de dados de sensores orbitais. No entanto, embora a plataforma compartilhe muitos códigos, às vezes é necessário desenvolvê-los dependendo da necessidade do usuário, o que é dificultado pela falta de tutoriais em português e pela ausência de trabalhos acadêmicos desenvolvidos em torno desta problemática.

O GEE e o Google Colaboratory (Google Colab) são ferramentas poderosas para o processamento digital de imagens. O GEE é uma plataforma baseada em nuvem para processamento de dados geoespaciais, permitindo acessar e processar diversos dados de sensoriamento remoto de diferentes fontes. Por outro lado, o Google Colab é uma plataforma de desenvolvimento em nuvem baseada em notebooks do Jupyter. Segundo Da Almeida (2020), o Colab

permite a criação de scripts em Python e o uso de bibliotecas de processamento de imagens e dados geoespaciais.

A integração dessas duas ferramentas pode permitir uma análise mais eficiente e ágil de grandes volumes de imagens, possibilitando a criação de rotinas. Levando em consideração todo o contexto apresentado, serão apresentadas as funcionalidades das plataformas, com o objetivo de abrir um novo leque de possibilidades para automação e agilidade, utilizando o Landsat-8, tornando o processo mais otimizado e trazendo novas possibilidades de aplicação.

Ao término do trabalho, constatamos que é viável desenvolver rotinas que tornem o processo mais versátil. A acessibilidade aos dados facilita a programação e promove um ambiente colaborativo entre os usuários, resultando em tempos de execução mais curtos em comparação com o volume de dados processados.

1.2 OBJETIVOS DO TRABALHO

O trabalho tem como objetivo principal explorar e demonstrar uma abordagem prática para o uso combinado do GEE e Google Colab no processamento com criação de rotinas e análise de imagens espaciais. Abordando os seguintes tópicos:

- i) Desenvolver um procedimento automatizado, criando rotinas utilizando linguagem Python através da integração do GEE e Google Colab;
- ii) Aplicação de índices utilizando o Landsat-8;
- iii) Avaliar os benefícios e limitações da integração do GEE e Google Colab.

1.3 JUSTIFICATIVA

O trabalho é motivado devido à escassez de trabalhos desenvolvidos em português com essa temática, onde existe a necessidade de otimizar o processamento digital de imagens provenientes do sensoriamento remoto por meio da criação de rotinas. A utilização da linguagem de programação Python

em conjunto com as plataformas do GEE e Colab buscam otimizar tarefas, tornando o processo mais prático e metodologias mais ágeis.

1.4 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte forma: será apresentado o referencial teórico com o intuito de oferecer a base conceitual do estudo. Em seguida, serão abordados os seguintes tópicos: materiais e métodos, detalhando os recursos utilizados, descrição da metodologia, área de estudo. Posteriormente, serão discutidos o desenvolvimento e os resultados obtidos, seguidos por uma análise e síntese dos resultados. E por fim, serão apresentadas as considerações finais.

2 REFERENCIAL TEÓRICO

Para um melhor entendimento dos princípios que caracterizam esse estudo, os tópicos a seguir irão abordar as questões que servirão como embasamento teórico para a aplicação desse estudo.

2.1 PROCESSAMENTO DIGITAL TRADICIONAL

A técnica mais utilizada no processamento digital de imagens é realizando todo o processamento através de alguma ferramenta de SIG (Sistema de Informações Geográficas), como o QGIS. Onde é necessário realizar diversas etapas para preparar e analisar as imagens de satélite.

Silva et. al. (2019), afirma que, para um processamento de imagem é necessário seguir etapas que geralmente incluem a aquisição da imagem, sua importação para o software, o pré-processamento e o realce.

O método tradicional para realizar o processamento, é iniciado a partir do momento que o usuário adquire a imagem, esse processo pode ser trabalhoso, pois além da necessidade de buscas em fontes confiáveis, existem parâmetros específicos que são obrigatórios para a seleção das imagens. Parâmetros como, data e resolução são primordiais para a obtenção das imagens nas quais deseja-se trabalhar.

Após as imagens obtidas, é necessário realizar a importação para alguma SIG, que necessita de comandos específicos para carregá-la na interface do usuário.

Com os procedimentos iniciais realizados, é necessário realizar o pré-processamento na imagem, como correções, remoções de ruídos e ajustes. A partir desses procedimentos, uma variedade de análises pode ser conduzida a partir de ferramentas existentes no software utilizado,

No entanto, ao comparar esse processo tradicional com o uso de ferramentas específicas para o processamento digital de imagens, como por exemplo o Google Earth Engine, percebe-se uma diferença significativa em termo de eficiência e conveniência.

Gómez-Dans et. al. (2016) destacam que o Google Earth Engine automatiza muitos desses processos e oferece acesso direto a uma vasta quantidade de dados de SR. Isso resulta em uma redução significativa no tempo e esforço para preparar e analisar imagens de satélite, permitindo uma abordagem mais rápida e eficaz para estudos de geoinformação e monitoramento.

2.2 GOOGLE EARTH ENGINE E GOOGLE COLABORATORY

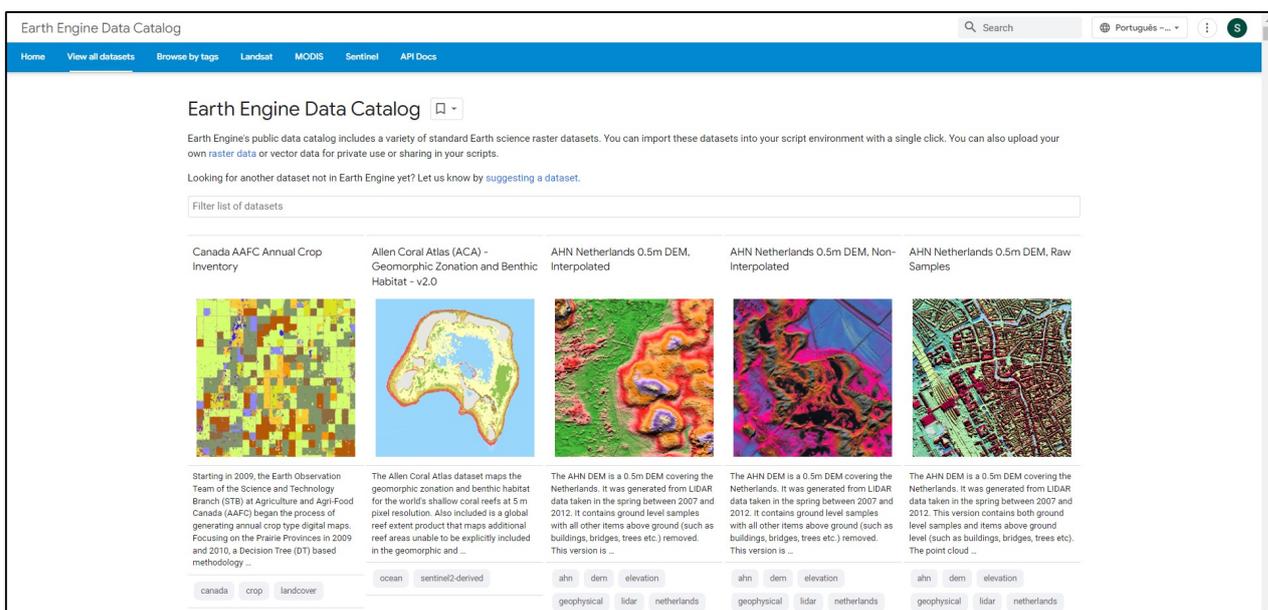
Devido a necessidade de tornar o processo mais adequado quando trabalhado com dados de SR, o processamento em plataformas baseadas em nuvem se mostra a forma mais prática. Visto que são tarefas mais intensivas onde existe a necessidade de trabalhar com grandes volumes de dados de imagens.

Segundo Fritz et. al. (2019), a combinação de computação em nuvem com técnicas de sensoriamento remoto para processamento de séries temporais é uma maneira promissora de explorar dados recentes de sensoriamento remoto.

O GEE é uma plataforma de geoprocessamento desenvolvida com o propósito de ter funcionalidades de aquisição, processamento e compartilhamento de dados geográficos. Baseada em nuvem, ela permite executar análises em larga escala por meio de APIs em Javascript e Python. Além das funcionalidades informadas, ele é totalmente gratuito, com possibilidade de visualização e download dos seus dados espaciais.

O GEE tem como principal proposta fornecer dados espaciais a partir de uma plataforma totalmente gratuita, desenvolvimento de algoritmos para aplicação geoespacial, permitindo que seja feita uma automatização de um grande conjunto de dados espaciais em escala global. Possui um repositório de conjuntos geoespaciais e de imagens de diversos satélites, de 1980 até os dias atuais.

Figura 1 - Plataforma do GEE e seus catálogos de dados



Fonte: Google (2023).

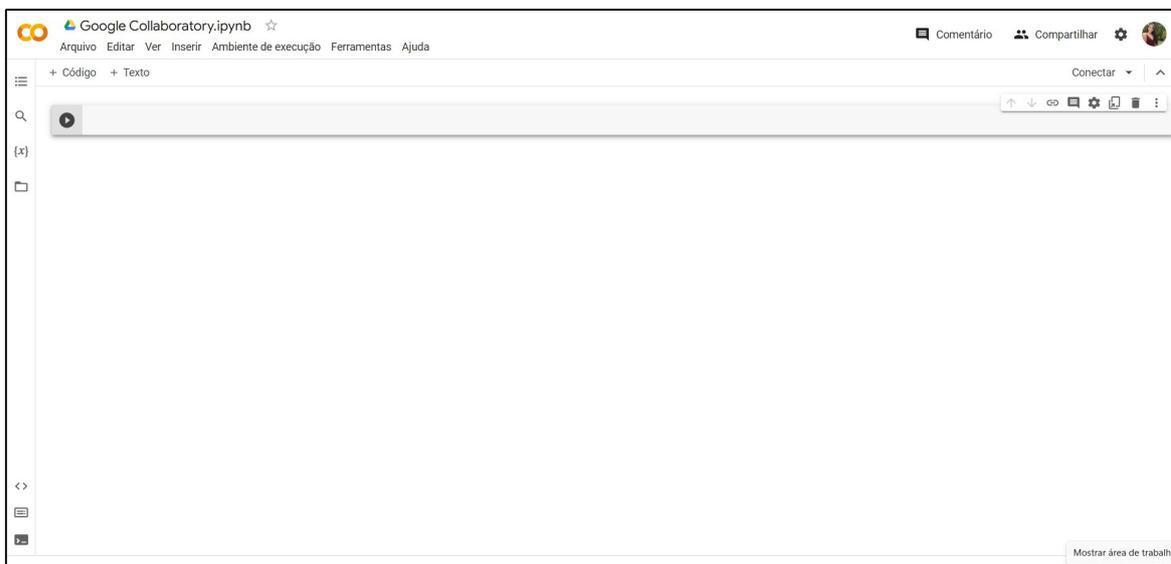
Segundo Tamiminia et. al. (2020), o GEE permite operações por pixel, banda ou coleções de imagens e diversas operações matemáticas de números reais, como operações com matrizes e processamento de pixels e outros.

O Google Colaboratory (Colab) é um serviço de armazenamento em nuvem voltado para criação e execução de códigos em Python em notebooks (que nada mais é que células de códigos agrupadas), sendo hospedado pelo Jupyter Notebook, onde a diferença entre ambos é que o Jupyter Notebook necessita de instalação na máquina e conseqüentemente é necessário realizar configurações na mesma, diferentemente do Colab que é diretamente feito através do navegador, (DA SILVA, 2020).

O Colab permite que seja realizado processamentos de dados em nuvem, com acesso a dados hospitalares de forma prática e veloz. Segundo Bisong (2019),

através do colab é possível acessar as principais bibliotecas do Python disponíveis, sendo possível de forma fácil o compartilhamento de scripts e a criação colaborativa entre os usuários.

Figura 2 - Ambiente de execução do Google Colaboratory



Fonte: Autora (2023).

O serviço é feito diretamente em um navegador, onde a facilidade está que o usuário não tem a necessidade de realizar nenhum tipo de instalação de software em sua máquina. Contando com as seguintes possibilidades:

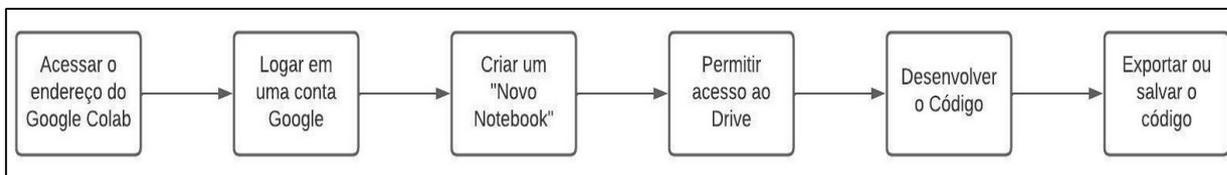
- Não existe a necessidade de configuração na máquina;
- Acesso gratuito a GPUs;
- Compartilhamento Fácil.

O Colab possibilita que o usuário seja capaz de ler, desenvolver e rodar seus códigos em seus notebooks, de maneira totalmente online e por ser feito através de um navegador o usuário tem a liberdade de modificá-los em qualquer momento e em qualquer máquina. Além de toda a possibilidade computacional já mencionada, o Colab permite que isso seja salvo na nuvem do usuário "drive".

A nuvem de computadores da Google permite que os códigos desenvolvidos pelos usuários sejam processados de forma gratuita, com um

excelente desempenho de processo e com uma quantidade bem grande de dados, (DA SILVA, 2020).

Figura 3 - Diagrama de blocos apresentando a forma de uso do Colaboratory



Fonte: Autora (2023).

Para que seja possível a integração do GEE ao Google Colab é necessária realizar a configuração da API Python Earth Engine no Colab. Dessa forma é possível trabalhar com os dados do GEE e realizar todo o processamento do mesmo na linguagem Python.

A integração das plataformas GEE e Google Colab proporcionam vantagens significativas para trabalhos com imagens de satélite e análise espacial. Incluindo vantagens como:

I. Acesso a dados abrangentes: O GEE oferece um amplo conjunto de dados de sensoriamento remoto, incluindo imagens de satélite multiespectrais, dados de radar, informações topográficas e muito mais. Esses dados são disponibilizados de forma gratuita ou por uma taxa mínima, permitindo o acesso a informações geoespaciais valiosas para uma variedade de aplicações.

II. Processamento escalável: O GEE possui uma infraestrutura de computação em nuvem poderosa, o que permite o processamento paralelo e distribuído de grandes volumes de dados geoespaciais. Isso significa que algoritmos complexos e intensivos em recursos podem ser executados de forma eficiente e rápida, reduzindo o tempo de processamento e aumentando a produtividade.

III. Ferramentas de análise espacial: O GEE oferece uma ampla gama de ferramentas e bibliotecas de análise espacial, permitindo a execução de tarefas como classificação de imagens, segmentação de imagens, detecção de mudanças, interpolação espacial, análise de padrões e muito mais. Essas

ferramentas facilitam a realização de análises complexas e a extração de informações relevantes das imagens de satélite.

IV. Colaboração e compartilhamento de código: O Google Colab permite que os usuários criem e compartilhem notebooks interativos, nos quais é possível escrever e executar código Python, além de documentar e visualizar os resultados. Isso facilita a colaboração em tempo real entre pesquisadores e a disseminação do conhecimento, permitindo que outros usuários reproduzam e ampliem a pesquisa realizada.

Autores como: Gorelick et. al. (2017), Rodríguez-Veiga et. al. (2017), Gautam et. al. (2022) abordam trabalhos que utilizam as plataformas para o processamento de imagens e sua análise espacial.

2.3 LINGUAGEM PYTHON

Ganhando destaque significativo nas últimas décadas e emergindo como uma das linguagens mais populares e versáteis no campo da ciência da computação, o Python é uma linguagem conhecida por sua sintaxe simples e legível.

Considerada uma linguagem de alto nível, o Python foi criado por Guido van Rossum em 1991, que se destaca por sua sintaxe clara, legível e facilidade de uso. Ela é interpretada, orientada a objetos e possui uma série de recursos avançados, como módulos, exceções e tipagem dinâmica. Além disso, o Python oferece tipos de dados de alto nível e uma extensa biblioteca padrão que abrange diversas áreas de processamento (MENEZES, 2010).

Com base nas informações fornecidas pela Python Software Foundation (PSF) em 2019, essa linguagem tornou-se extremamente popular entre os desenvolvedores, sendo amplamente utilizada em uma variedade de áreas, que podem se destacar com aplicações em web, ciências de dados e IA (Inteligência artificial), automação e sistemas embarcados. Essa popularidade é devido ao seu ecossistema rico e a vasta quantidade de bibliotecas disponíveis, que a tornam uma escolha versátil e poderosa para inúmeros projetos.

Millman e Aivazis (2011) afirmam que, a riqueza de software do Python, combinada com sua simplicidade e flexibilidade, faz dela uma escolha ideal para diversas atividades, como desenvolvimento web, análise e automação de dados.

Na literatura podemos encontrar em McKinney (2010), o papel do Python como uma linguagem fundamental para a análise de dados, impulsionada principalmente pelo desenvolvimento da biblioteca pandas, que oferece estruturas poderosas para análise de dados.

O Python tem uma característica forte de possuir uma comunidade ativa de colaboradores e usuários, que compartilham seus códigos e soluções. Essa colaboração torna a linguagem altamente interativa e eficaz, permitindo que desenvolvedores aproveitem as resoluções criadas por outros usuários para problemas comuns e raros (MENEZES, 2010).

Segundo Lemenkova (2022), o Python é uma linguagem de alto nível com muito sucesso, sendo uma ferramenta poderosa para as ciências de dados, permitindo a possibilidade de diversas tarefas para o processamento de imagens e análises espaciais através da integração de diversas bibliotecas. Ainda segundo o autor, é possível realizar calibração e correlação, reconstrução de imagem, identificar e extrair características, modelagens e muitas outras aplicações.

Clewley (2014) afirma que trabalhar com a linguagem Python permite que o sistema seja facilmente ampliado através das diversas bibliotecas disponíveis, como funções estatísticas.

Em resumo, o Python é uma linguagem de programação altamente acessível, com uma comunidade engajada e uma grande variedade de recursos, tornando-a uma escolha popular para projetos em diversas áreas da computação. Seja para desenvolvedores web, análise de dados, IA, automação ou qualquer outra aplicação, o Python se destaca como uma ferramenta poderosa e flexível.

2.4 SENSORIAMENTO REMOTO E PROCESSAMENTO DE IMAGENS

O sensoriamento remoto é uma técnica que envolve a aquisição de informações sobre a superfície terrestre sem que haja um contato direto com ela. Segundo Novo e Ponzoni (2010), o termo sensoriamento remoto se refere a um conjunto de técnicas destinadas a obtenção de informação sobre objetos, sem que haja contato físico com eles.

Meneses e Almeida (2012) utilizam uma definição mais clássica, definida como: uma técnica de obtenção de imagens dos objetos da superfície terrestre sem que haja contato físico de qualquer espécie entre o sensor e o objeto.

A aquisição de informação sobre os alvos é realizada por meio de sensores instalados em satélites que são capazes de coletar e registrar a radiação refletida ou emitida pelos objetos provenientes da superfície terrestre. Cada tipo de sensor tem suas próprias vantagens e limitações, e a escolha do sensor depende do objetivo da análise e das características da área de estudo. É necessário de três elementos básicos para a aquisição das informações por meio de sensores, sendo elas: fonte ou energia radiante, objeto de visada e sistema de imageamento óptico e detector.

Outras definições e conceitos que fazem parte dessa temática colaboram para o tema em questão. Portanto, os sensores também se diferenciam pelas suas características, como: resoluções espaciais, temporais, espectrais e radiométricas.

Sendo a resolução espacial relacionada ao tamanho do pixel do terreno, quanto menor a área do pixel, maior a resolução espacial. Portanto, ela é responsável por distinguir e medir os alvos.

A resolução temporal está diretamente ligada a frequência da passagem do sensor em um mesmo local, determinada por um intervalo de tempo. Portanto, essa resolução é importante para analisar as mudanças que ocorrem da superfície terrestre.

A resolução espectral refere-se a largura da faixa do espectro eletromagnético a qual o sensor é sensível, faixas mais largas conferem a uma resolução espectral menor. Segundo Novo (2010), ele define como resolução espectral a medida da largura das faixas espectrais e da sensibilidade do sistema sensor em distinguir entre dois níveis de intensidade do sinal de retorno.

Enquanto, a resolução radiométrica é a capacidade do sensor em distinguir pequenas variações na intensidade da radiação observada. Schowengerdt (1983), afirma que, existe uma definição chamada ND (Número Digital), onde a radiância de cada pixel para uma codificação digital, sendo um valor expresso em *bits*.

Segundo o DPI/INPE (2006), os dados de sensoriamento remoto possuem formatos de imagens digitais, podendo ser definidas como, uma função

bidimensional da radiação refletida ou emitida por uma cena, na forma de ' $I(x, y)$ '; onde temos que: ' I ' representa a intensidade da imagem a cada coordenada espacial (x, y) .

É possível ter diversas aplicações a partir do sensoriamento remoto, abrangendo diversas áreas como: monitoramento ambiental, gestão de recursos naturais, estudos climáticos, planejamento urbano e agrícola, detecção de mudanças e análises de desastres naturais. As imagens podem ser adquiridas em diferentes regiões do espectro eletromagnético, como regiões do visível, termal, com diferentes resoluções espectrais e espaciais.

As imagens adquiridas com o objetivo de análise necessitam passar por um pré-processamento, de acordo com o método desejado. Portanto, inicialmente deve ser definido a problemática para a resolução do problema e consequentemente a obtenção do resultado esperado.

O processamento de imagens espaciais enfrenta vários desafios, principalmente devido ao grande volume de dados gerados pelos sensores remotos. Esses desafios abrangem áreas como armazenamento, pré-processamento, classificação, extração de informações relevantes e análise espacial de conjuntos de dados volumosos. Esses desafios podem ser explicados como:

I. Armazenamento: Os sensores remotos produzem uma enorme quantidade de dados, especialmente quando se trata de imagens de alta resolução espacial e espectral. O armazenamento eficiente desses dados é um desafio significativo, exigindo infraestrutura de armazenamento robusta e escalável. Isso inclui a necessidade de sistemas de armazenamento distribuído, uso de técnicas de compressão de dados, organização adequada dos arquivos e o estabelecimento de estratégias de backup e recuperação de dados.

II. Pré-processamento: Antes de realizar qualquer análise, as imagens espaciais geralmente precisam passar por etapas de pré-processamento para corrigir distorções, remover ruídos e garantir que os dados estejam prontos para análise. O pré-processamento pode incluir correção geométrica, correção radiométrica, calibração atmosférica, fusão de imagens de diferentes sensores e correção de efeitos atmosféricos. Essas etapas exigem algoritmos e métodos específicos, bem como conhecimento especializado para garantir a qualidade e a precisão dos dados processados.

III. Classificação: A classificação de imagens espaciais é um processo fundamental para identificar e mapear diferentes objetos e características na imagem. No entanto, classificar grandes volumes de dados pode ser desafiador.

IV. Extração de informações relevantes: A extração de informações relevantes das imagens espaciais envolve identificar e extrair características específicas de interesse, como áreas urbanas, corpos d'água, vegetação ou estruturas geológicas. Isso requer o desenvolvimento de algoritmos e técnicas para realizar a segmentação de imagem, detecção de bordas, extração de características espectrais e texturais, entre outros. Esses métodos visam identificar padrões e características relevantes nos dados para posterior análise e interpretação.

V. Análise espacial de conjuntos de dados volumosos: A análise espacial envolve a aplicação de técnicas para estudar e entender a relação entre os dados espaciais. Com grandes conjuntos de dados, a análise espacial pode se tornar complexa e computacionalmente intensiva. Ela pode envolver a execução de operações como interpolação espacial, análise de proximidade, análise de padrões espaciais, modelagem espacial e criação de mapas temáticos.

Nesse contexto, a integração do GEE e Google Colab no processamento de imagens visa facilitar a realização dessas análises, permitindo o acesso a um grande volume de dados geoespaciais, a implementação de algoritmos e a execução de técnicas de análise de forma prática e escalável.

2.5 LANDSAT- 8 E SISTEMA OLI

De origem Norte Americana, o programa Landsat teve seu lançamento na segunda metade da década de 60, a partir de um projeto desenvolvido pela Administração Nacional de Aeronáutica e Espaço (NASA). Desde o seu lançamento, foram lançados 8 satélites da série, todos utilizando o princípio de imagem multiespectral.

O Landsat-8 é um satélite de sensoriamento remoto pertencente ao Programa *Landsat*, uma série de missões espaciais lideradas pela NASA (Agência Espacial Americana) e pelo USGS (Serviço Geológico dos Estados Unidos).

O satélite foi lançado em 11 de fevereiro de 2013, como sucessor da série *Landsat* que teve início em 1972. Seu principal objetivo é capturar imagens da superfície terrestre para fins de monitoramento e estudos ambientais, (Embrapa, 2022).

O Landsat-8 é equipado com o sensor Operational Land Imager (OLI) e o sensor Thermal Infrared Sensor (TIRS). Esses sensores capturam imagens em várias bandas espectrais, o que permite uma ampla variedade de análise e aplicações.

A *Ball Aerospace & Technologies Corporation* é a responsável pelo *Operational Land Imager (OLI)*, é o sistema que está a bordo do Landsat-8. Ele é capaz de capturar imagens em nove bandas espectrais, incluindo bandas na região do visível, infravermelho próximo e médio, bem como duas bandas no espectro do infravermelho térmico.

As imagens advindas do OLI possuem resolução espacial de 15m para a banda pancromática e 30 m para as bandas multiespectrais ao longo de uma faixa de 185 km de largura, com uma resolução temporal de 16 dias, pelas suas características, o OLI possibilita estudos em amplas áreas da superfície terrestre, possibilitando distinguir centros urbanos, áreas hídricas, e áreas com vegetação densa.

A tabela abaixo apresenta suas principais características:

Tabela 1 - Características do Sensor OLI

Características	Dado	Características	Dado
Bandas Espectrais	0,43-0,45µm (DBV)	Resolução espacial Largura da Faixa Revisita Quantização	15m
	0,45-0,51µm (B)		
	0,53-0,59µm (G)		(Pan)/30
	0,64-0,67µm (R)		m
	0,85-0,88 µm (NIR)		185 km
	1,57-1,65 µm (SWIR)		
	2,11-2,29 µm (SWIR)		
	0,50-0,68 µm (Pan)		16 dias
	1,36-1,38 µm (Cirrus)		
			16 bits

Fonte: USGS (2017).

2.6 ÍNDICE DE VEGETAÇÃO POR DIFERENÇA NORMALIZADA (NDVI)

O NDVI (*Normalized Difference Vegetation Index*) é um índice de sensoriamento remoto amplamente utilizado (Cohen et al., 2003) para avaliar e quantificar a presença de vegetação, em uma determinada área. Esse índice é útil em aplicações de monitoramento ambiental, estudos agrícolas, gestão de recursos naturais e análises climáticas.

O NDVI é um índice que fornece informações sobre saúde e densidade da vegetação em área específica. É baseado na diferença entre a reflectância da luz vermelha (banda próxima ao infravermelho) e a reflectância da luz infravermelha próxima. A vegetação saudável geralmente absorve bem a luz vermelha e reflete mais a luz infravermelha próxima devido a fotossíntese, o que resulta em valores positivos de NDVI. É possível calcular o NDVI, segundo a equação (1):

$$NDVI = (NIR - RED)/(NIR + RED) \quad (1)$$

Onde:

- NIR: reflectância na banda do infravermelho próximo;
- RED: reflectância na banda do vermelho;

Os valores de NDVI variam de -1 a +1, com valores positivos indicando maior presença de vegetação saudável e valores negativos indicando áreas sem vegetação ou com pouca vegetação.

2.7 ÍNDICE DE VEGETAÇÃO AJUSTADO AO SOLO (SAVI)

O SAVI (*Índice de vegetação ajustado ao solo*), desenvolvido por Huete (1988), foi desenvolvido para suprir algumas limitações observadas no NDVI. Portanto, o SAVI tem como objetivo melhorar os valores encontrados no NDVI, a

equação se diferencia do NDVI no termo $(1 + L)$, onde temos L como uma constante, é possível calcular conforme a equação (2):

$$SAVI = (1 + L) \times (NIR - R) / (NIR + R + L) \quad (2)$$

Onde:

- NIR: é a banda que corresponde a reflectância do infravermelho próximo;
- R: é a banda que corresponde a reflectância do vermelho;
- L: é um fator de ajuste.

O fator L pode apresentar valores variáveis, segundo Huete (1988), os valores podem variar de 0 a 1, onde para densidades baixas de vegetação podem utilizar o valor 1, para médias densidades de vegetação o valor 0,5 e para altas densidades de vegetação o valor 0,25.

O fator L é considerada uma constante empírica que possui a finalidade de minimizar a sensibilidade do índice de vegetação as variações de refletância dos tipos do solo.

2.8 ÍNDICE DE ÁGUA DE DIFERENÇA NORMALIZADA (NDWI)

O NDWI (*Índice de Água de Diferença Normalizada*) é utilizado para destacar as características da água, dessa forma, a reflectância do solo e da vegetação é consideravelmente reduzida, permitindo que a massa de água se destaque (MCFEETRS, 1996).

Segundo Jackson et. al. (2004) afirmam que, o NDVI saturava ao verificar mudanças na biomassa e nos estresses de umidade das plantas, enquanto, a representação do NDWI apresentava melhor as mudanças no conteúdo de umidade das vegetações.

O NDWI é calculado pela diferença normalizada entre o infravermelho próximo (NIR) e o verde (Green) dividido pela soma do infravermelho próximo (NIR) e o verde (Green), é possível calcular segundo a equação (3):

$$NDWI = (GREEN - NIR) / (GREEN + NIR) \quad (3)$$

Onde:

- NIR: reflectância na banda do infravermelho próximo;
- GREEN: reflectância na banda do verde;

Seus resultados variam de -1 a 1, onde os valores positivos indicam a presença de água, enquanto valores negativos indicam a presença de vegetação, McFeeters (1996) definiu zero como o limiar.

2.9 ÍNDICE DE ÁREA CONSTRUÍDA POR DIFERENÇA NORMALIZADA (NDBI)

O NDBI (*Índice de área construída por diferença normalizada*), é um índice utilizado para destacar o adensamento de construções e machas urbanas (Zha et al., 2003), é possível calcular conforme a equação (4):

$$NDBI = (IVM - IVP)/(IVM + IVP) \quad (4)$$

Onde:

- IVM: é a banda que corresponde ao infravermelho de ondas curtas ou médio;
- IVP: é a banda que corresponde ao infravermelho próximo;

Esses índices são uteis para monitorar a saúde da vegetação, monitoramento da água e do crescimento urbano, contribuindo para diversas aplicações relacionadas ao meio ambiente e ao uso sustentável dos recursos naturais.

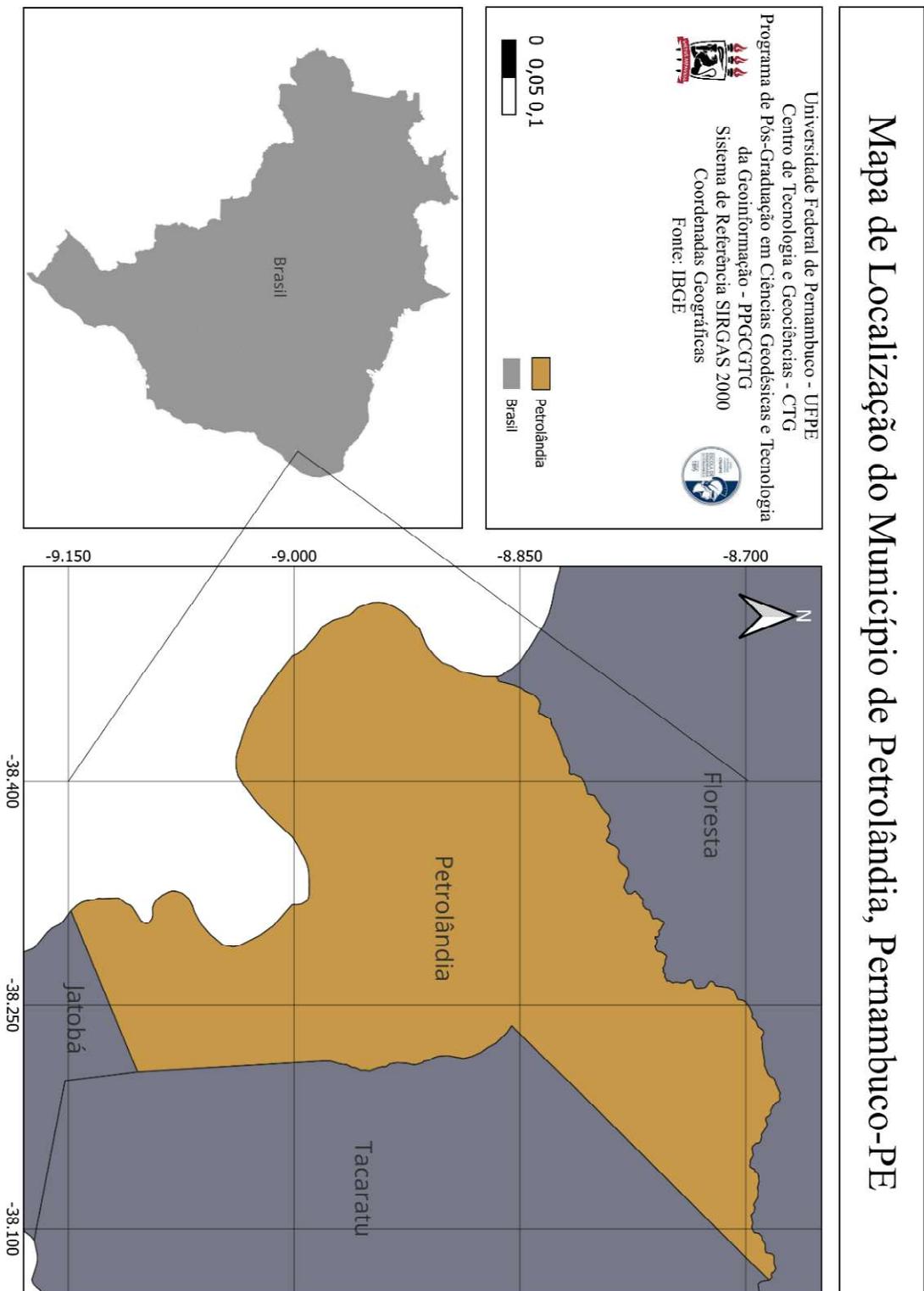
3 ÁREA DE ESTUDO

Situado no estado de Pernambuco, na região Nordeste do Brasil, Petrolândia (Figura 4) é um município do sertão de Pernambuco, que possui uma área territorial de aproximadamente 1056,6 km², localizado as margens do Rio São Francisco, à uma distância aproximada de 430km da capital Recife (IBGE, 2023). De acordo com o IBGE (2021), sua população é aproximadamente de 36.548 habitantes.

Situado a 319 metros de altitude, limita-se ao norte com município de Floresta, ao sul com município de Jatobá e o estado da Bahia, a oeste com o estado da Bahia e a Leste com município de Tacaratu, possuindo as seguintes coordenadas geográficas: Latitude 9° 4' 12" Sul, Longitude 38° 16' 54" Oeste.

Com um clima predominantemente quente na região, caracteriza-se por semiárido, com uma temperatura média anual em trono de 25°C, com estação chuvosa concentrada no período de janeiro, fevereiro e março, enquanto os períodos de seca se concentram em setembro, outubro e novembro. A vegetação é predominantemente a caatinga, que se caracteriza por possuir espécies vegetais com alta capacidade de retenção de água. Essas condições climáticas influenciam diretamente na vegetação predominante na região e nas atividades econômicas locais.

Figura 4 - Mapa de localização da área de estudo



Fonte: Autora (2023).

4 MATERIAIS E MÉTODOS UTILIZADOS

Nesta seção serão apresentados os materiais e métodos envolvidos para realização do trabalho.

4.1 DESCRIÇÃO DOS MATERIAS UTILIZADOS

Os materiais aqui apresentados serão divididos em dois grupos, esses grupos foram intitulados de dados e tecnologias.

Nesta pesquisa foram utilizados os seguintes materiais considerados como dados:

- I. Imagens de satélite do Landsat-8 na faixa do visível e infravermelho próximo em formato .tiff;
- II. Shapefile do IBGE, com a delimitação da área de interesse.

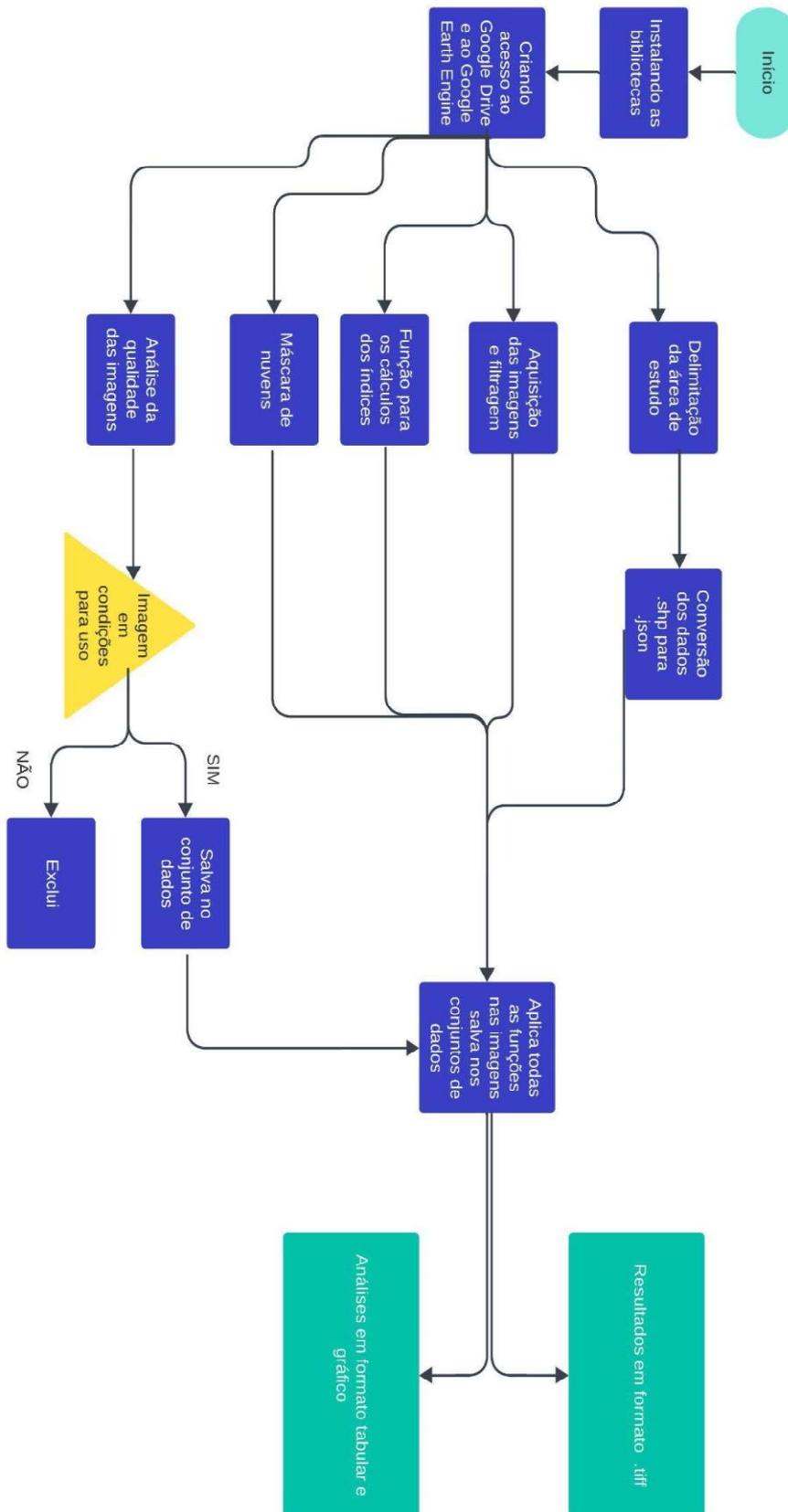
Nesta pesquisa foram utilizados os seguintes materiais considerados como tecnologias:

- I. Desenvolvimento do trabalho em linguagem Python e utilização das bibliotecas;
- II. Utilização do GEE e Google Colaboratory.

4.2 DESCRIÇÃO DA METODOLOGIA

A metodologia aplicada no desenvolvimento desta pesquisa pode ser visualizada no fluxograma Figura 5:

Figura 5 - Fluxograma da metodologia



A partir da revisão bibliográfica dos índices que serão utilizados e do uso da ferramenta do Colab integrado ao GEE, vamos verificar a potencialidade do processamento a partir da metodologia empregada.

É importante ressaltar que para o desenvolvimento da metodologia, foram considerados alguns tópicos para a obtenção dos resultados:

- O período utilizado para a análise e geração dos resultados dos índices calculados foram de 01 de janeiro de 2023 a 31 de dezembro de 2023.
- Para a análise dos índices em forma tabular, serão gerados gráficos e tabelas estatísticas com os dados obtidos para que seja possível observar o comportamento durante o ano de 2023.
- Para os mapas foi levado em consideração dois períodos para os índices, período: chuvoso e seco. Considerando o ano de 2023 e as imagens disponíveis nesse período.

4.2.1 Bibliotecas e acesso

A primeira etapa desempenha um papel essencial ao preparar o ambiente de execução do código Quadro 1, onde é feita toda a configuração do acesso e instalação das bibliotecas necessárias.

Quadro 1 - Código para instalação e acesso

```
#Instalando as bibliotecas

!pip install earthengine-api

!pip install geopandas

!pip install geemap

!pip altair

!pip install --upgrade earthengine-api

##Acesso ao Drive
```

```
from google.colab import drive

drive.mount('/content/drive', force_remount=True)

import ee

ee.Authenticate()

ee.Initialize(project='sabrina-411200')

##Importe as bibliotecas e pacotes que utilizaremos

import geopandas as gpd

import json

import pandas as pd

import geemap

import geemap.colormaps as cm

import altair as alt

import numpy as np
```

Fonte: Autora (2023).

Essa fase é responsável por instalar e importar as bibliotecas que permitem a interação com os dados e todas as funcionalidades do GEE, além de possibilitar recursos de visualização e manipulação das informações geoespaciais em formato DataFrame.

Com as bibliotecas devidamente instaladas, é necessário estabelecer o acesso ao Google Drive, é crucial que essa etapa seja respeitada, pois ela permite que seja integrada o ambiente de execução do Colab ao Drive, onde o usuário realiza a autenticação e acessa todos os arquivos armazenados, fazendo toda a atividade diretamente no código, facilitando o acesso direto e o armazenamento de qualquer resultado gerado pelo código.

Ambas as etapas são essenciais para garantir a preparação adequada do ambiente de execução, proporcionando todas as ferramentas necessárias, então iniciar de forma estruturada e integrada contribui para eficácia e conveniência do processo.

4.2.2 Delimitação da área de estudo

A delimitação da área de estudo realizada dentro do ambiente do Colab necessita que seja carregado um arquivo que contenha os dados geoespaciais (*shapefile* - *.shp*) que estão localizados no *Google Drive*, após isso, as informações do *shapefile* precisam ser convertidas em formato *.JSON*. Ambos são arquivos vetoriais geoespaciais, a diferença é que o *.json* permite ser aplicado em plataformas web. Portanto, é exibido esses dados em um mapa interativo usando a biblioteca *geemap* no ambiente Colab, a exibição permite que o usuário identifique se a delimitação ocorreu como esperado. Para que esse código fosse executado de maneira eficiente foi necessário utilizar as seguintes bibliotecas:

- *earthengine-api*: o objetivo é que seja feita a interação com a API do GEE, permitindo o acesso a dados e recursos disponíveis no Google Earth através do Python;
- *geopandas*: responsável por manipular e realizar análise de dados geoespaciais, incluindo leitura e escrita de arquivos no formato *shapefile*;
- *geemap*: Criação de mapas interativos usando o GEE;
- *altair*: Criação de visualizações de dados interativas e declarativas;
- *json*: Utilizada para trabalhar com dados no formato JSON (JavaScript Object Notation);

Quadro 2 - Código para delimitação da área de estudo

```
##Montando o acesso ao google drive  
  
from google.colab import drive  
  
drive.mount('/content/drive', force_remount=True)
```

```
# Carregando o arquivo shp

gdf =
gpd.read_file('/content/drive/MyDrive/Recife/Petrolandia.shp
')

##Filtrando as colunas que temos mais interesse

roi = gdf[['NM_MUN', 'geometry']]

roi.head()

##convertendo de shp para to json

roi = roi.to_json()

roi = json.loads(roi)

roi = roi['features']

##Verificando o processo

roi

##Definindo a região

region = ee.FeatureCollection(roi)

# Criando um mapa com a região de estudo

Map = geemap.Map(center=[-8.28, -35.97], zoom=10)
```

```
# Adicionando a FeatureCollection ao mapa com legenda
```

```
Map.addLayer(region, {}, 'Área de Estudo')
```

```
# Adicionando título ao mapa
```

```
Map.add_text('Área de Estudo', fontsize='16px',  
font_weight='bold', position='topleft')
```

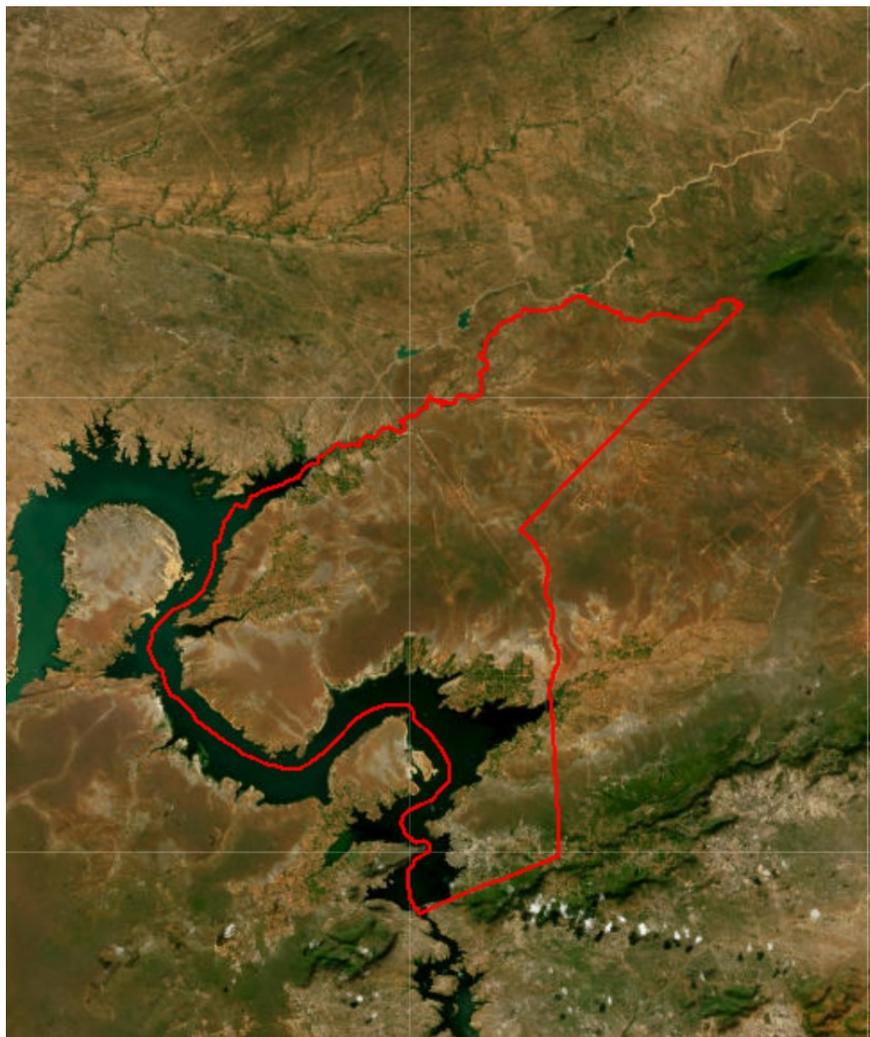
```
# Mostrando o mapa
```

```
Map
```

Fonte: Autora (2023).

Portanto, o código vai carregar um arquivo *shapefile*, converter as informações do *shapefile* em um formato apropriado para o *Earth Engine*, criando uma visualização interativa, essa visualização permite que seja exibido o polígono da área de estudo.

Figura 6 - Delimitação da área de estudo



Fonte: Autora (2023).

4.2.3 Aquisição de imagens

A base de dados do GEE abriga imagens desde a operação do Landsat-1 até o Landsat-9. Os dados gerados pelo Landsat são organizados em coleções, que determinam o nível de qualidade da imagem, tendo uma consistência garantida no uso destas, independente dos instrumentos utilizados na aquisição (Landsat Missions, [S.D.]).

Para esse projeto, serão utilizadas imagens fornecidas pelo Landsat-8. Tais imagens são pertencentes a coleção 2, nível 1. Com a definição do conjunto a ser utilizado, o próximo passo consiste em filtrá-lo por área e período.

Quadro 3 - Código para importar as imagens do Landsat-8

```
#Importando imagem do Landsat 8 collection e aplicando
filtros

l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')\
    .filterBounds(region)\
    .filterDate('2023-01-01','2023-12-31')\
    .map(maskL8sr)\
    .map()
```

Fonte: Autora (2023).

4.2.4 Máscara de nuvens

O landsat-8 é um satélite que captura imagens em várias bandas espectrais, incluindo bandas ópticas (visíveis e infravermelhas) e bandas térmicas. Entretanto, em algumas imagens, podem ocorrer a presença de nuvens, que podem prejudicar a interpretação dos objetos terrestres, especialmente quando se deseja analisar alterações ou caracterizar uma área.

A etapa consistiu em aplicar um código intitulado de máscara de nuvens nas imagens do landsat-8 de forma a preparar as imagens que serão utilizadas. O propósito é eliminar os pixels afetados por nuvens e sombras, permitindo uma melhor interpretação e análise das informações contidas na imagem a superfície da Terra.

O código apresentado na Tabela 3, apresenta uma função intitulada de *maskL8sr*, utilizada para criar uma máscara que elimina os pixels afetados por nuvens e sombras na imagem. As etapas consistem na seguinte forma na função abaixo:

- *qaMask*: o código seleciona a banda *QA_PIXEL*, que contém informações sobre a qualidade da imagem, e aplica operação bitwise para extrair os bits que representam as informações de nuvens (os 5 bits menos

significativos). Os pixels com valor 0 na máscara são considerados livres de nuvens;

- *saturationMask*: o código seleciona a banda *QA_RADSAT*, que contém informações sobre saturação radiométrica (pixels saturados), e cria uma máscara que considera apenas os pixels com valor 0, indicando que não há saturação radiométrica;

- *opticalBands* e *thermalBands*: o código faz algumas transformações nas bandas espectrais. Para as bandas ópticas, ele aplica fatores de escala e deslocamento para converter os valores digitais em refletância (unidade adimensional entre 0 e 1). Para as bandas térmicas, ele aplica fatores de escala e deslocamento para converter os valores digitais em temperatura de brilho em kelvin;

- *return*: o código retorna a imagem original, adicionando as bandas processadas *opticalBands* e *thermalBands* e aplicando as máscaras *qaMask* e *saturationMask*. A máscara *qaMask* é aplicada para eliminar os pixels afetados por nuvens e sombras, enquanto a máscara *saturationMask* é aplicada para eliminar os pixels saturados;

Quadro 4 - Código para remoção de nuvens

```
##Máscara de nuvens

def maskL8sr(image):

qaMask = image.select('QA_PIXEL').bitwise_and(int('11111',
2)).eq(0)

saturationMask = image.select('QA_RADSAT').eq(0)

opticalBands =
image.select('SR_B.').multiply(0.0000275).add(-0.2)

thermalBands =
image.select('ST_B.*').multiply(0.00341802).add(149.0)

return image.addBands(opticalBands, None, True)\
.addBands(thermalBands, None, True).updateMask(qaMask)\
.updateMask(saturationMask)
```

Fonte: Autora (2023).

Assim, o resultado é uma imagem com as bandas processadas e sem pixels afetados por nuvens, sombras e saturação radiométrica, tornando-as mais adequadas para análise e interpretação de características terrestres, como vegetação, uso do solo, recursos hídricos e entre outros.

4.2.5 NDVI, SAVI, NDWI E NDBI

Para realizar o cálculo NDVI, SAVI, NDWI e NDBI foram criadas funções que recebem uma imagem de satélite, e assim realiza-se o cálculo para essa imagem, adiciona as bandas necessárias, recorta a imagem baseada na região específica e mantém as propriedades importantes das bandas na imagem resultante.

Para que o código funcione de forma correta, é importante que a imagem que vai ser utilizada como entrada tenham as bandas SR_B5 e SR_B4 para o cálculo do NDVI, enquanto que para o SAVI foram necessárias as bandas SR_B5 e SR_B4, enquanto para o NDWI é necessário que tenha as seguintes bandas SR_B5 e SR_B3, para o NDBI é necessário que tenha as seguintes bandas SR_B5 e SR_B6. A variável da região deve estar corretamente definida como a área do recorte.

Quadro 5 - Função para calcular o NDVI

```
##Função NDVI (ÍNDICE DE VEGETAÇÃO NORMALIZADA)

def NDVI (img):

    Ndvi_image = img.normalizedDifference(['SR_B5',
    'SR_B4']).rename('NDVI')

    return
img.addBands(Ndvi_image).clip(region).copyProperties(img,
["system:time_start"])
```

Fonte: Autora (2023).

Quadro 6 - Função para calcular o SAVI

```
# Função SAVI

def SAVI(img):

    savi = img.expression(

        '(1 + L) * (NIR - RED) / (NIR + RED + L)',

        {

            'NIR': img.select('SR_B5'), # Banda 5:
Infravermelho próximo

            'RED': img.select('SR_B4'), # Banda 4:
Vermelho

            'L': 0.5 # Fator de ajuste

        }

    ).rename('SAVI')
```

```

# Adicionando a banda SAVI

img = img.addBands(savi)

# Recortando na região de interesse

img = img.clip(region)

# Copiando propriedades e adicionando a data formatada

date =
ee.Date(img.get('system:time_start')).format('YYYY-MM-dd')

img = img.set('date', date)

return img

```

Fonte: Autora (2023).

Quadro 7 - Função para calcular o NDWI

```

##Função NDWI

def NDWI(img):

    Ndwi_image = img.normalizedDifference(['SR_B3',
'SR_B5']).rename('NDWI')

    return
img.addBands(Ndwi_image).clip(region).copyProperties(img,
["system:time_start"])

```

Fonte: Autora (2023).

Quadro 8 - Função para calcular o NDBI

```

def NDBI(img):

    Ndbi_image = img.normalizedDifference(['SR_B5',
'SR_B6']).rename('NDBI')

```

```

img =
img.addBands(Ndbi_image).clip(region).copyProperties(img,
["system:time_start"])

date =
ee.Date(img.get('system:time_start')).format('YYYY-MM-dd')

img = img.set('date', date)

return img

```

Fonte: Autora (2023).

4.2.6 Conversão de dados

Para que fosse possível realizar análises estatísticas e gráficos do comportamento da série analisada, foi necessário converter as informações da imagem do *Earth Engine* em uma tabela reduzida. Em seguida, a função é aplicada a uma coleção de imagens que retorna uma tabela de dados com informações de média, mínimo, máximo, ano, mês, data e nome do município (propriedades da imagem) para cada pixel na região de estudo.

Quadro 9 - Código que converte as informações de imagem em tabela

```

##Função para converter informações da imagem em tabela

def reduce(image):

serie_reduce = image.reduceRegions(**{

'collection': region,

'reducer': ee.Reducer.mean().combine(**{

'reducer2': ee.Reducer.min(),

'sharedInputs': True

}).combine(**{

'reducer2': ee.Reducer.max(),

```

```

        'sharedInputs': True

    }),

    'scale': 30

})

date = image.get('date')

year = ee.Date(date).get('year')

month = ee.Date(date).get('month')

serie_reduce = serie_reduce.map(lambda f:
f.set({'year': year})) \

    .map(lambda f: f.set({'date': date})) \

    .map(lambda f: f.set({'month': month}))

return serie_reduce.copyProperties(image,
["system:time_start"])

##Aplicando a função de redução na Coleção

data_reduce = 18.select('NDVI').map(reduce) \

    .flatten() \

    .sort('date', True) \

    .select(['month', 'date', 'year', 'mean', 'min',
'max', 'NM_MUN'])

##Verificando os dados e as propriedades

```

```
print(data_reduce.first().propertyNames().getInfo())  
print(data_reduce.size().getInfo())
```

Fonte: Autora, 2023.

Em resumo, o código define uma função de redução para calcular as estatísticas (média, mínimo e máximo) para cada imagem da coleção em relação a região de interesse. Em seguida, aplica essa função a cada imagem, criando uma tabela reduzida com informações de média, mínimo, máximo, ano, mês, data e nome do município para cada pixel da região de interesse. Com esses dados obtidos, é aplicado aos dados uma ferramenta para entender a distribuição dos dados e identificar se existem dados com anomalias.

Portando a rotina abaixo ordenou os dados, e a partir do IQR (Intervalo entre Quartis) identificamos os dados que diferem do padrão esperado. Na rotina a lógica utilizada foi:

- Ordenar os dados, assim identificando os meses que possuíam mais de uma data;
- Calcular o primeiro quartil (Q1), que seria os valores que estavam abaixo dos 25% dos dados;
- Calcular o terceiro quartil (Q3), que seria os valores que estavam abaixo dos 75% dos dados;
- O IQR é definido pela diferença entre o Q1 e Q3.

Quadro 10 - Remoção de dados

```
# Dados agrupados por mês  
grouped_by_month = df_savi.groupby('month')  
  
# Calcular o IQR para cada mês  
iqr_data = grouped_by_month['SAVImean', 'SAVImin',  
'SAVImax'].apply(lambda x: pd.Series({  
    'mean_iqr': np.percentile(x['SAVImean'], 75) -  
np.percentile(x['SAVImean'], 25),  
    'min_iqr': np.percentile(x['SAVImin'], 75) -  
np.percentile(x['SAVImin'], 25),  
    'max_iqr': np.percentile(x['SAVImax'], 75) -
```

```

np.percentile(x['SAVImax'], 25)
}))

# Identificar e manter a data que está dentro do intervalo
definido pelo IQR para cada métrica
dates_to_keep = []
for month, group in grouped_by_month:
    month_iqr = iqr_data.loc[month]
    month_data = group.set_index('date')
    filtered_data = month_data[
        (month_data['SAVImean'] >=
(month_data['SAVImean'].median() - 1.5 *
month_iqr['mean_iqr'])) &
        (month_data['SAVImean'] <=
(month_data['SAVImean'].median() + 1.5 *
month_iqr['mean_iqr'])) &
        (month_data['SAVImin'] >=
(month_data['SAVImin'].median() - 1.5 *
month_iqr['min_iqr'])) &
        (month_data['SAVImin'] <=
(month_data['SAVImin'].median() + 1.5 *
month_iqr['min_iqr'])) &
        (month_data['SAVImax'] >=
(month_data['SAVImax'].median() - 1.5 *
month_iqr['max_iqr'])) &
        (month_data['SAVImax'] <=
(month_data['SAVImax'].median() + 1.5 *
month_iqr['max_iqr']))
    ]
    dates_to_keep.append(filtered_data.reset_index().iloc[0]
)

# Criar um novo DataFrame apenas com as datas a serem
mantidas
df_savi_filtered = pd.DataFrame(dates_to_keep)

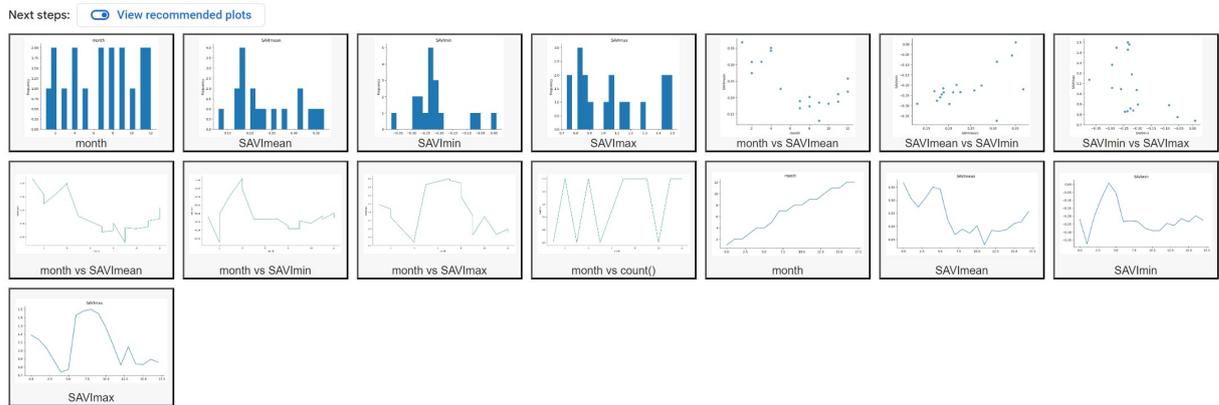
# Contar o número de datas removidas
num_removed_dates = len(df_savi) - len(df_savi_filtered)
print("Número de datas removidas:", num_removed_dates)

```

Fonte: Autora, 2023.

A partir dos dados estatísticos gerados, o código recomenda plots gráficos de visualização, que é uma forma interessante para visualizar os dados de forma mais rápida, Figura 7.

Figura 7 - Visualizações gráficas recomendadas



Fonte: Autora (2023).

Porém, foi desenvolvido outra forma para visualizar os dados a partir de gráficos que se adequam mais a necessidade dos resultados, assim podemos ver o comportamento baseado no mínimo, máximo e média para cada data analisada, e assim gerar um comportamento ao longo do ano, para isso foi realizado a seguinte rotina:

Quadro 11 - Código para visualizar os dados a partir de gráficos

```
import matplotlib.pyplot as plt

# Converter as colunas de data para o formato de data
df_savi['date'] = pd.to_datetime(df_savi['date'])

# Ordenar o DataFrame pela data
df_savi = df_savi.sort_values('date')

# Criar uma série temporal para cada métrica (média, mínimo, máximo)
plt.plot(df_savi['date'], df_savi['SAVImean'], label='Média SAVI')
plt.plot(df_savi['date'], df_savi['SAVImin'], label='Mínimo SAVI')
plt.plot(df_savi['date'], df_savi['SAVImax'], label='Máximo SAVI')

# Adicionar rótulos e legenda
plt.title('Série Temporal SAVI')
plt.xlabel('Data')
plt.ylabel('Valor SAVI')
plt.legend()

# Definir os intervalos no eixo y com incremento de 0.20
plt.yticks(
    [i / 100 for i in range(int(plt.ylim()[0] * 100),
int(plt.ylim()[1] * 100) + 1, 20)]
```

```
)  
  
# Exibir o gráfico  
plt.show()
```

Fonte: Autora (2023).

O Quadro 11, apresenta a rotina que permite a visualização dos gráficos, definindo um intervalo de incremento 0,20, onde os valores são gerados de acordo com o intervalo atual do eixo y.

Para que fosse possível ver o comportamento da série analisada de forma suavizada, foi criado um código que gera uma visualização dos dados em formato gráfico, nele utilizamos a média para observar como o gráfico de suavização se comporta, portanto, é possível observar o comportamento da média para cada mês, nele utilizamos uma linha de tendência suavizada, que podemos visualizar os padrões para cada mês e a tendência dos dados.

Quadro 12 - Código para gerar gráficos com linhas de tendência

```
# Converter as colunas de data para o formato de data (se  
ainda não estiverem)  
  
df_ndvi['date'] = pd.to_datetime(df_ndvi['date'])  
  
# Ordenar o DataFrame pela data  
  
df_ndvi = df_ndvi.sort_values('date')  
  
# Criar o gráfico de séries temporais  
  
plt.figure(figsize=(10, 6))  
  
# Plotar a série temporal para NDVI médio  
  
plt.plot(df_ndvi['date'], df_ndvi['NDVImean'],  
label='Média NDVI', color='green', marker='o')
```

```

# Adicionar tendência (pode ajustar conforme necessário)

plt.plot(df_ndvi['date'],
df_ndvi['NDVImean'].rolling(window=3).mean(),
label='Tendência', color='red', linestyle='--')

# Adicionar rótulos e legenda

plt.title('Série Temporal NDVI')

plt.xlabel('Data')

plt.ylabel('Valor NDVI')

plt.legend()

# Exibir o gráfico

plt.show()

```

Fonte: Autora (2023).

4.2.7 Visualização dos dados

O objetivo em questão é criar uma representação visual de um mapa, para a área determinada. Portanto, utilizamos os arquivos de imagem em formato *.tiff* que contém os dados de interesse. O arquivo é carregado usando a biblioteca *rasterio*, ele é processado para extrair os valores de interesse.

Com a utilização da biblioteca *matplotlib* o código cria uma visualização dos resultados. Para fornecer mais informações, o código também inclui uma escala no mapa, essa escala mostra uma estimativa de quantos em quantos metros correspondem a cada pixel no mapa, ajudando a entender o tamanho real. O código pode ser ajustável tanto para a visualização mensal como diária, abaixo é apresentada a ideia base para a visualização dos dados.

Quadro 13 - Plotagem dos resultados

```
# Carregar a imagem NDVI com rasterio

with rasterio.open(tiff_path) as dataset:

    ndvi = dataset.read(1) # Lê a primeira banda (NDVI)

# Criar a figura e o eixo norm=colors.Normalize(vmin=-1,
vmax=1)

fig, ax = plt.subplots(figsize=(10, 8))

# Plotar a imagem NDVI com grade de coordenadas

cax = ax.imshow(ndvi, cmap='RdYlGn')

plt.colorbar(cax, label='NDVI')

# Adicionar título

ax.set_title('NDVI')

# Adicionar legenda

plt.annotate('Valores de NDVI:\n-1: Solo\n0: Área
mista\n1: Vegetação densa', xy=(0.02, 0.98),
xycoords='axes fraction', ha='left', va='top')

# Exibir a imagem

plt.show()
```

Fonte: Autora (2023).

No geral, o código cria uma visualização informativa que permite avaliar os dados da área específica, ajudando a identificar padrões.

4.2.8 Exportação dos dados

O objetivo em questão é realizar a exportação dos dados da coleção de Imagens do Landsat-8, após os cálculos desenvolvidos nas imagens, dessa forma os dados são armazenados no Google Drive como arquivos .tiff, o que permite que o usuário possa também acessar os dados em ambiente local.

Quadro 14 - Exportação dos dados para o Google Drive

```
# Exportando a coleção de imagens para o Google Drive
output_folder = '/content/drive/MyDrive/SAVI/'

# Função para exportar cada imagem da coleção
def export_image(image):
    date =
ee.Date(image.get('system:time_start')).format('YYYY-MM-
dd').getInfo()
    filename = output_folder + 'SAVI_' + date + '.tif'

# Configuração dos parâmetros de exportação
export_params = {
    'image': image.select(['SAVI']),
    'description': 'SAVI_export',
    'folder': output_folder,
    'fileNamePrefix': 'SAVI_' + date,
    'scale': 30,
    'region': region.geometry().bounds(),
    'fileFormat': 'GeoTIFF',
    'formatOptions': {
        'cloudOptimized': True
    }
}

# Executando a exportação
task = ee.batch.Export.image.toDrive(**export_params)
task.start()

# Aplicando a função de exportação para cada imagem na
coleção
l8_list = l8.toList(l8.size())
for i in range(l8.size().getInfo()):
    export_image(ee.Image(l8_list.get(i)))

# Monitorando o status das tarefas de exportação
import time
```

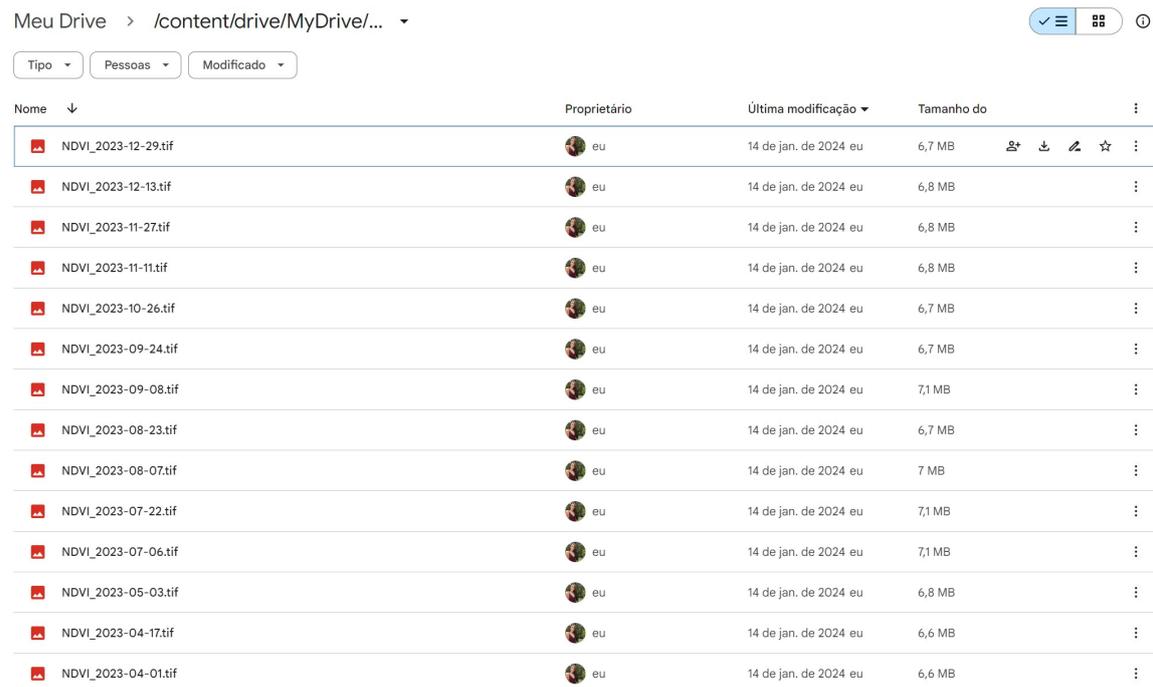
```
while any(task.active() for task in ee.batch.Task.list()):
    time.sleep(30) # Aguarda 30 segundos

print("Exportação concluída!")
```

Fonte: Autora (2023).

A Quadro 14 apresenta a rotina desenvolvida, nela foi possível automatizar o processo de exportação, onde os dados exportados serão armazenados no Google Drive, Figura 8, que recebe a imagem como entrada e salva a mesma a partir da data da imagem, ou seja, extraíndo a data que está contida no metadado da imagem e por fim quando todas a imagens são exportadas recebemos a informação que toda a exportação está concluída.

Figura 8 - Dados exportados



The screenshot shows a Google Drive interface with a list of files. The files are named with a date and a .tif extension, representing NDVI data. The columns include file name, owner, last modification date, and file size. The files are sorted by name in descending order.

Nome	Proprietário	Última modificação	Tamanho do
NDVI_2023-12-29.tif	eu	14 de jan. de 2024 eu	6,7 MB
NDVI_2023-12-13.tif	eu	14 de jan. de 2024 eu	6,8 MB
NDVI_2023-11-27.tif	eu	14 de jan. de 2024 eu	6,8 MB
NDVI_2023-11-11.tif	eu	14 de jan. de 2024 eu	6,8 MB
NDVI_2023-10-26.tif	eu	14 de jan. de 2024 eu	6,7 MB
NDVI_2023-09-24.tif	eu	14 de jan. de 2024 eu	6,7 MB
NDVI_2023-09-08.tif	eu	14 de jan. de 2024 eu	7,1 MB
NDVI_2023-08-23.tif	eu	14 de jan. de 2024 eu	6,7 MB
NDVI_2023-08-07.tif	eu	14 de jan. de 2024 eu	7 MB
NDVI_2023-07-22.tif	eu	14 de jan. de 2024 eu	7,1 MB
NDVI_2023-07-06.tif	eu	14 de jan. de 2024 eu	7,1 MB
NDVI_2023-05-03.tif	eu	14 de jan. de 2024 eu	6,8 MB
NDVI_2023-04-17.tif	eu	14 de jan. de 2024 eu	6,6 MB
NDVI_2023-04-01.tif	eu	14 de jan. de 2024 eu	6,6 MB

Fonte: Autora (2023).

4.2.9 Tempo de execução

Para determinar o tempo de execução, foi desenvolvido um código que registra o momento inicial da execução do programa. Quando o usuário inicia a execução do programa, esse instante é marcado e armazenado na variável *start_time*. Esse registro permite medir o intervalo de tempo gasto durante a

execução do programa até a obtenção dos resultados, o que proporciona uma avaliação precisa do desempenho do programa em termos de eficiência temporal.

Quadro 15 - Código que retorna o tempo de execução total do programa

```
# Registrar o tempo inicial
start_time = time.time()

# Registrar o tempo final
end_time = time.time()

# Calcular o tempo total de execução
total_time = end_time - start_time

print(f"tempo total de execução: {total_time: .2f}
segundos")
```

Fonte: Autora (2023).

Todavia, é importante destacar que o tempo de execução pode variar conforme a quantidade de dados a serem processados e a complexidade das operações realizadas.

5 RESULTADOS

Aqui será apresentado as rotinas desenvolvidas no trabalho, suas etapas de implementação juntamente com os resultados obtidos.

5.1 Resultados obtidos a partir da programação

Com base nos dados calculados, foram realizadas análises estatísticas dos índices calculados para o ano de 2023. Durante esse processo, sete datas foram removidas com base no Intervalo Interquartil (IQR), a fim de identificar os dados que diferem dos dados esperados. A tabela resultante, apresentadas na Tabelas 12, 13, 14, e 15, foi ordenada e selecionada para exibir as propriedades relevantes, oferecendo uma visão resumida e organizada dos dados da série temporal.

Tabela 2 - Resultado obtido da análise de mínimo, máximo e média para cada data analisada NDVI

month	date	year	NDVI _{mean}	NDVI _{min}	NDVI _{max}	NM_MUN
1	2023-01-11	2023	0.245093	-0.146742	0.792030	Petrolândia
2	2023-02-12	2023	0.205409	-0.249462	0.757093	Petrolândia
3	2023-03-16	2023	0.205715	-0.057112	0.591930	Petrolândia
4	2023-04-17	2023	0.228190	-0.036651	0.515330	Petrolândia
5	2023-05-03	2023	0.151023	-0.156429	0.952643	Petrolândia
7	2023-07-22	2023	0.125899	-0.156571	0.999449	Petrolândia
8	2023-08-23	2023	0.134381	-0.194487	0.854155	Petrolândia
9	2023-09-24	2023	0.122897	-0.163465	0.550664	Petrolândia
10	2023-10-26	2023	0.120585	-0.173089	0.697509	Petrolândia
11	2023-11-27	2023	0.140217	-0.157052	0.553719	Petrolândia
12	2023-12-29	2023	0.171901	-0.150510	0.571697	Petrolândia

Fonte: Autora (2023).

Tabela 3 – Resultado da análise de mínimo, máximo e média para cada data analisada SAVI

month	date	year	SAVI _{mean}	SAVI _{min}	SAVI _{max}	NM_MUN
1	2023-01-11	2023	0.245093	-0.146742	0.792030	Petrolândia
1	2023-01-11	2023	0.367633	-0.220107	1.188019	Petrolândia
2	2023-02-12	2023	0.308107	-0.374188	1.135498	Petrolândia
3	2023-03-16	2023	0.308567	-0.085666	0.887875	Petrolândia
4	2023-04-17	2023	0.342279	-0.054975	0.772984	Petrolândia
5	2023-05-03	2023	0.226530	-0.234636	1.428768	Petrolândia
7	2023-07-22	2023	0.188845	-0.234850	1.498966	Petrolândia
8	2023-08-23	2023	0.201568	-0.291723	1.281057	Petrolândia
9	2023-09-24	2023	0.184343	-0.245192	0.825984	Petrolândia
10	2023-10-26	2023	0.180874	-0.259627	1.046244	Petrolândia
11	2023-11-27	2023	0.210321	-0.235571	0.830567	Petrolândia
12	2023-12-29	2023	0.257847	-0.225760	0.857534	Petrolândia

Fonte: Autora (2023).

Tabela 4 - Resultado da análise de mínimo, máximo e média para cada data analisada NDWI

month	date	year	NDWI _{mean}	NDWI _{min}	NDWI _{max}	NM_MUN
1	2023-01-11	2023	0.245093	-0.146742	0.792030	Petrolândia
1	2023-01-11	2023	-0.251669	-0.996585	0.145724	Petrolândia
2	2023-02-12	2023	-0.225529	-0.998212	0.332394	Petrolândia
3	2023-03-16	2023	-0.217101	-0.699885	0.048979	Petrolândia
4	2023-04-17	2023	-0.232941	-0.474273	0.024983	Petrolândia
5	2023-05-03	2023	-0.152989	-0.994254	0.115926	Petrolândia
7	2023-07-22	2023	-0.141961	-0.999145	0.174901	Petrolândia
8	2023-08-23	2023	-0.171683	-0.973616	0.174612	Petrolândia
9	2023-09-24	2023	-0.166444	-0.509234	0.145165	Petrolândia
10	2023-10-26	2023	-0.172579	-0.597419	0.129950	Petrolândia
11	2023-11-27	2023	-0.186156	-0.522388	0.173292	Petrolândia

12	2023-12-29	2023	-0.206020	-0.978621	0.163369	Petrolândia
-----------	------------	------	-----------	-----------	----------	-------------

Fonte: Autora (2023).

Tabela 5 - Resultado da análise de mínimo, máximo e média para cada data analisada NDBI

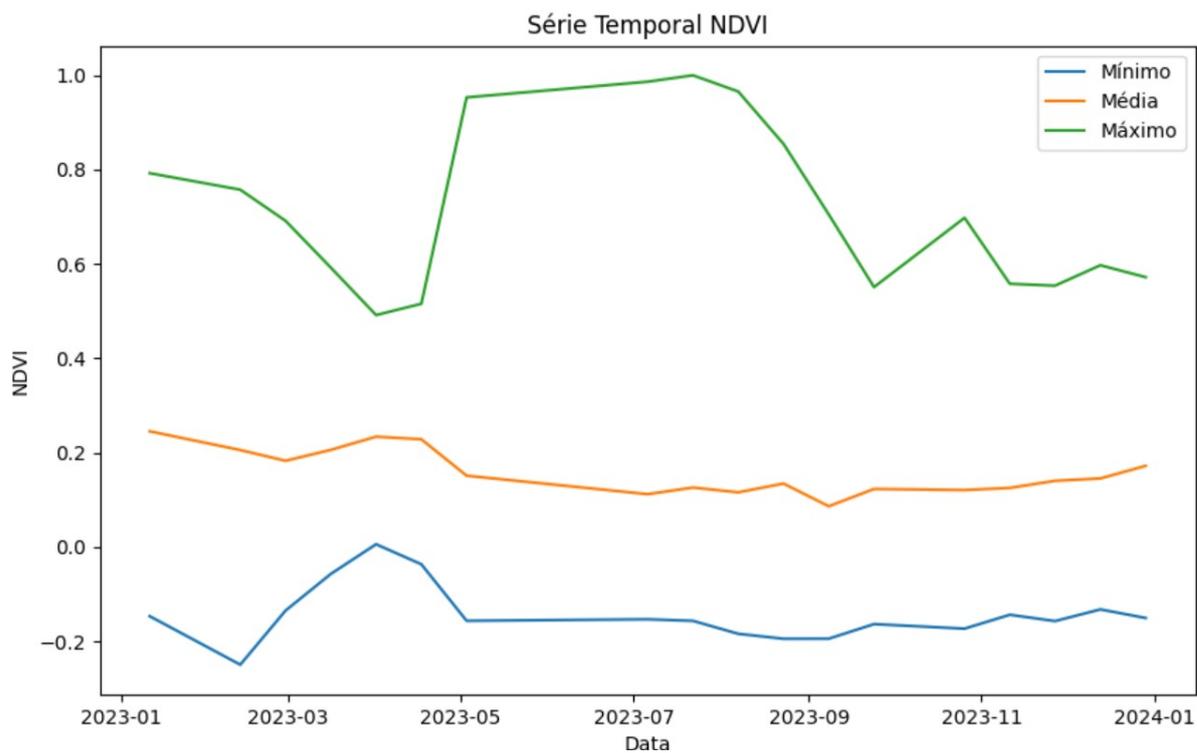
month	date	year	NDBImean	NDBImin	NDBImax	NM_MUN
1	2023-01-11	2023	0.245093	-0.146742	0.792030	Petrolândia
1	2023-01-11	2023	0.038746	-0.312340	0.386315	Petrolândia
2	2023-02-12	2023	0.000390	-0.348176	0.403713	Petrolândia
3	2023-03-16	2023	0.040452	-0.189879	0.320969	Petrolândia
4	2023-04-17	2023	0.099532	-0.103563	0.350204	Petrolândia
5	2023-05-03	2023	0.132204	-0.373492	0.367388	Petrolândia
7	2023-07-22	2023	0.010252	-0.444829	0.473567	Petrolândia
8	2023-08-23	2023	-0.083581	-0.359985	0.365663	Petrolândia
9	2023-09-24	2023	-0.103776	-0.240469	0.352443	Petrolândia
10	2023-10-26	2023	-0.104121	-0.379078	0.361502	Petrolândia
11	2023-11-27	2023	-0.085918	-0.280971	0.364631	Petrolândia
12	2023-12-29	2023	-0.050583	-0.402307	0.384827	Petrolândia

Fonte: Autora, 2023.

A organização dos dados obtidos desta maneira é de extrema importância para o monitoramento, pois permite que a observação das tendências ao longo dos períodos específicos sejam analisados e assim permite a contextualização das variações ao longo do tempo.

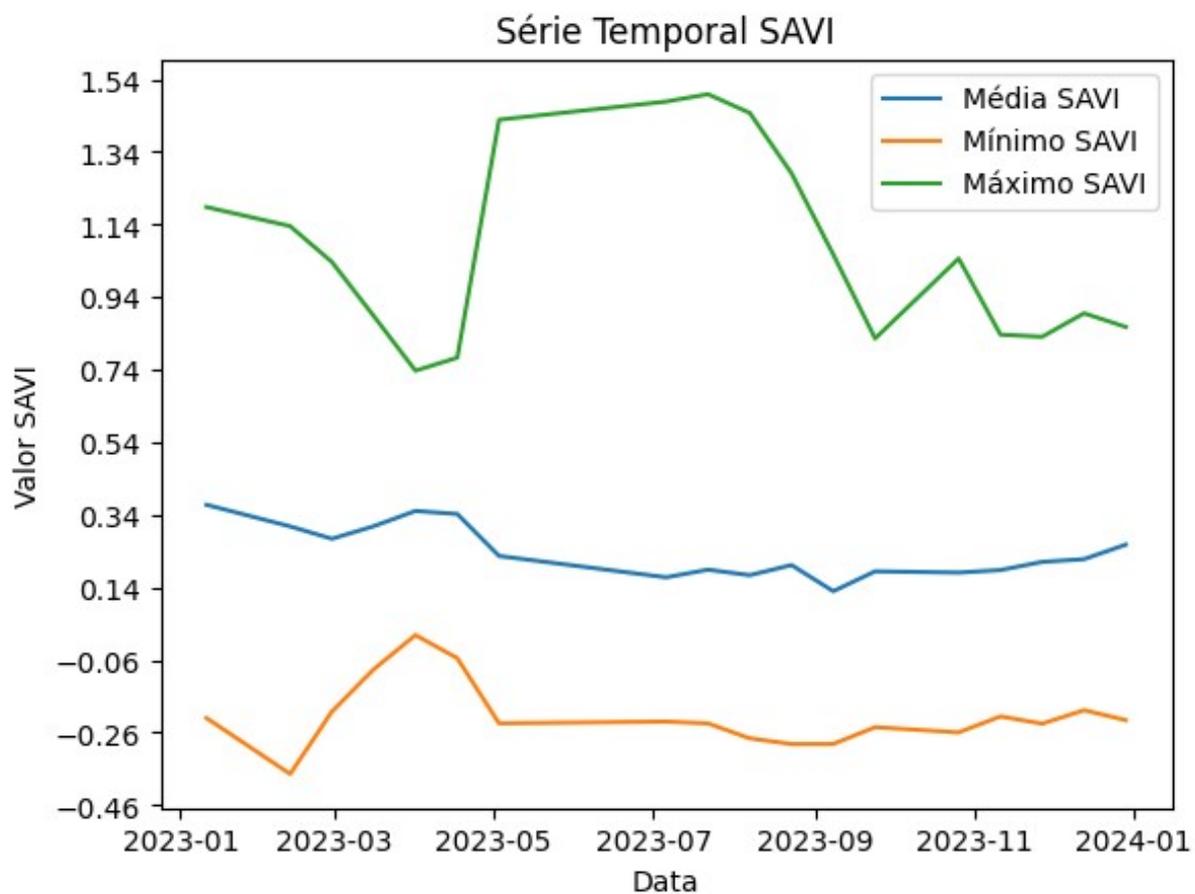
Com base nisso, apresentamos os resultados da análise temporal de 2023 para os índices NDVI, SAVI, NDWI e NDBI, conforme ilustrados nas Figuras, 9, 10, 11 e 12, respectivamente. É interessante observar que os gráficos do NDVI e SAVI apresentam semelhanças, visto que o SAVI é uma variação do NDVI que leva em consideração a correção do solo.

Figura 9 - Gráfico temporal NDVI



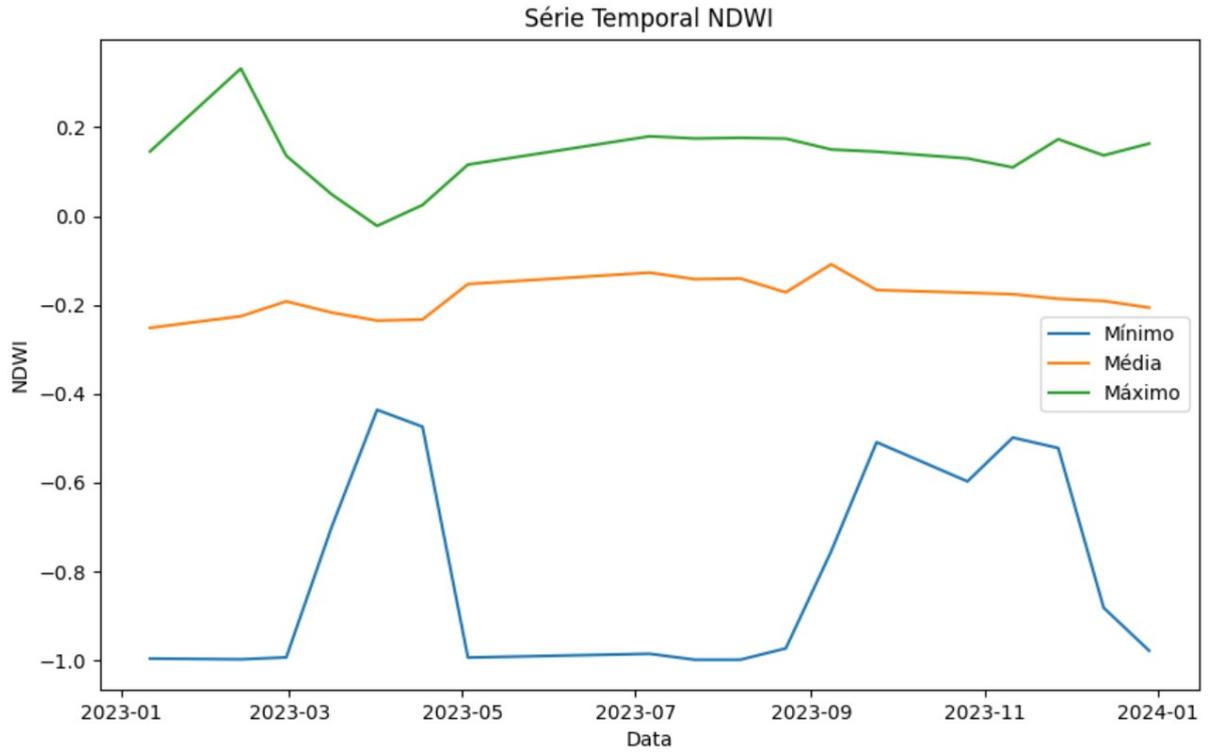
Fonte: Autora (2023).

Figura 10 - Gráfico temporal SAVI



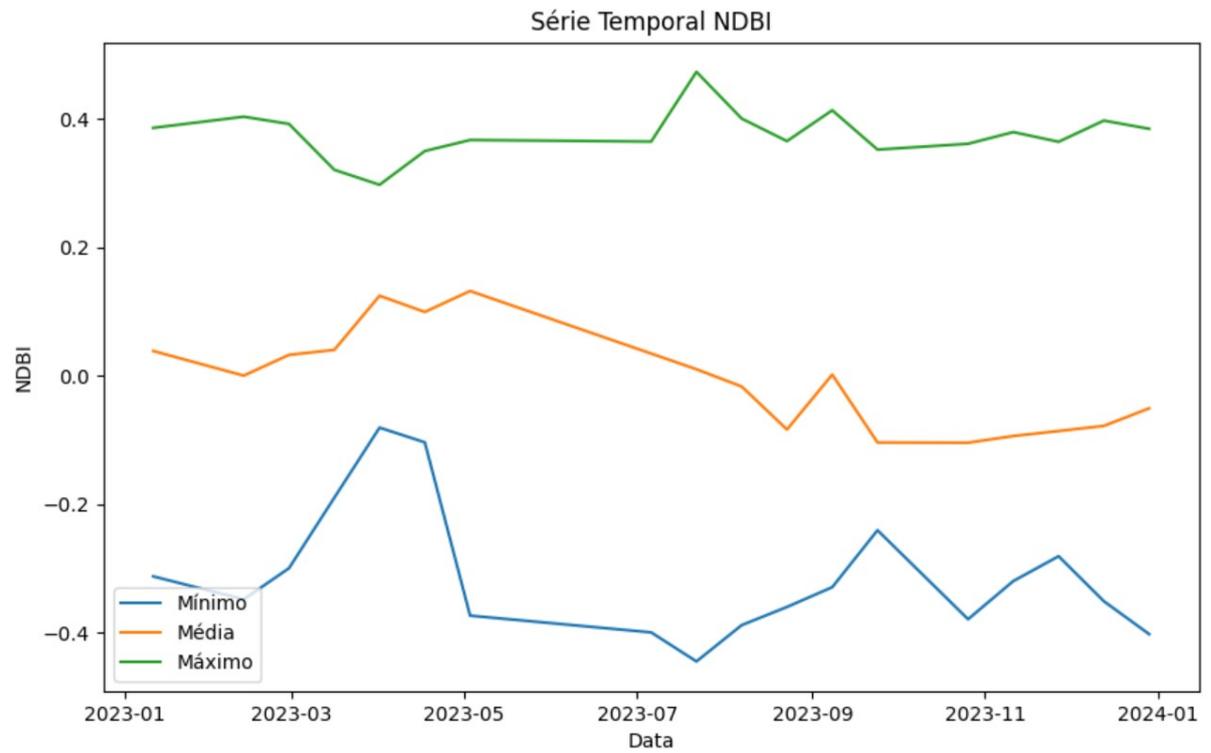
Fonte: Autora (2023).

Figura 11 - Gráfico temporal NDWI



Fonte: Autora (2023).

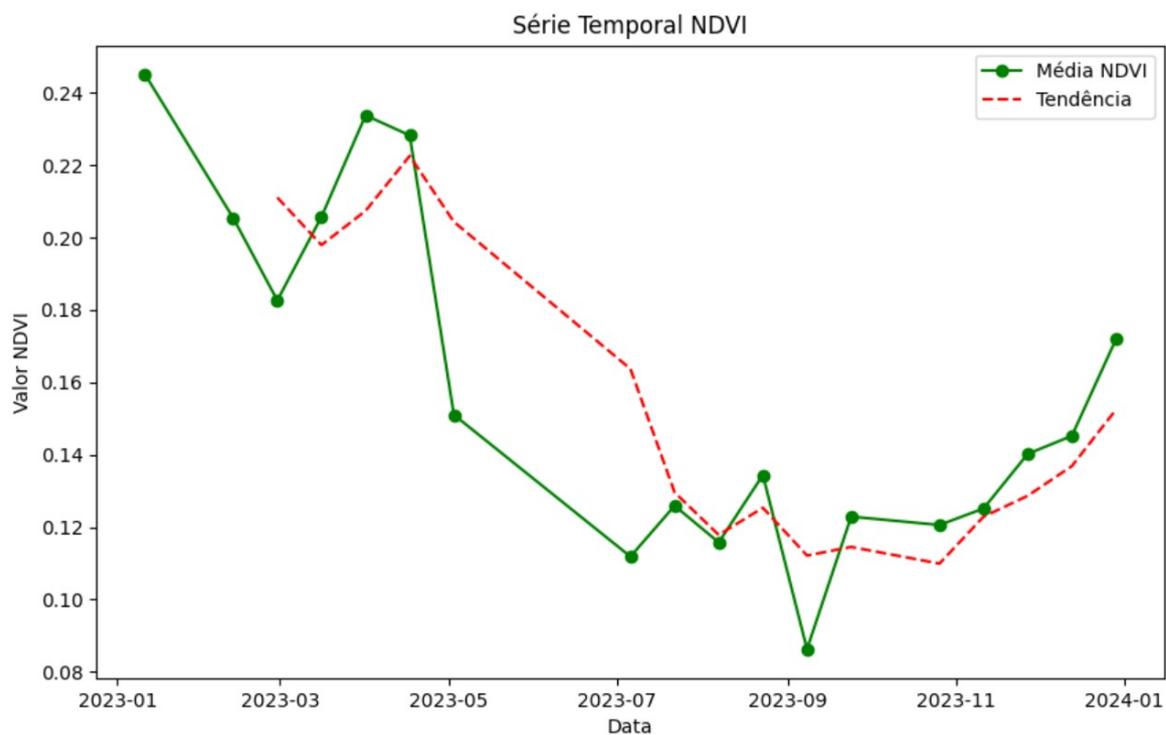
Figura 12 - Gráfico temporal NDBI



Fonte: Autora (2023).

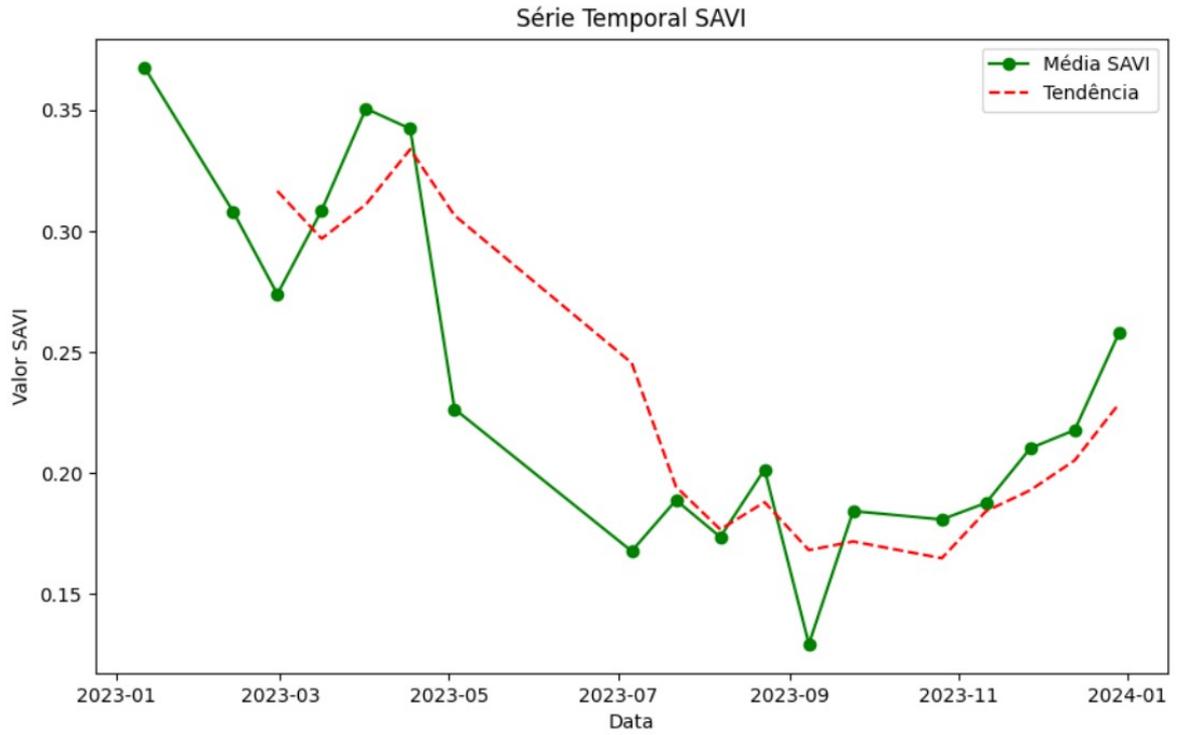
Abaixo, apresentamos os resultados da análise, nos quais procedemos a suavização dos dados a partir de uma linha de tendência. Nela observamos que o gráfico possui uma tendência subjacente dos dados ao longo do tempo, nela podemos observar as variações e projetar o comportamento futuro das variações. Figura 13, 14, 15 e 16.

Figura 13 - Gráfico temporal com linha de tendência NDVI



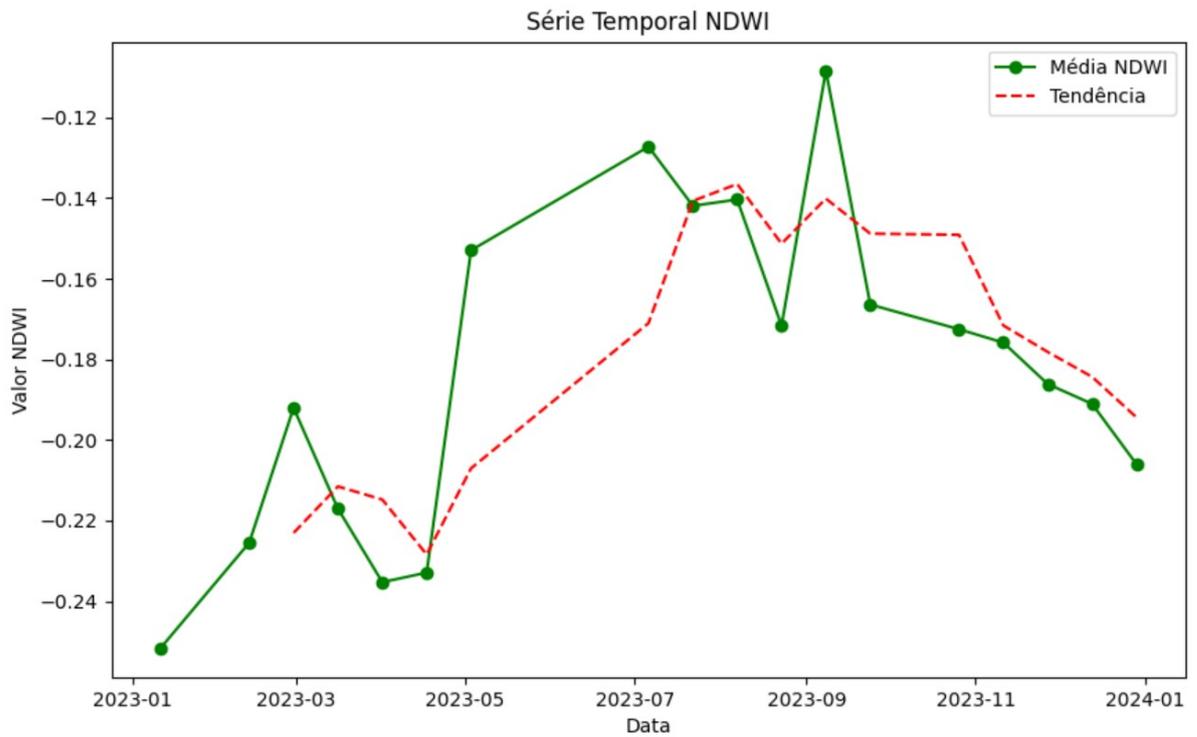
Fonte: Autora (2023).

Figura 14 - Gráfico temporal com linha de tendência SAVI



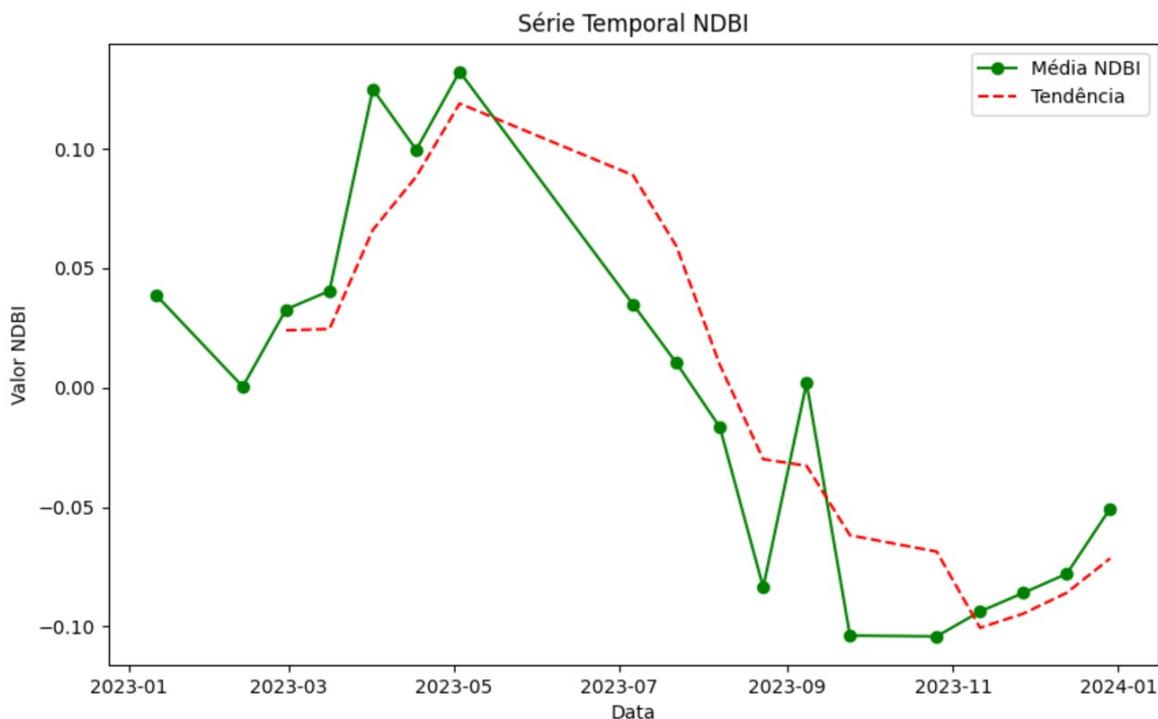
Fonte: Autora (2023).

Figura 15 - Gráfico temporal com linha de tendência NDWI



Fonte: Autora (2023).

Figura 16 - Gráfico temporal com linha de tendência NDBI

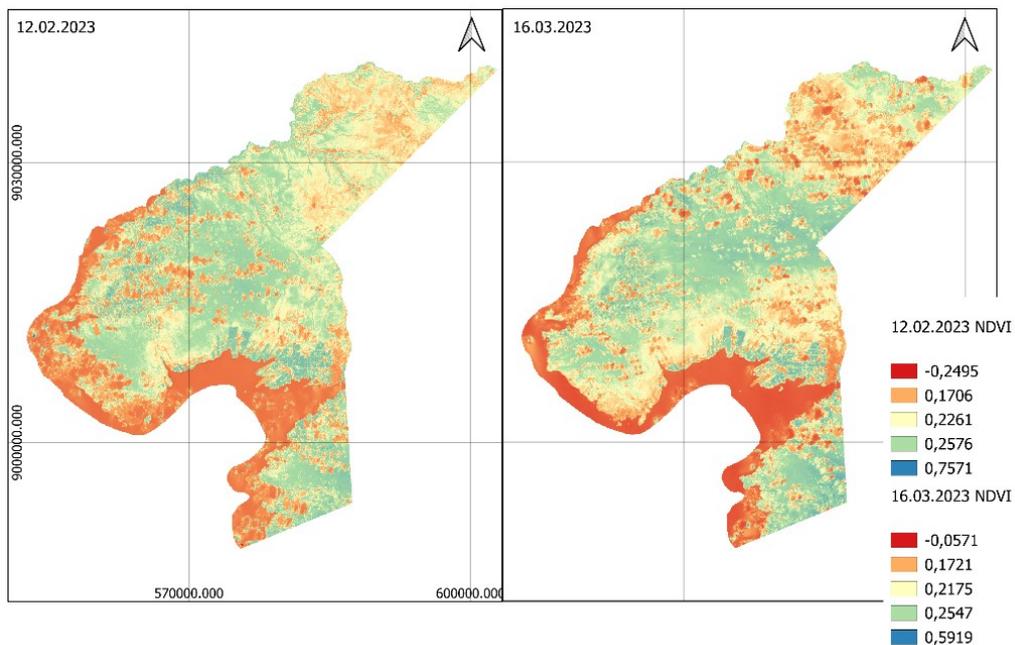


Fonte: Autora (2023).

A seguir são apresentados os resultados obtidos em forma de mapa dos índices calculados, conforme visualizado nas Figuras 17, 18, 19 e 20. Estas figuras exibem mapas representativos dos índices em períodos de chuva e seca. O propósito é observar de forma ilustrativa as variações que ocorrem nesses períodos.

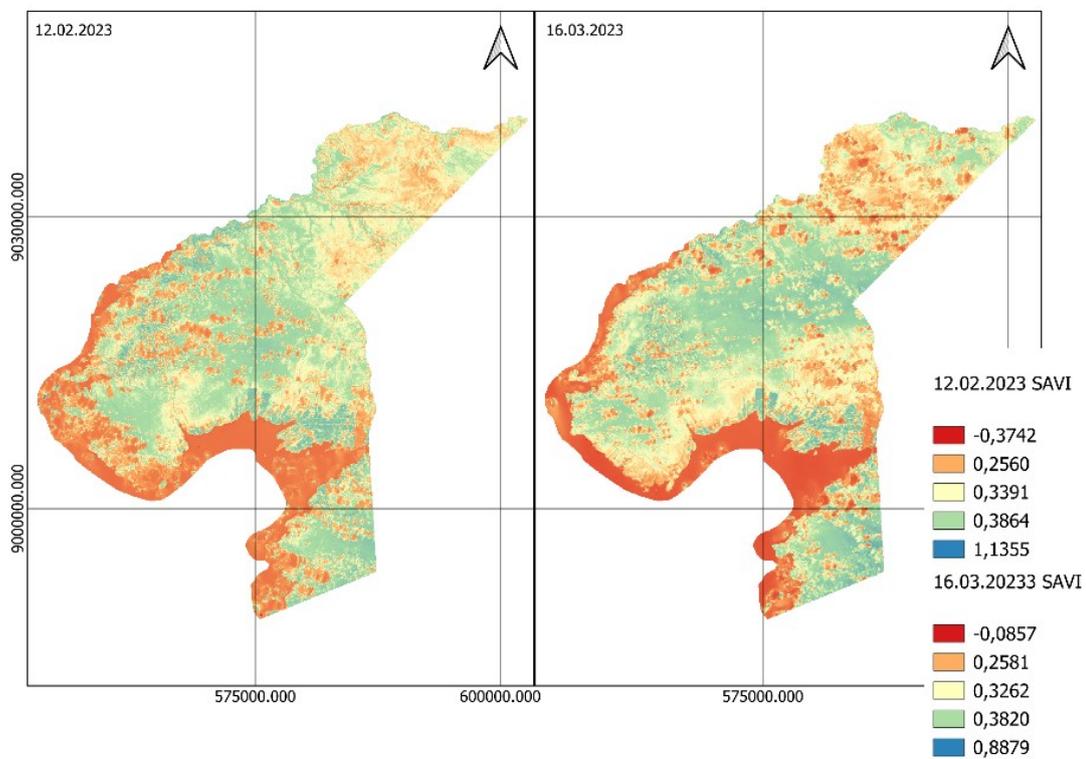
Ao observarmos os mapas gerados durante os dois períodos, podemos perceber os diferentes níveis de vigor vegetativo e umidade do solo, como também podemos observar a discrepância entre os resultados encontrados do NDBI no período chuvoso e de seca. Portanto, podemos considerar que os cálculos e a geração dos mapas a partir da programação são condizentes com o esperado.

Figura 17 - NDVI (Período de chuva)



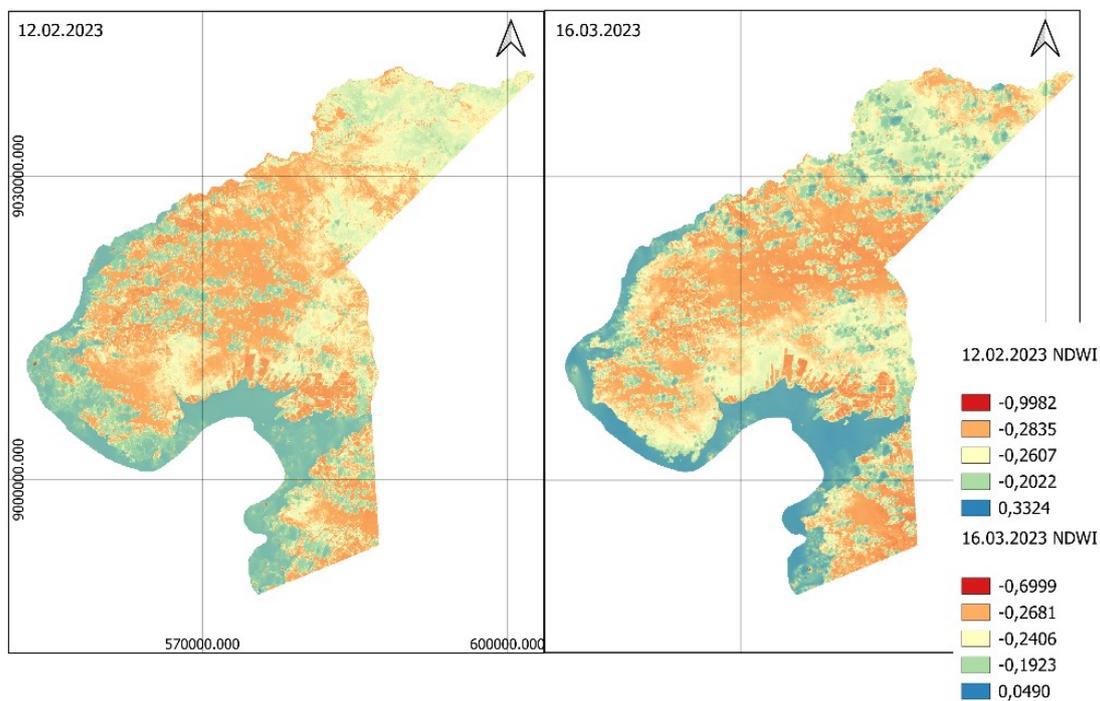
Fonte: Autora (2023).

Figura 18 - SAVI (Período de chuva)



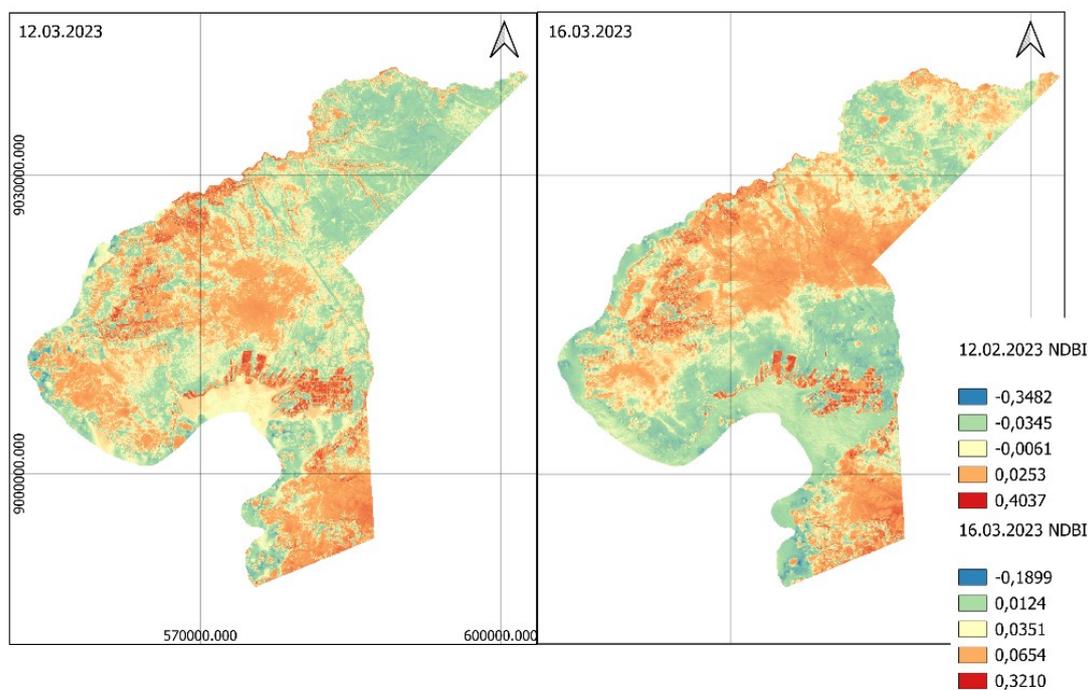
Fonte: Autora (2023).

Figura 19 - NDWI (Período de chuva)



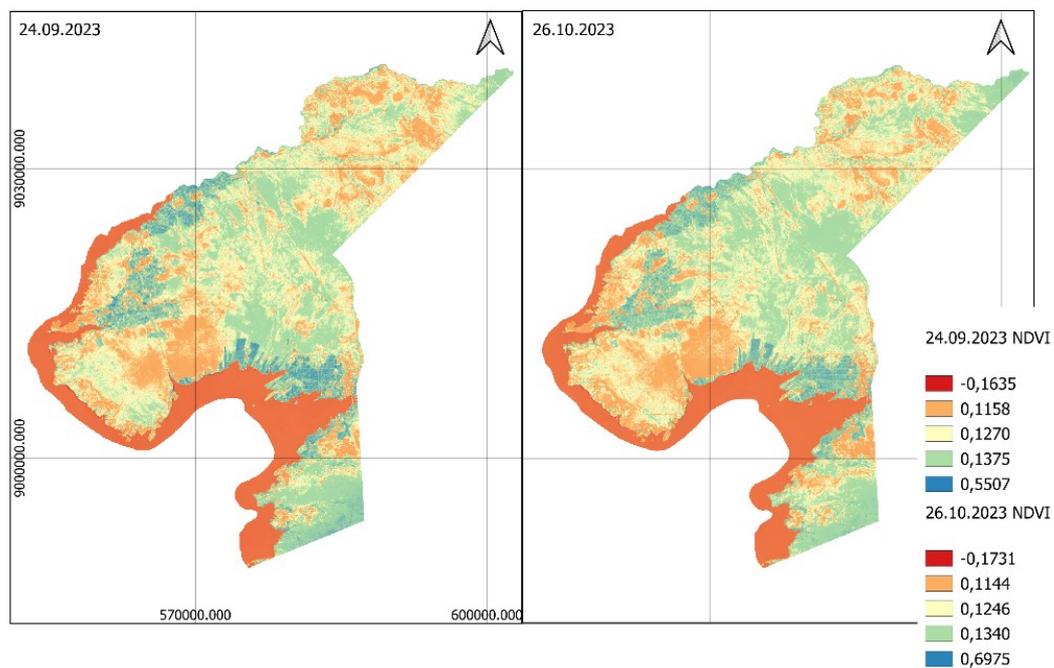
Fonte: Autora (2023).

Figura 20 - NDBI (Período de chuva)



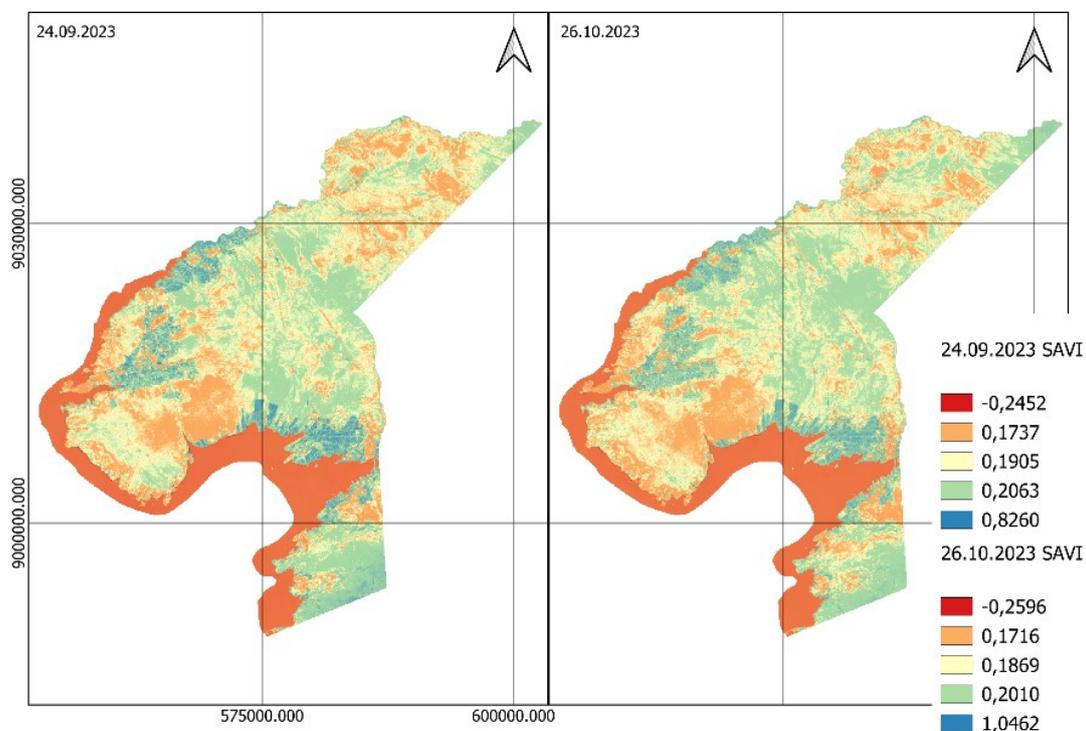
Fonte: Autora (2023).

Figura 21 - NDVI (Período de seca)



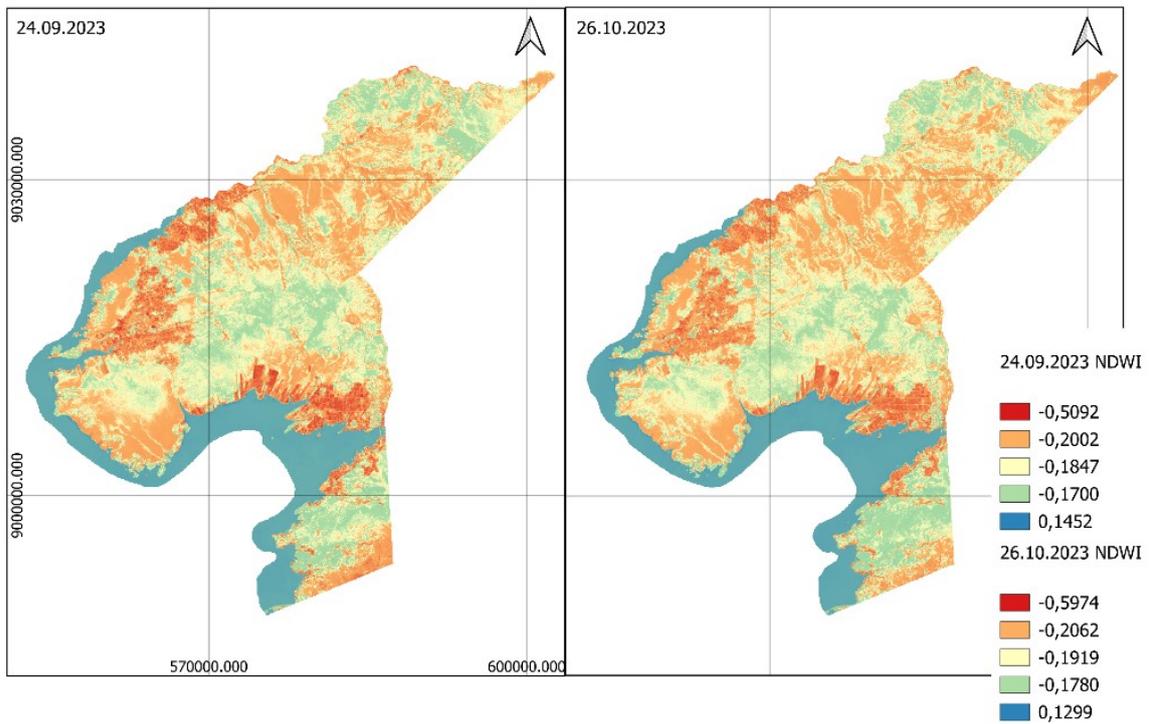
Fonte: Autora (2023).

Figura 22 - SAVI (Período de seca)



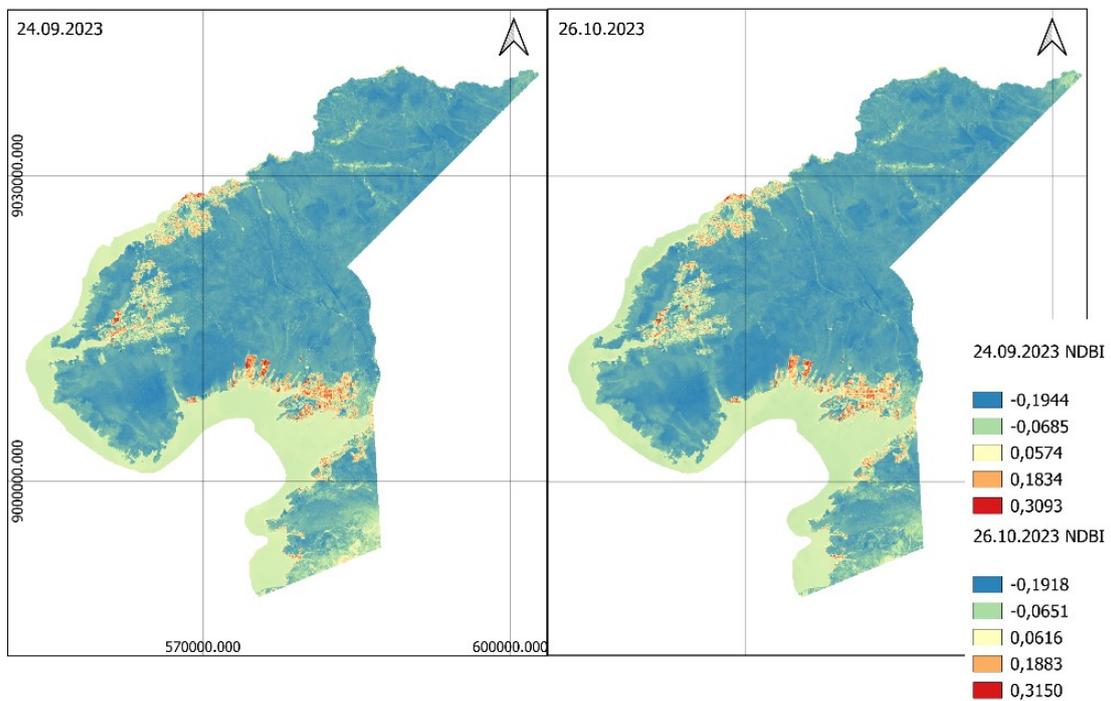
Fonte: Autora (2023).

Figura 23 - NDWI (Período de seca)



Fonte: Autora (2023).

Figura 24 - NDBI (Período de seca)



Fonte: Autora (2023).

Por fim, a partir dos resultados obtidos pela programação o tempo de execução do processo é significativamente reduzido, onde permitiu que a atividade fosse concluída em uma fração de tempo muito menor caso fosse feita de forma manual. Para o presente trabalho, o tempo de processamento foi de 00:07:35, levando em consideração todo o processamento e exportação dos dados para o drive.

CONSIDERAÇÕES FINAIS

O trabalho apresentou uma metodologia para gerar processamentos de forma automatizada a partir da criação de rotinas, a partir do que foi apresentado, é possível visualizar o potencial na criação de rotinas para a análise espacial. Neste trabalho, foram apresentados uma sequência de códigos que tem como finalidade de criação de rotina para quatro índices, onde a aplicação do estudo foi para o sensor OLI.

O trabalho lidou com uma sequência lógica para que o processamento e a funcionalidade do código fossem atingidos, podemos citar que os parâmetros iniciais foram definidos como:

- Definição da área de estudo;
- Definição das datas de interesse;
- Definição do sensor.

O programa trabalha de forma versátil, permitindo ao usuário escolher o período de análise, onde o tempo de execução é determinado pelo número de imagens e pixels envolvidos. A independência do poder de processamento da máquina é uma vantagem significativa, e o tempo de execução é calculado por meio de um código que registra o momento inicial da execução e mede o intervalo até a conclusão do programa. Em resumo, o programa demonstrou eficiência no tempo de execução, beneficiado pelas otimizações do ambiente de execução.

Podemos abordar os seguintes pontos positivos observados ao longo da pesquisa:

i. O tempo de execução não estar vinculado ao poder de processamentos da máquina que é utilizada pelo usuário, significando dizer que, o programa terá uma performance similar independente da capacidade do computador utilizado, o que facilita sua aplicação.

ii. Eficiência na manipulação dos dados, visto que, além do processo ser automatizado e não necessitar da capacidade do computador, o trabalho é desenvolvido em nuvem, o que permite que até os dados gerados sejam salvos na nuvem.

iii. Facilidade de acesso aos dados de satélite.

iv. Colaboração que esse ambiente permite que seja realizada entre os pesquisadores.

v. O tempo de execução do programa depende diretamente da quantidade de imagens selecionadas e do número de pixels contidos em cada imagem. Quanto mais imagens e pixels existirem, mais operações serão realizadas para obter os resultados desejados, especialmente durante as etapas iniciais de leitura de cada imagem e na geração de análises estatísticas.

Por outro lado, essa abordagem também apresenta algumas desvantagens a serem consideradas.

i. Eventos climáticos adversos, como cobertura persistente de nuvens, podem afetar a qualidade dos dados utilizados, impactando a precisão dos resultados, onde se faz necessário analisar os dados de forma a identificar quais dados podem ser utilizados.

ii. O item 5.1.7 destaca os resultados no período chuvoso e de seca, onde é possível perceber que os dados dos meses de fevereiro e março contêm uma cobertura persistente de nuvens. Embora o algoritmo de processamento aplique correções nas imagens com nuvens, é essencial considerar que nuvens podem gerar sombras, afetando a reflectância dos alvos e alterando os números digitais nas imagens. Essas alterações podem confundir o algoritmo e levar a resultados imprecisos.

iii. É crucial considerar que diferentes tipos de imagens podem apresentar comportamentos distintos ao serem processadas com o algoritmo, portanto, existe a necessidade de realizar experimentos para que seja possível avaliar o potencial do código em outras aplicações.

iv. A qualidade, resolução e características específicas das imagens podem influenciar os resultados e a interpretação dos índices. Portanto, é recomendado realizar testes e análises comparativas utilizando diferentes fontes de imagens para entender a sensibilidade do algoritmo às variações dos dados.

v. Em suma, é importante considerar datas com menor ocorrência de nuvens como estratégia para garantir a qualidade dos dados, como também outras técnicas de processamento, como fusão de imagens em diferentes datas ou métodos de correção atmosférica mais robustos.

Em resumo podemos observar que esse ambiente de trabalho possibilita o aprimoramento contínuo do trabalho, possibilitando a integração com outros pesquisadores onde podemos compartilhar melhorias e insights valiosos para futuros desenvolvimentos na área.

Os resultados obtidos nesse trabalho mostram-se úteis, principalmente levando em consideração o objetivo da criação de rotinas. Podemos citar como contribuições por este trabalho:

- Apresentação de novas plataformas gratuitas GEE e Google Collaboratory que auxiliam a criação de rotinas em nuvem;
- Facilidade para o pré-processamento de um grande volume de dados, a partir de rotinas de programação.
- Aplicação de rotinas para as imagens do Landsat-8.

Todo o desenvolvimento criado neste trabalho apresentou aplicações práticas de modo a auxiliar futuros pesquisadores de diferentes áreas. Os métodos aqui apresentados e discutido, formam uma base para a implementação de novos projetos. Uma gama de melhorias e ajustes podem ser feitos buscando um resultado mais satisfatório. O projeto é compartilhável em nuvem e ficará disponível para futuros aprimoramentos.

REFERÊNCIAS

Almeida, K. C. de; Costa, A. P. F. da; Rocha, T. B. da. Utilização do Google Earth Engine para análise comparativa da linha de costa de dois deltas da costa leste brasileira entre 1986 e 2021: Delta do rio São Francisco (AL/SE) e delta do rio Paraíba Do Sul (RJ). *Geo UERJ*, [S. l.], n. 42, 2023. DOI: 10.12957/geouerj.2023.74650. Disponível em: <https://www.e-publicacoes.uerj.br/geouerj/article/view/74650>. Acesso em: 16 jun. 2023.

Bisong, E. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA: Apress, 2019.

Cohen, W. B.; Maier-Sperger, T. K.; Gower S. T.; Turner, D. P. An improved strategy for regression of biophysical variables and Landsat ETM+ data. *Remote Sensing of Environment*, v.84, p.561-571, 2003.

Conheça o COLAB. Google, 2022. Disponível em: <https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR>. Acesso em: 12 de fevereiro de 2022.

Crosta, A. P. *Processamento digital de imagens de sensoriamento remoto*. Campinas, SP: IG/UNICAMP, 1992.

Clewley, Daniel et al. A python-based open source system for geographic object-based image analysis (GEOBIA) utilizing raster attribute tables. *Remote Sensing*, v. 6, n. 7, p. 6111-6135, 2014.

Da Silva, M. D. Aplicação da Ferramenta Google Collaboratory para o Ensino da Linguagem Python. In: *Anais da IV Escola Regional de Engenharia de Software*. SBC, 2020. p.67-76.

DPI/INPE. *Sensoriamento Remoto*. [2006?]. Disponível em: http://www.dpi.inpe.br/spring/portugues/tutorial/introducao_sen.html. Acesso em: 22 jun. 2023.

Embrapa Territorial. *Satélites de Monitoramento*. Campinas, 2018. Disponível em: <https://www.embrapa.br/satelites-de-monitoramento>. Acesso em: 16 jan. 2023.

Freitas, C. A. S. de et al. Remote Sensing applied to the analysis of the Heat Islands phenomenon in the city of Juazeiro do Norte - CE. *Research, Society and Development*, [S. l.], v. 11, n. 12, p. e237111234265, 2022. DOI: 10.33448/rsd-v11i12.34265. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/34265>. Acesso em: 19 nov. 2022.

Fritz, S. et al. A comparison of global agricultural monitoring systems and current gaps. *Agricultural Systems*, v. 168, n. December 2017, p. 258–272, 2019.

Gómez-Dans, J. L., et al. (2016). Google Earth Engine, a new cloud-computing platform for global-scale environmental data analyses. *Remote Sensing of Environment*, 202, 18–27.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18-27.

Han, W. et al. Spatiotemporal Variation of NDVI in Anhui Province from 2001 to 2019 and Its Response to Climatic Factors. *Forests* 2022, 13, 1643. <https://doi.org/10.3390/f13101643>

Huete, A.R. (1988) A Soil Adjusted Vegetation Index (SAVI). *Remote Sensing of Environment*, 25, 295-309. [http://dx.doi.org/10.1016/0034-4257\(88\)90106-X](http://dx.doi.org/10.1016/0034-4257(88)90106-X)

Jackson, T. J. et al. Vegetation water content mapping using Landsat data derived normalized difference water index for corn and soybeans. *Remote Sensing of Environment*, v. 92, n. 4, p. 475-482, 2004.

Johary, R. et al. Detection of Large-Scale Floods Using Google Earth Engine and Google Colab. *Remote Sens.* 2023, 15, 5368. <https://doi.org/10.3390/rs15225368>

Kinakh, V. et al. Geoinformation technology of analysis and visualization of spatial data on greenhouse gas emissions using Google Earth Engine. *Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017*, v. 1, p. 212–215, 2017.

Landsat - Land Remote Sensing Satellite. Embrapa, 2022. Disponível em: <https://www.embrapa.br/satelites-de-monitoramento/missoes/landsat>. Acesso em: 12 de fevereiro de 2023.

Landsat. Inpe, 2022. Disponível em: <http://www.dgi.inpe.br/documentacao/satelites/landsat>. Acesso em: 10 de fevereiro de 2022.

Lemenkova, P.; Debeir, O. Satellite Image Processing by Python and R Using Landsat 9 OLI/TIRS and SRTM DEM Data on Côte d'Ivoire, West Africa. *J. Imaging* 2022, 8, 317. <https://doi.org/10.3390/jimaging8120317>

Medeiros, C. P. “Processamento de Imagem e Aprendizagem de Máquina para Classificação de Anomalias em Vias Públicas”, Universidade Federal de Pernambuco, Centro de Informática, 2018.

Menezes, P. R.; Almeida, T. DE. Introdução ao processamento de imagem de sensoriamento remoto. 1a. ed. Brasília: UnB, 2012.

Menezes, N. N. Coutinho. Introdução à programação com Python. São Paulo: Novatec, 2010.

Millman, K. J., & Aivazis, M. (2011). Python for Scientists and Engineers. *Computing in Science & Engineering*, 13(2), 9–12.

McFeeters, S.K. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing*, v.17, n.7, p.1425-1432, 1996.

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference.

Novo, E.M.L.M., 2010. *Sensoriamento remoto: Princípios e aplicações*. Edgard Blücher, São Paulo, SP. 388 p.

Puget, J. F. "The most popular language for machine learning is" Acesso em: *The Most Popular Language for Machine Learning Is ... (https_medium.com)*, 2021.

Rodríguez-Veiga, P. et al. Assessment of Sentinel-2 for mapping boreal forest cover and change. *Remote Sensing of Environment*, 196, 126-137, 2017.
Schowengerdt, R. A. *Techniques for image processing and classifications in remote sensing*. Academic, New York, 1983.

Tamiminia, H. et al. ISPRS Journal of Photogrammetry and Remote Sensing Google Earth Engine for geo-big data applications : A meta-analysis and systematic review. *ISPRS Journal of Photogrammetry and Remote Sensing*, v. 164, n. January, p. 152–170, 2020.

Yang, C. et al. Utilizing Cloud Computing to address big geospatial data challenges. *Computers, Environment and Urban Systems*, v. 61, p. 120–128, 2017.

Zha, Y.; Gao, J.; Ni, S. Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. *International Journal of Remote Sensing*. v. 24, n. 3, p. 583-659, 2003.

APÊNDICE A – Google Collaboratory e o Google Earth Engine (GEE) para a análise multitemporal das mudanças de temperatura ocasionadas por ilhas de calor a partir do TerraClimate

¹Universidade Federal de Pernambuco, Avenida Professor Moraes Rego, 1235 – Cidade Universitária, Recife – PE, 50670-901, sabrina.gomes@ufpe.br; Universidade Federal de Pernambuco, Avenida Professor Moraes Rego, 1235 – Cidade Universitária, Recife – PE, 50670-901, ana.candeias@ufpe.br

Resumo

A cartografia é uma ciência que possibilita estudos em diversas temáticas e tomadas de decisões. Diversas pesquisas no ramo da ciência possuem grande auxílio por parte das geotecnologias e da detecção remota. É possível integrar as geotecnologias a plataformas gratuitas como o Google Collaboratory (Google Colab) e o Google Earth Engine (GEE). Nesse sentido, o presente trabalho tem como proposta realizar a análise multitemporal referente aos anos de 2015 a 2021 das mudanças de temperatura ocasionadas por Ilhas de Calor a partir do TerraClimate, um conjunto de dados de clima mensal e balanço hídrico climático para superfícies terrestres globais e com auxílio do Landsat-8 detectar a vegetação presente na área utilizando o Google Collaboratory e o Google Earth Engine como plataformas de execução do processamento de dados.

Palavras-chave — Google Earth Engine, Processamento de Imagem, Ilhas de Calor, TerraClimate.

Abstract

Cartography is a science that enables studies on various topics and decision-making. Several researches in the branch of science have great help on the part of geotechnologies and remote sensing. It is possible to integrate geotechnologies to free platforms such as Google Collaboratory (Google Colab) and Google Earth Engine (GEE). In this sense, the present work proposes to carry out a multitemporal analysis referring to the years 2015 to 2021 of temperature changes caused by Heat Islands from TerraClimate, a set of monthly climate data and climate water balance for global land surfaces and with the help of Landsat-8 to detect the vegetation present in the area using Google Collaboratory and Google Earth Engine as platforms for the execution of data processing.

Key words — Google Earth Engine, Image Processing, Heat Islands, TerraClimate.

1. INTRODUÇÃO

Os centros urbanos dos municípios Brasileiros vem crescendo e apresentando diversas mudanças estruturais relacionadas ao seu uso e ocupação do solo.

As ilhas de calor urbana (ICU) são utilizadas para identificar as alterações do clima local a partir de sua espacialização. A alteração do clima pode ser provocada por diversos fatores que alteram a cobertura de dinâmica do solo, como: diminuição no percentual de vegetação, impermeabilização desordenada do solo, diminuição do fluxo de calor latente, aumento de fluxo de calor sensível e entre outros.

Segundo Buyantuyev et al. (2009), devido as transformações da cobertura do solo (substituição vegetal, terras agrícolas por superfícies impemeáveis) acarretam em ilhas de calor, dessa forma alterando o fluxo de energia da superfície, reduzindo a evapotranspiração e aumentando a absorção de energia solar.

A vegetação é uma grande aliada para a contenção do aumento de temperatura, ela gera melhoria na condição de vida dos centros urbanos, auxiliando na amenização de temperatura, consumo de energia e melhoria na qualidade do ar.

Realizando uma análise espaço-temporal sem contato físico direto, o sensoriamento remoto pode auxiliar na análise dos impactos naturais e antrópicos locais, permitindo o monitoramento e comparação temporal.

Com uma forte centralização vertical de bairros, principalmente na região litorânea, um tráfego intenso de veículos, pouca arborização e parques, grande parte do solo impermeabilizado, a cidade do Recife carrega essas fortes características de ilha de calor. Portanto, para extrair e identificar parâmetros que influenciam no aumento de temperatura, são utilizados aqui dados do dataset TerraClimate e dados do Landsat-8, onde será feita a relação entre os NDVI e da Temperatura de Superfície para identificar ilhas de calor na cidade

utilizando o Google Collaboratory e o Google Earth Engin como plataformas para execução de um grande número de dados.

2. ÁREA DE ESTUDO

Localizada no estado de Pernambuco, na parte litorânea do estado, a cidade do Recife está localizada ao norte com os municípios de Olinda e Paulista, ao sul de Jaboatão dos Guararapes, a leste com o Oceano Atlântico e a oeste com São Lourenço da Mata e Camaragibe, com coordenadas geográficas de: latitude $8^{\circ}04'03''$ e longitude de $34^{\circ}55'00''$.

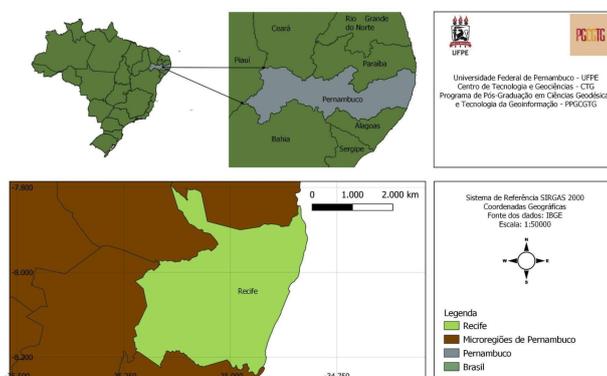


Figura 1. Mapa da Área de Estudo.

A região metropolitana tem uma forte caracterização geográfica possuindo uma baixa topografia, intensa urbanização (com foco em construções verticais na área litorânea), alta densidade demográfica, ocupação desordenada nas margens da cidade e altos valores ecológicos ignorados em todo o seu processo.

É importante notar que a mesma é altamente urbanizada, sofrendo influência dessa urbanização e centralização, com vias asfaltadas e grande influência de poluentes.

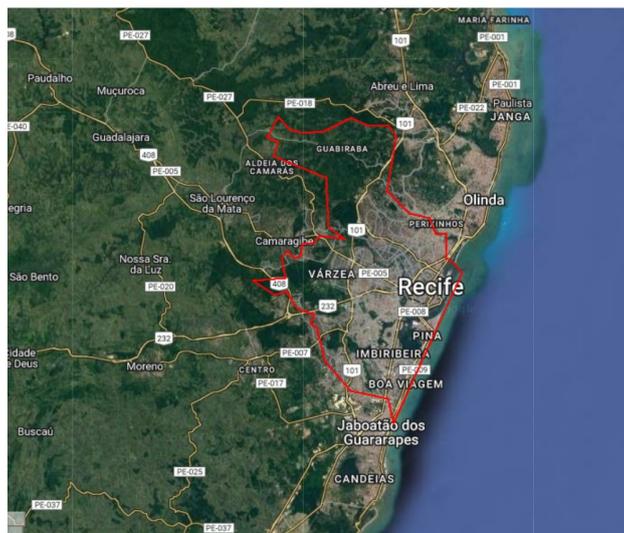


Figura 2. Área delimitada através do Google Collaboratory.

3. MATERIAIS E MÉTODOS

3.1. Materiais

Para realização do trabalho foram utilizados os seguintes materiais:

- Dados do dataset TerraClimate (Combinação de dados climatológicos de alta resolução espacial do conjunto de dados WorldClim, com dados variantes no tempo do CRU Ts4.0 (Unidade de Pesquisa Climática) e da JRA55 (Reanálise Japonesa de 55 anos).
- Imagens do satélite Landsat-8 sensor OLI (Operational Land Imager) faixa do visível;
- Arquivos em formato shapefile fornecidos pelo IBGE;
- Plataforma Google Collaboratory;
- Plataforma do Google Earth Engine (GEE);
- Software QuantumGis versão 3.22.5.

3.2 Métodos

A metodologia aplicada no desenvolvimento desta pesquisa pode ser visualizada no fluxograma (Figura 3).



Figura 3. Fluxograma da Metodologia.

3.3 Processamento de dados

O processamento dos dados referentes a temperatura e ao NDVI foram realizados utilizando o software Google Earth Engine vinculado ao Google Collaboratory. Abaixo é possível visualizar trechos do código em linguagem Python, utilizados para processar os dados (Figura 4).

```

# Calculo da média mensal

years = range(2015,2021)
months = range(1,13)

def calcMonthlyMean(imageCollection):
    mylist = ee.List([])
    for y in years:
        for m in months:
            collection_month = imageCollection.filter(ee.Filter.calendarRange(y, y, 'year'))\
                .filter(ee.Filter.calendarRange(m, m, 'month')).median()

            ##Após definir as variáveis anuais e mensais.
            ##Devemos adicionar essa coleção a lista
            mylist = mylist.add(collection_month.set('year', y)
                .set('month', m) #Inserir propriedade mês
                .set('date', ee.Date.fromYMD(y,m,1).format('YYYY-MM-dd')) #Inserir a propriedade data
                .set('system:time_start',ee.Date.fromYMD(y,m,1)))
            ##Fechemos a nossa função criando uma coleção de imagens
    return ee.ImageCollection.fromImages(mylist)

TerraClimate = ee.ImageCollection('IDAHO_EPSCOR/TERRACLIMATE')\
    .filterBounds(region)\
    .filterDate('2015-01-01', '2020-12-31')\
    .select('tmnx')
  
```

Figura 4. Média mensal para aplicação no TerraClimate.

O código gerado permite que seja trabalhado um grande quantitativo de imagens, onde é selecionado por meio do código com conexão ao GEE as imagens e conseqüentemente a área de estudo e os anos de interesse. De forma que o mesmo não ocupa memória e processamento da máquina do usuário.

3.4 Índice de Vegetação por Diferença Normalizada - NDVI e Temperatura

O índice de Vegetação da Diferença Normalizada (NDVI) é um índice capaz de analisar plantações, monitoramento de secas, auxiliando na previsão de incêndios e entre outros. Segundo Rouse et al. (1973), o NDVI normaliza a razão simples para o intervalo de -1 a 1, seguindo a seguinte equação:

$$\text{NDVI} = (\text{NIR} - \text{RED}) / (\text{NIR} + \text{RED}) \quad (1)$$

O NDVI é calculado relacionando a reflectância medida nas porções do vermelho e infravermelho do espectro do eletromagnético das bandas 5 e 4 do Landsat-8.

Possuindo diversos dados de clima mensal e balanço hídrico climático para superfícies terrestres globais, o TerraClimate é um conjunto de dados (dataset), onde os mesmos encontram-se disponíveis em: https://developers.google.com/earth-engine/datasets/catalog/IDAHO_EPSCOR_TERRACLIMATE.

Esse dataset é disponibilizado na plataforma do Google Earth Engine e seus dados podem ser processados para realizar diversas análises.

Lemenkova (2022), afirma que o conjunto de dados do TerraClimate contribui para utilização de análises ambientais em ambientes de programação simples aplicadas para a cartografia, onde é possível representar parâmetros climáticos e ambientais com precisão e eficácia.

Através de interpolação climaticamente auxiliada e combinação de dados climatológicos de alta resolução espacial, é possível obter informações sobre a temperatura da superfície. Segundo Abatzoglou (2018), o TerraClimate possui um grande potencial de aplicação, onde o mesmo possui uma variedade de escalas espaciais, onde é possível trabalhar com séries temporais de clima local e balanço hídrico, até regiões onde são desprovidos de muitos dados.

O TerraClimate é utilizado para diversos fins de pesquisas, uma das aplicações foi feita por Abdi (2019), que projetava uma estrutura baseada em objetos para discernir florestas infestadas de insetos através de dados do Landsat-8 OLI e dados auxiliares a partir do TerraClimate para quantificar efeitos dos riscos climáticos. Lemenkova (2022), utiliza dados do TerraClimate com cobertura global para resolver o processamento de dados cartográficos, onde a aplicação foi o mapeamento baseado em script.

4. RESULTADOS

As análises permitem visualizar que a média do NDVI ao longo dos anos para a cidade do Recife estão em torno de $0,4 < x < 0,5$ (Tabela 1). Isso nos permite indicar que a presença de vegetação no Recife é baixa. Abaixo é possível visualizar os dados contendo os índices de vegetação para cada ano.

Segundo Liu & Ming (2000), os valores de NDVI maiores que 0,6 estão associados à vegetação mais densa, enquanto que as que variam entre 0,3 e 0,6 apresentam uma cobertura menos densa ou áreas que estão próximas a regiões verdes.

ANO	NDVI Min	NDVI Médio	NDVI Max
2015	-0.938	0.420	0.993
2016	-0.831	0.481	0.998
2017	-0.836	0.437	0.984
2018	-0.906	0.431	0.995
2019	-0.875	0.468	0.998
2020	-0.807	0.456	0.997
2021	-0.874	0.469	0.998

Tabela 1. Análise do NDVI ao longo de 6 anos.

É possível observar que a temperatura média anual na cidade do Recife varia entre 27 graus celsius e 29 graus celsius, outro fato importante a ser observado é que a media máxima chegou a 29 graus, indicando um aumento de temperatura quando comparado as análises dos primeiros anos.

Segundo Filgueiras (2022), algumas áreas do globo possuem poucos dados de solo obtidos para o desenvolvimento do TerraClimate, como o caso da região do Brasil, isso significa dizer que existe uma validação limitada dos dados. Portanto correlacionar as informações da fonte de dados do TerraClimate com o landsat-8 se torna eficaz para entender a dinâmica dos resultados.

ANO	Temp Min	Temp Médio	Temp Max
2015	27,666	28,17	28,4
2016	28	28,286	28,533
2017	27,35	27,847	28,075
2018	28	28,788	29,008
2019	28,208	28,712	28,983
2020	28,341	28,843	29,125
2021	28,116	28,594	28,858

Tabela 2. Análise do NDVI ao longo de 6 anos.

Abaixo é possível observar os gráficos referentes as análises dos dados anuais (Figura 5 e 6).

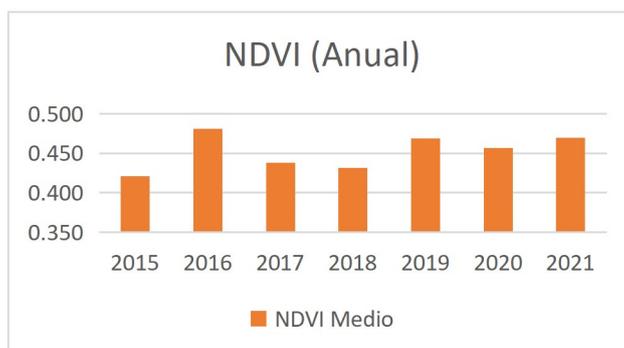


Figura 5. Gráfico do NDVI ao longo de 6 anos.

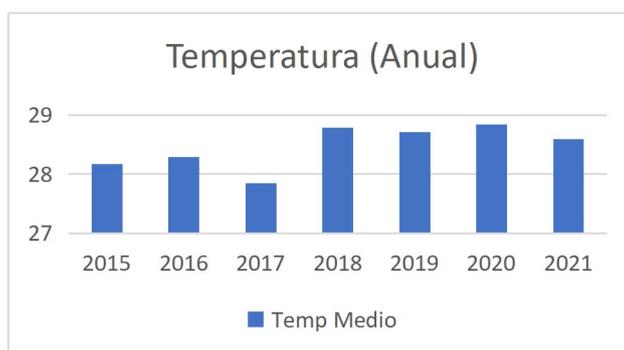


Figura 6. Gráfico de Temperatura ao longo de 6 anos.

Abaixo é apresentado o gráfico da média mensal para os anos de 2015 a 2021, considerando a média para cada mês (Figura 7).

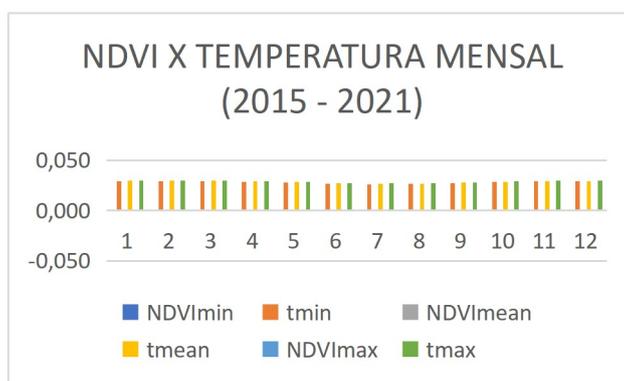


Figura 7. Gráfico do NDVI x TEMPERATURA.

5. CONCLUSÕES

A partir do desenvolvimento do trabalho e seus respectivos resultados, é importante ressaltar alguns pontos: I) A relação do índice de Vegetação e a temperatura nas grandes cidades; II) Correlação entre os dados advindos do TerraClimate e Landsat-8; III) A necessidade de uma série temporal maior para entender com mais eficácia o desempenho do TerraClimate para a região brasileira.

Através do tratamento e processamento das imagens de um grande número de dados do satélite Landsat-8 e do TerraClimate através do Google Earth Engine e Google Collaboratory, os dados apresentados na cidade do

Recife possuem diferenças significativas ao longo dos anos referentes a sua temperatura. Isso mostra a relação entre os baixos índices de vegetação e a temperatura, onde é possível relacionar que, os baixos índices de vegetação influenciam nas altas temperaturas, que favorecem no aumento de temperatura e conseqüentemente no aparecimento de ilhas de calor na região.

Vale ressaltar que os dados do TerraClimate possuem limitação de quantidade de dados de solo para algumas regiões do Brasil, o que permite entender que é importante correlacionar a informação que é utilizada do TerraClimate com outras fontes de dados, como no caso do Landsat-8.

Portanto, é necessário realizar uma avaliação mais profunda do conjunto de dados (para avaliar se as variáveis de entrada são confiáveis), utilizando uma série temporal maior e uma correlação mais densa das fontes de dados. Só assim será possível conhecer o desempenho do TerraClimate para o Território Brasileiro.

6. REFERÊNCIAS

- [1] Buyantuyev, A., Wu, J. Urban heat islands and landscape heterogeneity: linking spatiotemporal variations in surface temperatures to land-cover and socioeconomic patterns. *Landscape Ecol* 25, 17–33 (2010). <https://doi.org/10.1007/s10980-009-9402-4>
- [2] ROUSE, J.W.; HAAS, R.H.; SCHELL, J.A.; DEERING, D.W. Monitoring vegetation systems in the great plains with ERTS. In: *EARTH RESOURCES TECHNOLOGY SATELLITE SYMPOSIUM*, 3., 1973, Washington. Proceedings. Washington: NASA, 1973. v.1, p.309-317.
- [3] LEMENKOVA, Polina. Mapping Climate Parameters over the Territory of Botswana Using GMT and Gridded Surface Data from TerraClimate. In: *ISPRS International Journal of Geo-Information*. 2022, 11(9): 473.
- [4] ABATZOGLOU, J., DOBROWSKI, S., PARKS, S. et al. TerraClimate, a high-resolution global dataset of monthly climate and climate water balance from 1958-2015. *Sci Data*, 5, 170191 (2018).
- [5] ABDI, Omid. Climate-Triggered Insect Defoliators and Forest Fires Using Multitemporal Landsat and TerraClimate DATA in NE Iran: An Application of GEOBIA TreeNet and Panel Data Analysis. *Sensors*, 2019, 19, n. 18, 3965.

[6] MING, Tsay Chu; LIU, William Tse Horng. Estimativa de albedo, temperatura e NDVI do Estado de São Paulo através de dados AVHRR do satélite NOAA. In: Congresso Brasileiro De Meteorologia, 11. 2000. Rio de Janeiro. Anais...Rio de Janeiro: CBMET, 2000.. Artigos, p. 3843-3849. CD-ROM, On-line. Disponível em: <

<http://www.cbmet.com/cbm-files/> >. Acesso em: 27 set. 2014.

[7] FILGUEIRAS, Roberto. VENANCIO, P. Luan. Aleman, C. Catariny. CUNHA, F. F. Comparison and calibration of TerraClimate climatological variables over the Brazilian territory. In: Journal of South American Earth Sciences, V. 117, 2022. 102882.

APÊNDICE B – Integração do Google Earth Engine e Google Collaboratory no Processamento Digital de Imagens: Uma abordagem prática para análise espacial com grande volume de dados

Sabrina de Oliveira Gomes – Universidade Federal de Pernambuco, UFPE, E-mail: sabrina.gomes@ufpe.br

Ana Lúcia Candeias – Universidade Federal de Pernambuco, UFPE, E-mail: ana.candeias@ufpe.br

GT-06: Gestão & Conservação dos Recursos

Introdução

O processamento de grandes volumes de dados é um desafio comum enfrentado por muitos profissionais e pesquisadores em diversas áreas, como a cartografia, agricultura, ecologia, climatologia, hidrologia e entre outros. A análise de dados de imagens é uma ferramenta essencial para a compreensão de fenômenos e processos espaciais complexos, mas pode ser uma tarefa difícil e demorada quando se lida com grandes quantidades de dados.

Nesse contexto, o Google Earth Engine (GEE) e o Google Collaboratory (Google Colab) se apresentam como ferramentas poderosas para o processamento digital de imagens em grande escala. O Google Earth Engine é uma plataforma de processamento de dados geoespaciais baseada em nuvem, que permite acessar e processar uma grande variedade de dados de sensoriamento remoto de diferentes fontes, como satélites e drones. Já o Google COLAB é uma plataforma de desenvolvimento em nuvem baseada em notebooks do Jupyter, que possibilita a criação de scripts em Python e o uso de bibliotecas de processamento de imagens e dados geoespaciais.

A integração dessas duas ferramentas pode permitir uma análise mais eficiente e ágil de grandes volumes de imagens, possibilitando a geração de resultados precisos e de alta qualidade. O trabalho apresenta as funcionalidades das plataformas, mais especificamente, como objetivo de identificar e avaliar como a plataforma pode auxiliar no contexto de análise de dados em um estudo de caso analisando uma série temporal da presença de vegetação a partir do NDVI (*Normalized Difference Vegetation Index*). A área escolhida é o Recife, Pernambuco.

Referencial Teórico

O Google Earth Engine (GORELICK et al, 2017) e o Google COLAB (VERDE, PATIAS, MALLINIS, 2022) se apresentam como ferramentas poderosas para o processamento digital de imagens em grande escala. O GEE tem como principal proposta fornecer dados espaciais a partir de uma plataforma totalmente gratuita, desenvolvimento de algoritmos para aplicação geoespacial, permitindo que seja feita uma automatização de um grande conjunto de dados espaciais em escala global. O Google Collaboratory, conhecido também como Google Colab, é um serviço de armazenamento em nuvem voltado para criação e execução de códigos em Python.

O sensoriamento remoto é uma tecnologia que torna possível obter imagens da superfície terrestre a distâncias remotas. Possuindo técnicas e ferramentas que são ideais para análise de dados da superfície terrestre. Os dados obtidos de forma remota servem de suporte para vários estudos sobre a superfície da Terra, principalmente quando se leva em consideração a possibilidade de realizar análises temporais.

Existem diversos autores que utilizam a detecção remota como base para estudos ambientais. Entre eles, Pessi et al. (2019) que estudou a relação do NDVI com a temperatura da superfície terrestre para o planejamento urbano, com o objetivo de entender a influência da vegetação para a temperatura nas cidades de médio porte. Freitas et al. (2022) utilizou o sensoriamento remoto aplicando análise de NDVI e Temperatura das imagens do LANDSAT-5 e Landsat-8 para identificar ilhas de calor no município de Juazeiro do Norte, realizando uma análise de serie temporal de 15 anos. Essas análises multitemporais envolvem várias imagens, tempo de processamento, e espaço físico de armazenamento.

O Índice de Vegetação por Diferença Normalizada (NDVI) possibilita identificar as áreas vegetadas ou não vegetadas e muito utilizado para uso e ocupação de terras.

A automatização do NDVI usando GEE e Google Colab para uma série temporal de imagens é apresentado nesse trabalho. A vantagem é trabalhar na nuvem e com os bancos de imagens sem necessidade de se fazer download e/ou ocupar espaço físico da máquina local. Além disso, o código pode ser reaproveitado para ser usado em outras áreas.

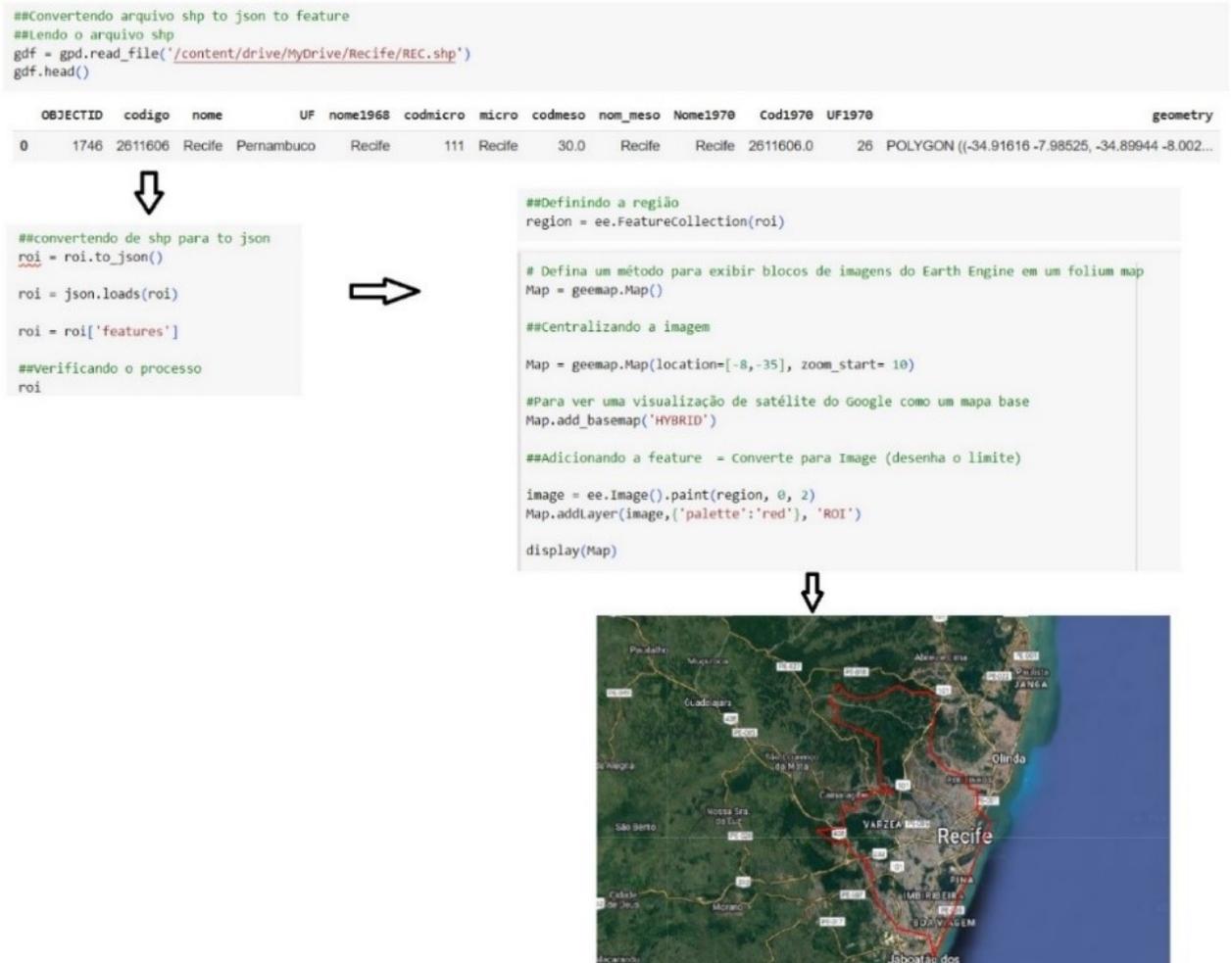
Para a estimativa da vegetação na área de estudo, optou-se pelo Índice de Vegetação por Diferença Normalizada (NDVI). Ele é obtido através da razão entre a diferença das refletividades do Infravermelho próximo (ρ_{IV}) e do vermelho (ρ_V), pela soma delas, abaixo é possível visualizar a fórmula (1) e, onde, ρ_{IV} é a reflectância do infravermelho (banda 5, para o OLI 8) e ρ_V é a reflectância do vermelho (banda 4, para o OLI 8).

$$NDVI = \frac{\rho_{IV} - \rho_V}{\rho_{IV} + \rho_V} \quad (1)$$

Materiais e Métodos

Nesta pesquisa foram utilizados os seguintes materiais: I. Imagens de satélite advindos do Landsat-8 na faixa do visível em formato .tiff; II. Shapefile advindos do IBGE, com a delimitação da área de interesse. Foram executados linhas de código para delimitar a área de interesse e integrar os dados do Landsat-8 na Figura 1. Na Figura 2 tem-se o código para remoção das nuvens. O cálculo do NDVI em linguagem de programação Python é mostrado na Figura 3.

Figura 1 – Seleção da área de estudo a partir do script em Python.



Fonte: Autor (2023)

Figura 2 – Código para máscara de nuvens.

```
##Máscara de nuvens
def maskL8sr(image):

    qaMask = image.select('QA_PIXEL').bitwise_and(int('11111', 2)).eq(0)
    saturationMask = image.select('QA_RADSAT').eq(0)
    opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2)
    thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0)

    return image.addBands(opticalBands, None, True)\
        .addBands(thermalBands, None, True).updateMask(qaMask)\
        .updateMask(saturationMask)
```

Fonte: Autor (2023)

Figura 3 – Função para o cálculo do NDVI.

##Função NDVI (ÍNDICE DE VEGETAÇÃO NORMALIZADA)

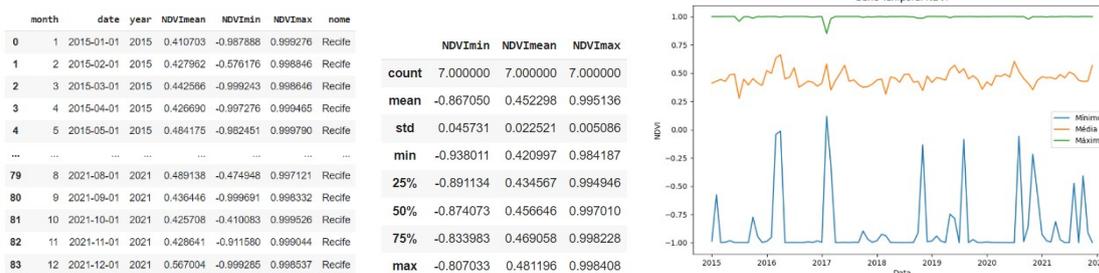
```
def NDVI (img):
    Ndvi_image = img.normalizedDifference(['SR_B5', 'SR_B4']).rename('NDVI')
    return img.addBands(Ndvi_image).clip(region).copyProperties(img, ["system:time_start"])
```

Fonte: Autor (2023)

Discussão e Resultados

A Figura 6 mostra a série temporal do NDVI, as estatísticas e o gráfico da série temporal mensal, de 2015 até 2022. Observa-se que o valor médio de NDVI, nesse período, ficou em 0.452298, e com um desvio padrão de 0.022521

Figura 6 – Série Temporal do NDVI, Estatísticas e Gráfico da série temporal do NDVI



Fonte: Autor (2023)

Considerações finais

Ao final deste trabalho, convém destacar que a maior relevância é que ele pode servir de apoio metodológico e teórico a futuros trabalhos que venham investigar e analisar problemas ambientais e que optem por utilizar dados de sensoriamento remoto para uma grande escala de tempo (série temporal), utilizando técnicas de sensoriamento remoto e processamento digital de imagem e base tecnológica do Google Earth Engine e Google Colaboratory.

Referências

FREITAS, C. A. S. de; MOREIRA, L. C. J.; CALOU, V. B. C.; FIGUEIREDO, J. V. de; SOUSA, R. R. de; SAMPAIO, G. G. C.; SARAIVA, K. R.; CARVALHO, C. M. de; SILVA, M. R. da. Remote Sensing applied to the analysis of the Heat Islands phenomenon in the city of Juazeiro do Norte - CE. *Research, Society and Development*, [S. l.], v. 11, n. 12, p. e237111234265, 2022. DOI: 10.33448/rsd-v11i12.34265. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/34265>. Acesso em: 13 jul. 2022.

GORELICK, N., HANCHER, M., DIXON, M., ILYUSHCHENKO, S., THAU, D., MOORE, R. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*. v. 202. p. 18–27, 2017. <https://doi.org/10.1016/j.rse.2017.06.031>

PESSI, D. D., PIMENTEL, C. F., CÂNDIDO, A. K. A. A., MIRANDA JUNIOR, P. L., MATOS DA SILVA, N. Análise da relação entre NDVI e a temperatura da superfície terrestre como técnica no planejamento urbano dos municípios. *Terr@ Plural*, [S. l.], v. 13, n. 3, p. 237–251, 2019. Disponível em: <https://revistas.uepg.br/index.php/tp/article/view/13100>. Acesso em: 5 de jun. 2023.

VERDE, N.; PATIAS, P.; MALLINIS, G. A Cloud-Based Mapping Approach Using Deep Learning and Very-High Spatial Resolution Earth Observation Data to Facilitate the SDG 11.7.1 Indicator Computation. *Remote Sens*. 2022, 14, 1011. <https://doi.org/10.3390/rs14041011>