

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

**DESENVOLVIMENTO DE PLATAFORMA DE
VISUALIZAÇÃO DE VEREDAS NO CERRADO BRASILEIRO**

**EDUARDO PETRONE MOTTA
FILLIPE PINHEIRO LIMA DA SILVA
GUSTAVO CÉSAR PRIETO**

São Paulo - SP, dezembro de 2024

**EDUARDO PETRONE MOTTA
FILLIPE PINHEIRO LIMA DA SILVA
GUSTAVO CÉSAR PRIETO**

**DESENVOLVIMENTO DE PLATAFORMA DE VISUALIZAÇÃO DE
VEREDAS NO CERRADO BRASILEIRO**

Projeto de Formatura apresentado junto ao programa de **Bacharelado em Engenharia** da **Escola Politécnica da Universidade de São Paulo**, como requisito parcial à obtenção do título de **Bacharel em Engenharia Elétrica**.

Orientador:

Prof. Dr. Pedro Luiz Pizzigatti Corrêa

Coorientadora:

Dra. Marina Jeaneth Machicao Justo

São Paulo - SP, dezembro de 2024

Dedicamos às nossas famílias e aos nossos amigos, os quais sempre nos apoiaram durante toda a graduação e foram alento nos momentos mais difíceis.

AGRADECIMENTOS

Ao Prof. Dr. Pedro Luiz Pizzigatti Corrêa e à Dra. Marina Jeaneth Machicao Justo por nos orientar, fornecer suporte e respaldo técnico para a elaboração deste trabalho.

A Wellington Dias de Queiroz, pesquisador do Laboratório de Ciência de Dados e Inteligência Artificial da Engenharia de Computação da Escola Politécnica da USP, pelo apoio fundamental nos últimos e importantíssimos ajustes do projeto.

Aos pesquisadores do INPE, especialmenete Alexandre Barbosa, pela colaboração e prestatividade para compreensão da etapa de ciência de dados.

*“Educação é uma descoberta progressiva de
nossa própria ignorância.”*

Voltaire

RESUMO

Este trabalho, desenvolvido como parte de um projeto de formatura, apresenta o desenvolvimento de uma plataforma computacional voltada para facilitar a visualização de veredas, ecossistemas característicos do bioma Cerrado brasileiro que desempenham um papel fundamental na regulação hídrica e na manutenção da biodiversidade. Frente às crescentes ameaças enfrentadas por essas áreas, o projeto adota uma abordagem tecnológica baseada no uso de imagens de satélite e ferramentas de análise geoespacial, promovendo a identificação automatizada das veredas e sua apresentação em uma interface acessível. O objetivo principal é fornecer uma plataforma que permita a visualização eficiente de dados geoespaciais e ambientais, subsidiando pesquisas e decisões estratégicas relacionadas à gestão sustentável e à preservação ambiental. A etapa de ciência de dados foi praticamente implementada por completo em momento anterior a esse trabalho, estruturada em arquitetura modular, com Python como linguagem principal. Foram integradas bibliotecas que possibilitaram o processamento de imagens geoespaciais, a análise de dados e o desenvolvimento de modelos preditivos. Adicionalmente, o projeto utilizou serviços da AWS (Amazon Web Services), incluindo EC2, para a execução de algoritmos de Inteligência Artificial e a hospedagem do sistema, e S3, para o armazenamento de dados, como imagens de satélite e resultados processados. Como resultado final, foi desenvolvida uma plataforma web interativa baseada no framework Django. Essa plataforma, hospedada na internet, tem a capacidade de permitir, quando as imagens estiverem disponíveis, a visualização das veredas identificadas por meio de um mapa dinâmico, proporcionando uma interface acessível voltada para pesquisadores, gestores ambientais e formuladores de políticas públicas. A ferramenta facilita o acesso direto aos dados geoespaciais processados e à análise das informações geradas, consolidando o impacto do projeto no suporte a decisões informadas relacionadas à conservação ambiental.

Palavras-chave: Engenharia, Computação, Ciência de Dados, Desenvolvimento Web, Cerrado, Veredas, Biodiversidade

ABSTRACT

This paper, developed as part of a graduation project, presents the development of a computational platform aimed at facilitating the visualization of “veredas,” characteristic ecosystems of the Brazilian Cerrado biome that play a crucial role in water regulation and biodiversity maintenance. Given the increasing threats faced by these areas, the project adopts a technological approach based on the use of satellite imagery and geospatial analysis tools, promoting the automated identification of veredas and their display on an accessible interface. The main goal is to provide a platform that allows for the efficient visualization of geospatial and environmental data, supporting research and strategic decisions related to sustainable management and environmental preservation. The data science phase was almost entirely implemented prior to this work, structured in a modular architecture, with Python as the primary language. Libraries that enabled the processing of geospatial images, data analysis, and the development of predictive models were integrated. Additionally, the project utilized services from AWS (Amazon Web Services), including EC2 for running artificial intelligence algorithms and hosting the system, and S3 for data storage, such as satellite images and processed results. As a final result, an interactive web platform based on the Django framework was developed. This platform, hosted on the internet, has the capacity to allow, when images are available, the visualization of identified veredas through a dynamic map, providing an accessible interface aimed at researchers, environmental managers, and public policy makers. The tool facilitates direct access to processed geospatial data and analysis of the generated information, consolidating the project’s impact on supporting informed decisions related to environmental conservation.

Keywords: Engineering, Computing, Data Science, Web Development, Cerrado, Veredas, Biodiversity

LISTA DE FIGURAS

Figura 1.1 – Disposição da Área Nexus no Brasil.	24
Figura 2.1 – Disposição geográfica do Cerrado Brasileiro.	27
Figura 2.2 – Fotografia de uma vereda.	29
Figura 2.3 – Representação esquemática de uma vereda.	29
Figura 2.4 – Representação ilustrativa de campos de estudo da Inteligência Artificial.	31
Figura 2.5 – Representação esquemática de Redes Neurais.	33
Figura 4.1 – Plataforma de visualização de desenvolvida por GIUZIO et al. (2022).	55
Figura 4.2 – Representação esquemática do fluxo de trabalho utilizado no artigo.	56
Figura 8.1 – Plataforma de visualização de desenvolvida ao longo do projeto.	78
Figura 8.2 – Seleção de veredas na plataforma.	79
Figura A.1 – Representação ilustrativa do fluxo de trabalho.	84

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CD	Entrega Contínua
CI	Inetgração Contínua
CNN	Convolutional Neutral Networks
CSP	Content Policy
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheet
DaaS	Data as a Service
DOM	Document Object Model
DRY	Don't Repeat Yourself
EC2	Elastic Compute Cloud
FMA	Floof Mapping Algorithm
FPR	Taxa de Falsos Positivos
GAN	Generative Adversarial Networks
GCP	Google Cloud Platform
GEE	Google Earth Engine
GPU	Graphics Processing Unit
HAND	Altura Acima da Drenagem mais Próxima
HTML	HyperText Markup Language
HTTPS	Hypertext transfer protocol secure
IA	Inteligência Aritificial
IaaS	Infrastructure as a Service
IBGE	Instituto Brasileiro de Geografia e Estatística
INPE	Instituto Nacional de Pesquisas Espaciais
IoT	Internet das Coisas
LASSO	Operador de Seleção e Encolhimento Absoluto Mínimo
LESS	Leaner Style Sheets
LGPD	Lei Geral de Proteção de Dados Pessoais

LSTM	Long Short-Term Memory
MFA	Autenticação Multifator
MNDWI	Índice Modificado de Diferença Normalizada da Água
MVC	Model-View-Controller
MVT	Modelo-View Template
MVVM	Model-View-ViewModel
NDVI	Índice de Vegetação por Diferença Normalizada
NoSQL	Not Only SQL
ORM	Object-Relational Mapping
OWASP	Open Worldwide Application Security Project
PaaS	Platform as a Service
PWA	Progressive Web Applications
REST	Representational State Transfer
RNN	Recurrent Neural Networks
S3	Simple Storage Service
SaaS	Software as a Service
SASS	Syntactically Awesome Style Sheets
SEO	Search Engine Optimization
SP	São Paulo
SPA	Single Page Applications
SQL	Structured Query Language
TPR	Taxa de Verdadeiros Positivos
TPU	Tensor Processing Unit
UI	Interface de Usuário
UX	Experiência de Usuário
W3C	World Wide Web Consortium
WHATWG	Web Hypertext Application Technology Working Group
WRD	Desastres Relacionados à Água
XSS	Cross-Site Scripting

SUMÁRIO

1	INTRODUÇÃO	22
1.1	Motivação	22
1.2	Objetivos	25
2	ASPECTOS CONCEITUAIS	27
2.1	Cerrado	27
2.2	Veredas	28
2.3	Inteligência Artificial	30
2.3.1	Machine Learning	31
2.3.2	Deep Learning	32
2.4	Desenvolvimento WEB	34
2.4.1	Back-end	35
2.4.2	Front-end	36
3	TECNOLOGIAS	39
3.1	Amazon Web Services	39
3.1.1	Amazon EC2	41
3.1.2	Amazon S3	42
3.1.3	Amazon SageMaker	44
3.2	Google Earth Engine	45
3.3	Python	46
3.3.1	Django	47
3.4	HTML	49
3.5	CSS	50
3.6	JavaScript	51
4	TRABALHOS RELACIONADOS	54
4.1	GIUZIO ET AL. (2023)	54

4.2	MEHMOOD ET AL. (2021)	55
5	METODOLOGIA	58
5.1	Fluxo de Trabalho	58
5.2	Requisitos	59
5.2.1	Requisitos Funcionais	60
5.2.2	Requisitos Não Funcioanis	61
6	DESENVOLVIMENTO BACK-END	63
6.1	Escolha e Justificativas das Tecnologias Utilizadas	63
6.1.1	Python e Django	63
6.1.2	Google Earth Engine	65
6.2	Arquitetura do Sistema	66
6.2.1	<i>Models</i> (Modelos)	66
6.2.2	<i>Views</i> (Visões)	66
6.2.3	<i>Controllers</i> (Contraladores)	67
6.3	Implementação	67
6.3.1	Aquisição de Arquivos do Amazon S3	68
6.3.2	Extração e Conversão de Coordenadas Geográficas de Imagens .tif	69
6.3.3	Aquisição e Conversão de Imagens de Satélite para o Google Earth Engine	70
6.3.4	Transferência de Imagens ao Google Earth Engine	72
6.3.5	Aplicação do Google Earth Engine	73
7	DESENVOLVIMENTO FRONT-END	75
7.1	Escolha e Justificativas das Tecnologias Utilizadas	75
7.1.1	HTML	75
7.1.2	JavaScript	76
7.1.3	CSS	76
7.1.4	Justificativa da Escolha	77

8	CONCLUSÃO	78
8.1	Resultados	78
8.2	Contribuições	79
8.3	Trabalhos Futuros	80
	REFERÊNCIAS	82
	APÊNDICE A – FLUXO DE TRABALHO DO PROJETO	84

1 INTRODUÇÃO

Este trabalho é submetido à Escola Politécnica da Universidade de São Paulo como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica. Desenvolvido em colaboração com pesquisadores do Instituto Nacional de Pesquisas Espaciais (INPE), participantes do Projeto Nexus, e com o Grupo de Estudos, Pesquisa e Extensão em Big Data da Escola Politécnica da USP, o trabalho consiste no desenvolvimento e disponibilização, na internet, de uma plataforma interativa capaz de mostrar graficamente a evolução da disposição das veredas situadas na área Nexus.

A área Nexus refere-se a uma região brasileira composta pelos biomas Caatinga e Cerrado, localizada entre as bacias hidrográficas do Rio São Francisco (SFRB) e do Rio Parnaíba. Este território é de grande relevância devido à sua importância agrícola e energética, sendo objeto do Projeto Nexus, que busca promover o desenvolvimento sustentável dessa região estratégica (NEXUS, 2024).

1.1 MOTIVAÇÃO

O Brasil é o país mais biologicamente diverso do mundo, estando classificado no topo entre os 17 países megadiversos do mundo, atrás apenas da Indonésia em termos de endemismo de espécies. O país abriga dois *hotspots* de biodiversidade (a Mata Atlântica e o Cerrado), seis biomas terrestres e três grandes ecossistemas marinhos. Atualmente, são conhecidas pelo menos 103 mil espécies de animais e 43 mil espécies de plantas, o que representa 70% das espécies animais e vegetais catalogadas no mundo. Estima-se que o Brasil abrigue entre 15 e 20% da diversidade biológica do planeta, com o maior número de espécies endêmicas em escala global. A biodiversidade do Brasil continua a se expandir, com uma média de 700 novas espécies de animais descobertas a cada ano (CDB, 2024).

Nesse contexto, sabe-se que a biodiversidade é um recurso extremamente importante, não apenas pelos serviços ambientais que fornece, mas também pelas oportunidades que apresenta para o desenvolvimento e uso sustentável. Representado por mais de 200 povos indígenas e 170 idiomas, o Brasil também é megadiverso do ponto de vista cultural. Este grande número de comunidades e aldeias locais possui um conhecimento considerável sobre espécies de flora e fauna, incluindo sistemas tradicionais de manejo desses recursos naturais. A contribuição dessas comunidades é, portanto, fundamental para a conservação e o uso sustentável dos recursos genéticos e biológicos do país (CDB, 2024).

Como reflexo do aumento do reconhecimento da abundância dos recursos naturais, cresce também a atenção dada à relevância do Brasil na gestão ambiental, tanto em

discussões do setor público quanto privado. Simultaneamente, temas associados à ciência da computação, como inteligência artificial e uso de dados de sensoriamento remoto, estão em destaque, ampliando os debates sobre o uso de tecnologia na preservação ambiental.

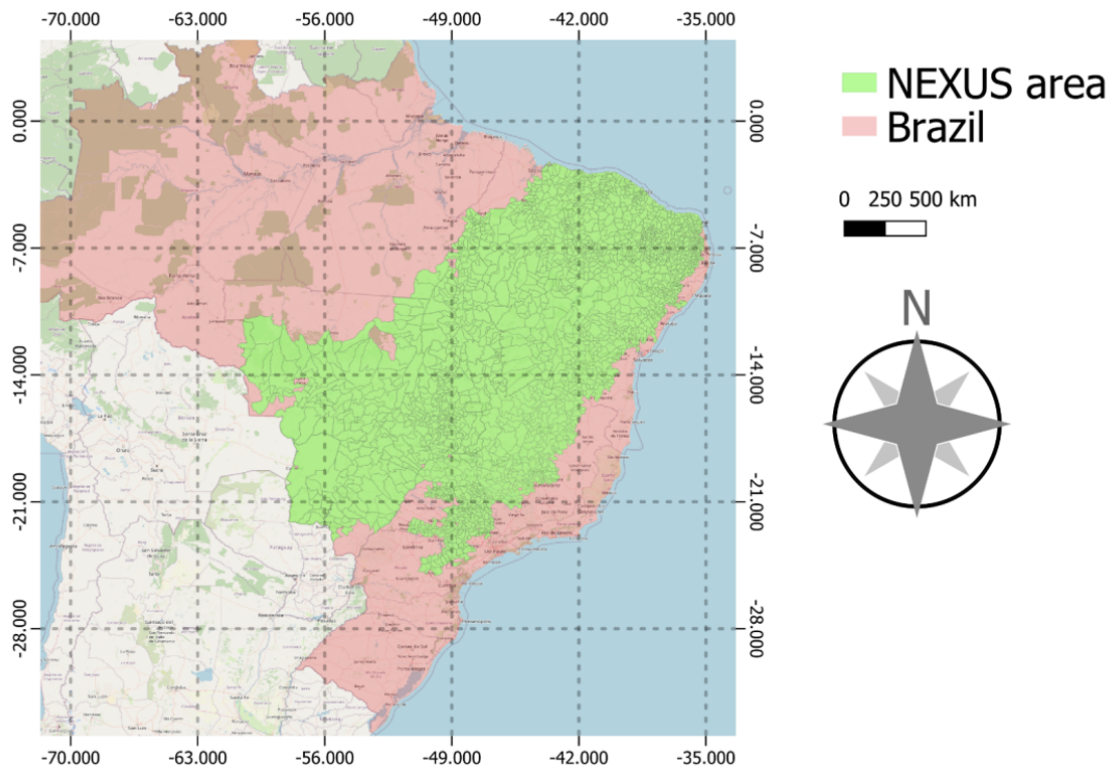
Neste cenário, destaca-se a importância da implementação de políticas eficazes de preservação ambiental que garantam a conservação desses recursos para as gerações presentes e futuras. Em 2018, o governo brasileiro reforçou o valor estratégico da conectividade ecológica, considerando-a essencial na definição de áreas prioritárias para conservação, restauração e uso sustentável da vegetação nativa.

Com essa visão, o Projeto Nexus, coordenado pelo INPE e patrocinado pela FAPESP, foi criado com o objetivo de desenvolver métodos que promovam a conectividade entre unidades da paisagem natural e ofereçam caminhos para a transição para um futuro sustentável nos biomas Cerrado e Caatinga. O projeto adota uma abordagem participativa que integra métodos qualitativos e quantitativos das ciências naturais e sociais, visando conciliar as dimensões econômica, social e ambiental da sustentabilidade (NEXUS, 2024).

Focado inicialmente no Cerrado, o projeto busca contribuir para a gestão sustentável das savanas neotropicais, propondo estratégias para o uso racional de seus recursos naturais essenciais, como os estoques de terras para expansão agrícola e o potencial para geração de energia solar e eólica. Entre suas metas, destacam-se a criação de indicadores de sustentabilidade, a projeção de cenários futuros sobre o uso do solo e mudanças climáticas, e a disseminação de dados e resultados por meio de uma plataforma acessível à comunidade científica e à sociedade em geral. Esses esforços visam consolidar um modelo interdisciplinar de pesquisa capaz de orientar ações para a sustentabilidade e a preservação da biodiversidade (NEXUS, 2024).

A região do Nexus, situada entre os biomas da Caatinga e do Cerrado, inclui áreas de bacias hidrográficas importantes, como as dos rios São Francisco e Parnaíba, e ocupa cerca de 3,4 milhões de quilômetros quadrados, representando aproximadamente 40% do território nacional. Com mais de 2.500 municípios, a região é o centro da agricultura comercial brasileira e se destaca por sua diversidade de recursos naturais. A crescente importância da preservação desse território incentivou o Projeto Nexus a buscar métodos de identificação e análise de elementos ambientais relevantes, como as veredas do Cerrado.

Figura 1.1 – Disposição da Área Nexus no Brasil.



Fonte: GIUZIO et al. (2022).

As veredas são áreas úmidas frequentemente compreendidas como florestas de galeria em zonas alagadiças. Essas áreas possuem importância estratégica única para a conservação, devido a diversos fatores, incluindo os serviços ecossistêmicos que oferecem, como o armazenamento e a filtragem do excesso de água da chuva. Por desempenharem esse papel, as veredas funcionam como reservatórios de grandes quantidades de carbono em seus solos orgânicos. Além disso, as veredas estão diretamente associadas aos sistemas de drenagem das nascentes das principais bacias hidrográficas do Cerrado, que, por sua vez, correspondem a cerca de dois terços das bacias hidrográficas brasileiras.

Para realizar a identificação, o projeto adotou plataformas de sensoriamento remoto, com destaque para o *Google Earth Engine* (GEE). O GEE oferece uma integração ampla de dados de sensoriamento remoto em alta resolução, acumulados ao longo de mais de quatro décadas, sendo uma ferramenta crucial para a estimativa de indicadores ambientais e sociais. Utilizando técnicas de *Deep Learning* e aprendizado por transferência (*transfer learning*), o projeto busca desenvolver um modelo de identificação das veredas, que possibilite compreender sua distribuição e degradação ao longo do tempo.

O uso de modelos de *Deep Learning* para identificar e analisar a evolução da

distribuição das veredas no cerrado apresenta vantagens significativas, como a eliminação da necessidade de coleta presencial de dados e a automação da análise, resultando em uma compreensão mais abrangente e dinâmica das mudanças ambientais.

1.2 OBJETIVOS

O objetivo deste trabalho é desenvolver uma plataforma dedicada à identificação e monitoramento temporal e espacial das veredas no cerrado brasileiro, utilizando modelos de aprendizado profundo. A estrutura do projeto está baseada em duas atividades principais: o experimento de Ciência de Dados, voltado para a criação dos modelos de identificação das veredas, e o desenvolvimento da plataforma de visualização.

Com o avanço das pesquisas do projeto Nexus, buscou-se desenvolver métodos eficazes para identificar e classificar os elementos de biodiversidade na região de estudo, incluindo um algoritmo específico para avaliar as condições atuais e futuras das veredas no cerrado. Para isso, foram avaliadas plataformas de processamento de dados geoespaciais, e o Google Earth Engine (GEE) destacou-se por sua vasta base de dados de sensoriamento remoto de alta resolução, acumulando registros de mais de quatro décadas. Esta plataforma viabiliza a aplicação de modelos de Deep Learning, utilizando grandes volumes de dados para gerar indicadores ambientais de forma precisa.

Essa abordagem visa ampliar a compreensão da deterioração das veredas no cerrado, tanto em escala espacial quanto temporal, reduzindo a necessidade de monitoramento constante em campo e automatizando a análise de dados. Assim, propõe-se o desenvolvimento de uma plataforma para visualização das áreas identificadas como veredas, permitindo que pesquisadores, autoridades e outros interessados possam explorar a evolução dessas áreas na região do Nexus.

A primeira atividade, o experimento de Ciência de Dados, já foi implementada em fases anteriores do projeto, incluindo o desenvolvimento de um sistema para a identificação de veredas utilizando redes neurais convolucionais, inspirado na arquitetura UNet. Esse sistema, desenvolvido por pesquisadores do Projeto Nexus, captura tanto características gerais quanto detalhes específicos das veredas, empregando múltiplas iterações e técnicas de transferência de aprendizado para aprimorar a acurácia dos resultados. A metodologia está estabelecida, restando apenas a atualização dos modelos com dados mais recentes, o que contribuirá para a continuidade e atualidade das análises.

O presente trabalho concentra-se, assim, na segunda atividade: o desenvolvimento da plataforma de visualização. Esta plataforma contará com uma interface interativa e intuitiva, permitindo a exploração dos dados de identificação de veredas ao longo de

diferentes períodos e áreas. O *backend* será responsável por processar as solicitações e executar os modelos atualizados. A arquitetura da plataforma proporcionará uma base aberta de dados, permitindo consultas, solicitações de novas identificações e o *download* dos resultados gerados.

A relevância deste projeto para a sociedade e para o meio ambiente é significativa. Ao facilitar o acesso e a interpretação de dados sobre as veredas, o sistema permite que pesquisadores e gestores ambientais monitorem áreas de grande importância ecológica, auxiliando na identificação de tendências de degradação ou recuperação dessas regiões.

Para a sociedade, os dados fornecidos pela plataforma apoiam a conscientização sobre a importância das veredas, que são fontes de biodiversidade, reguladores climáticos e reservatórios hídricos essenciais para a sustentabilidade do cerrado.

Assim, um dos objetivos fundamentais do projeto é proporcionar uma ferramenta acessível e poderosa que contribua para o desenvolvimento de políticas públicas e práticas de conservação ambiental, garantindo a preservação e o uso sustentável das veredas, essenciais para a saúde ambiental e o bem-estar das comunidades dependentes desse ecossistema.

2 ASPECTOS CONCEITUAIS

Esta seção apresenta diversos conceitos essenciais relacionados aos problemas e soluções discutidos neste projeto. Com um viés mais abrangente, a seção tem como objetivo elucidar conceitos relevantes para o trabalho relacionados aos campos de estudo da geografia, da ciência de dados e do desenvolvimento web.

2.1 CERRADO

O Cerrado brasileiro é o segundo maior bioma da América do Sul, abrangendo aproximadamente 2 milhões de quilômetros quadrados, o que corresponde a cerca de 23% do território nacional. Situado principalmente no Planalto Central, o Cerrado se estende por diversos estados, incluindo Goiás, Mato Grosso, Mato Grosso do Sul, Minas Gerais, e partes de São Paulo, Bahia, Tocantins, Maranhão e Piauí. Este bioma é reconhecido por sua grande biodiversidade, sendo considerado uma das savanas mais ricas do mundo em termos de espécies vegetais e animais.

Figura 2.1 – Disposição geográfica do Cerrado Brasileiro.



Fonte: IBGE.¹

¹ Disponível em: <<https://www.poder360.com.br/opiniao/a-dupla-riqueza-do-cerrado-brasileiro/>>. Acesso em: 15 de setembro, 2024.

O Cerrado é caracterizado por um clima tropical sazonal, com duas estações bem definidas: uma estação chuvosa, que ocorre de outubro a abril, e uma estação seca, que se estende de maio a setembro. Esse regime climático, aliado à sua geologia e topografia, molda uma vegetação única, que varia desde campos limpos e cerrados típicos, com vegetação baixa e esparsa, até formações florestais densas, conhecidas como catanduvras. As espécies vegetais do Cerrado possuem adaptações notáveis ao fogo e à seca, como raízes profundas, cascas espessas e folhas resistentes, que lhes permitem sobreviver em condições ambientais extremas.

A fauna do Cerrado também é altamente diversificada, abrigando espécies endêmicas e ameaçadas de extinção, como o lobo-guará (*Chrysocyon brachyurus*), o tatu-canastra (*Priodontes maximus*), e o tamanduá-bandeira (*Myrmecophaga tridactyla*). Além de sua riqueza biológica, o Cerrado desempenha um papel crucial na hidrologia do Brasil. Muitas das principais bacias hidrográficas brasileiras, como as do rio Tocantins, São Francisco e Paraná, têm suas nascentes nas áreas de Cerrado, o que torna esse bioma essencial para o abastecimento hídrico de grandes regiões do Brasil.

No entanto, o Cerrado enfrenta graves ameaças, principalmente devido à expansão agrícola e pecuária, que nas últimas décadas resultou na conversão de vastas áreas naturais em plantações de soja, milho, e pastagens. Este processo de desmatamento e degradação ambiental tem levado à perda acelerada de biodiversidade e à alteração dos regimes hídricos locais, comprometendo a sustentabilidade ecológica do bioma.

A preservação deste bioma é fundamental não apenas para a manutenção da biodiversidade brasileira, mas também para a proteção dos recursos hídricos e o equilíbrio climático regional. Estratégias de conservação eficazes exigem uma abordagem integrada que concilie a proteção ambiental com o desenvolvimento econômico sustentável, garantindo a viabilidade a longo prazo desse ecossistema vital.

2.2 VEREDAS

O termo “vereda” apresenta muitas definições, a mais comum sendo um caminho estreito, senda. Neste trabalho, a definição de vereda que nós usaremos é a de um pequeno córrego ou riacho presente no bioma do Cerrado, que gera um terreno brejoso e geralmente é cercado por vegetais adaptadas a solos encharcados, como as palmeiras do gênero *Mauritia*, popularmente conhecidas como buritis.

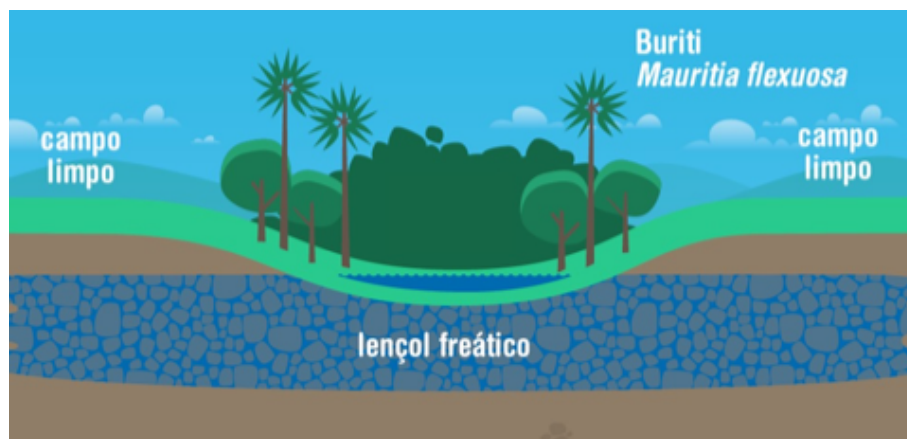
Figura 2.2 – Fotografia de uma vereda.



Fonte: Autor desconhecido.²

Essas áreas desempenham um papel crucial na hidrologia do Cerrado, atuando como reguladoras do ciclo hidrológico, uma vez que são responsáveis pela recarga de aquíferos e pelo abastecimento de cursos d'água que integram grandes bacias hidrográficas brasileiras. Sua importância para hidrologia local é alta, sendo a desertificação do bioma ligada à degradação das veredas por causa de agropecuária, exploração da argila presente no solo brejoso da vereda, urbanização, etc (ARAÚJO et al., 2002).

Figura 2.3 – Representação esquemática de uma vereda.



Fonte: Página "Árvore, Ser Tecnológico" no Facebook.³

² Disponível em: <<https://advambiental.com.br/artigo/inexistencia-de-area-de-preservacao-permanente-de-vereda/>>. Acesso em: 15 de setembro, 2024.

³ Disponível em: <<https://www.facebook.com/arvoresertecnologico/posts/no-cerrado-as-veredas-sao-areas-umidas-produtoras-de-agua-nesses-locais-em-que-a/1401862339956368>>. Acesso em: 15 de setembro, 2024.

Do ponto de vista ecológico, as veredas são ambientes de alta biodiversidade, abrigando uma variedade de espécies vegetais e animais, muitas das quais endêmicas ou ameaçadas de extinção. A vegetação das veredas é composta por uma estratificação de plantas herbáceas, arbustivas e arbóreas, que variam conforme a proximidade com o curso d'água central. Essa diversidade de flora proporciona habitats específicos para uma gama de fauna adaptada a essas condições particulares.

Além de seu valor para o ambiente em que estão inseridas, as veredas têm importância socioeconômica para as comunidades locais, que delas dependem para atividades como o extrativismo de frutos, folhas e madeira, além do uso da água para irrigação e consumo. No entanto, essas áreas estão sob crescente ameaça devido à expansão da fronteira agrícola, à urbanização e às mudanças climáticas, que alteram o regime hídrico e podem levar à degradação desses ecossistemas.

Dada sua relevância para a conservação da biodiversidade e para a manutenção dos recursos hídricos no Cerrado, as veredas têm sido objeto de estudos científicos e políticas públicas voltadas para a sua preservação. A compreensão da dinâmica ecológica dessas áreas e o desenvolvimento de estratégias eficazes para sua proteção são fundamentais para assegurar a sustentabilidade ambiental do Cerrado como um todo.

2.3 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) tem como objetivo desenvolver dispositivos capazes de simular as faculdades humanas de raciocínio, percepção, tomada de decisões e resolução de problemas, em suma, a capacidade de demonstrar inteligência.

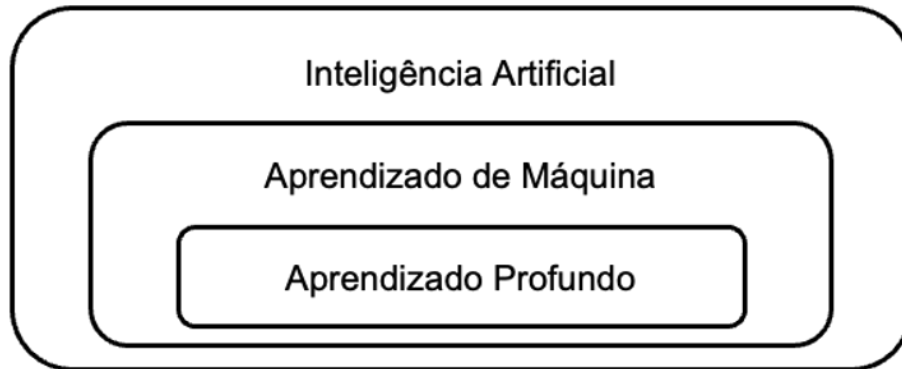
No âmbito da Ciência da Computação, a IA é frequentemente definida como o estudo de 'agentes inteligentes', ou seja, sistemas que são capazes de interpretar seu ambiente e tomar decisões que aumentem a probabilidade de atingir seus objetivos de maneira eficiente. Em um sentido mais amplo, o termo 'inteligência artificial' é atribuído a tecnologias que conseguem realizar tarefas que normalmente requerem inteligência humana, tais como a aprendizagem e a resolução de problemas (SHINDE; SHAH, 2018).

Um aspecto crucial da IA é o aprendizado de máquina ('machine learning'), que se configura como um dos principais subcampos da IA. Esta área de estudo se concentra no desenvolvimento de algoritmos que permitem às máquinas melhorar seu desempenho em tarefas específicas com base em dados.

Dentro do aprendizado de máquina, o aprendizado profundo se destaca como uma área de inovação particular. Este subconjunto se diferencia por sua capacidade de processar

e aprender a partir de grandes volumes de dados, utilizando redes neurais artificiais que imitam, em parte, as conexões neuronais do cérebro humano. À medida que o campo do aprendizado de máquina se expande, a pesquisa contínua e a aplicação de novas tecnologias continuam a ser fundamentais para a evolução da IA e para o aumento de sua relevância em diversos domínios.

Figura 2.4 – Representação ilustrativa de campos de estudo da Inteligência Artificial.



Fonte: Elaborado pelos autores.

2.3.1 Machine Learning

O aprendizado de máquina, ou machine learning, é um subcampo essencial da IA que se dedica ao desenvolvimento de técnicas computacionais voltadas para a aprendizagem, bem como à construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado consiste em um programa de computador que toma decisões com base em experiências acumuladas por meio da solução bem-sucedida de problemas anteriores.

O aprendizado de máquina é um ramo da ciência da computação que, de forma geral, visa permitir que os computadores 'aprendam' sem serem diretamente programados. Ele tem suas origens no movimento de inteligência artificial dos anos 1950 e enfatiza objetivos e aplicações práticas, particularmente previsão e otimização. Os computadores 'aprendem' no aprendizado de máquina ao melhorarem seu desempenho em tarefas por meio da 'experiência' (BI et al., 2019).

Na prática, 'experiência' geralmente significa ajustar-se aos dados; assim, não há uma fronteira clara entre o aprendizado de máquina e as abordagens estatísticas. De fato, se uma determinada metodologia é considerada 'aprendizado de máquina' ou 'estatística' muitas vezes reflete tanto sua história quanto diferenças reais, e muitos algoritmos (por exemplo, o operador de seleção e encolhimento absoluto mínimo (LASSO), regressão por etapas) podem ou não ser considerados aprendizado de máquina, dependendo de quem você pergunta. Ainda assim, apesar das semelhanças metodológicas, o aprendizado de

máquina é filosoficamente e praticamente distinto. Com o risco de uma (considerável) simplificação, o aprendizado de máquina geralmente enfatiza a precisão preditiva em vez da inferência orientada por hipóteses, focando geralmente em grandes conjuntos de dados de alta dimensionalidade (ou seja, com muitas covariáveis) (BI et al., 2019).

Nas últimas décadas, o aprendizado de máquina tem testemunhado avanços significativos, especialmente no campo do aprendizado profundo, onde redes neurais compostas por múltiplas camadas têm mostrado uma capacidade notável de aprender representações complexas de dados. Esse progresso tem beneficiado áreas como a visão computacional, o processamento de linguagem natural e os sistemas de recomendação, entre outros.

Paralelamente ao avanço técnico, há um crescente foco na interpretabilidade e na legibilidade dos modelos de aprendizado de máquina, especialmente à medida que eles são aplicados em domínios sensíveis como saúde, finanças e justiça. Questões éticas, como a mitigação de vieses nos modelos, têm se tornado centrais na pesquisa e na aplicação dessas tecnologias. Assim, o aprendizado de máquina não apenas continua a ser um componente vital da IA, mas também desempenha um papel cada vez mais crítico na transformação da tecnologia e da sociedade, com seu impacto esperado a crescer exponencialmente nos próximos anos.

2.3.2 Deep Learning

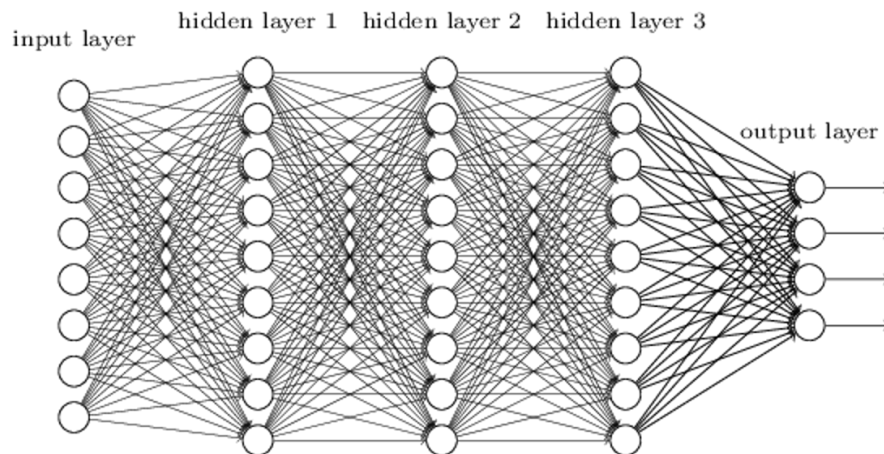
Deep Learning é um modelo de aprendizagem de máquina gerado pela sobreposição de módulos de representação não-lineares, cada um deles transformando a representação de dados dada a eles para uma representação mais abstrata, permitindo portanto o tratamento dos dados em vários níveis. Estes módulos de representação utilizam os próprios dados que eles processam para gerar seu próprio aprendizado e, portanto, a sua evolução (LECUN et al., 2015).

Esse subcampo do Machine Learning tem revolucionado diversas áreas, incluindo visão computacional, processamento de linguagem natural e reconhecimento de fala, devido à sua capacidade de aprender representações hierárquicas dos dados. Diferentemente dos métodos tradicionais de aprendizado de máquina, que dependem de extração manual de características, o aprendizado profundo permite que modelos aprendam diretamente dos dados brutos, extraíndo automaticamente características relevantes através de suas camadas.

As redes neurais profundas, que constituem a base do aprendizado profundo, são compostas por múltiplas camadas de neurônios artificiais, onde cada camada transforma

as saídas da camada anterior em representações mais abstratas e complexas. A primeira camada, chamada de camada de entrada, recebe os dados brutos, enquanto a última camada, conhecida como camada de saída, gera a previsão final. As camadas intermediárias, ou camadas ocultas, são responsáveis por aprender as representações intermediárias dos dados. A capacidade dessas redes de capturar relações complexas e não lineares entre os dados faz com que o aprendizado profundo seja particularmente eficaz em tarefas como classificação de imagens, tradução automática e diagnóstico médico.

Figura 2.5 – Representação esquemática de Redes Neurais.



Fonte: Autor desconhecido.⁴

O aprendizado profundo utiliza várias técnicas e arquiteturas para otimizar o desempenho das redes neurais. Redes convolucionais (Convolutional Neural Networks, CNNs), por exemplo, são amplamente utilizadas em tarefas de visão computacional devido à sua habilidade de capturar padrões espaciais em imagens. Já as redes neurais recorrentes (Recurrent Neural Networks, RNNs) e suas variantes, como as redes LSTM (Long Short-Term Memory), são particularmente eficazes no processamento de sequências temporais, sendo amplamente aplicadas em tarefas de processamento de linguagem natural e séries temporais.

Além das arquiteturas, técnicas como regularização, normalização de batch (batch normalization), dropout, e otimização de gradientes são empregadas para melhorar a robustez e a eficiência dos modelos de aprendizado profundo. O treinamento dessas redes requer grandes volumes de dados e poder computacional significativo, o que tem sido facilitado pelos avanços em hardware, como GPUs (unidades de processamento gráfico) e TPUs (unidades de processamento tensorial), bem como pelo acesso a grandes bases de dados e a plataformas de aprendizado distribuído.

⁴ Disponível em: <<https://sigmoidal.ai/redes-neurais-python-keras-2>>. Acesso em: 15 de setembro, 2024.

O estado da arte no aprendizado profundo continua a evoluir rapidamente, impulsionado por inovações em arquiteturas de redes neurais, algoritmos de otimização e técnicas de treinamento. Modelos como as redes GANs (Generative Adversarial Networks), que permitem a geração de dados sintéticos realistas, e os transformadores, que revolucionaram o processamento de linguagem natural com modelos como BERT e GPT, são exemplos de como o campo tem avançado em direções inesperadas e altamente impactantes.

2.4 DESENVOLVIMENTO WEB

O desenvolvimento web é um campo multifacetado que abrange a criação, construção e manutenção de websites e aplicações web, englobando tanto o front-end, que se refere à interface com a qual o usuário interage, quanto o back-end, que trata da lógica do servidor e da gestão de dados. Desde sua origem nos anos 1990, o desenvolvimento web tem evoluído rapidamente, acompanhando os avanços tecnológicos e as mudanças nas expectativas dos usuários. Esse campo se tornou fundamental para a comunicação, negócios, entretenimento e inúmeras outras áreas da vida moderna.

O serviço de front-end e o serviço de back-end são os dois serviços independentes que compõem a aplicação web. Através de chamadas de API, esses dois serviços podem se comunicar entre si. Enquanto o front-end da aplicação gerencia a interação do usuário, o back-end lida com a lógica de negócios e banco de dados. Há vários benefícios em ter um serviço distinto para o front-end e o back-end, incluindo desenvolvimento rápido, atualizações simples e alta capacidade de manutenção. No entanto, isso também apresenta desvantagens, como o risco de executar duas instâncias diferentes do mesmo programa simultaneamente, o que interfere no pipeline de integração e implantação contínua (DINKU, 2022).

O desenvolvimento web moderno é caracterizado pelo crescente uso de arquiteturas baseadas em microsserviços. Diferente da abordagem monolítica tradicional, onde todas as funcionalidades de uma aplicação são integradas em um único código-base, os microsserviços dividem a aplicação em serviços independentes, cada um executando uma função específica. Essa abordagem melhora a escalabilidade, facilita a manutenção e permite a implantação contínua, pois os desenvolvedores podem atualizar partes individuais da aplicação sem interromper todo o sistema.

Outro aspecto crucial do desenvolvimento web contemporâneo é a integração com a computação em nuvem. A computação em nuvem facilita a implementação de práticas DevOps, que integram desenvolvimento e operações em um ciclo contínuo de desenvolvimento, teste e implantação. Essa integração promove automação, reduz o tempo de lançamento no mercado e melhora a qualidade do software.

No âmbito dos bancos de dados, o desenvolvimento web tem visto uma diversificação significativa. Enquanto os bancos de dados relacionais continuam a ser amplamente utilizados, os bancos de dados NoSQL ganharam destaque devido à sua flexibilidade e capacidade de lidar com grandes volumes de dados não estruturados. Além disso, tecnologias como GraphQL estão mudando a forma como as APIs são desenvolvidas, permitindo consultas mais eficientes e personalizadas em comparação com as tradicionais APIs REST.

A segurança também se tornou uma preocupação central no desenvolvimento web. Com o aumento das ameaças cibernéticas, a proteção de dados sensíveis e a defesa contra ataques como Cross-Site Scripting (XSS), SQL Injection e Cross-Site Request Forgery (CSRF) tornaram-se fundamentais. Ferramentas e práticas de segurança, como HTTPS, Content Security Policy (CSP) e autenticação multifatorial, são agora integradas desde o início do processo de desenvolvimento.

2.4.1 Back-end

O desenvolvimento back-end é um componente crucial para o funcionamento de aplicações web e móveis, sendo responsável pela implementação da lógica de negócios, gerenciamento de bancos de dados, autenticação de usuários e integração de sistemas. Esta área tem evoluído de forma acelerada nos últimos anos, incorporando novas tecnologias, arquiteturas e práticas para responder às crescentes demandas por escalabilidade, segurança e desempenho.

Inicialmente, a arquitetura monolítica tradicional, onde todos os componentes de uma aplicação estão integrados em um único pacote, tem sido gradualmente substituída por arquiteturas baseadas em microserviços. Nesse novo modelo arquitetural, a aplicação é fragmentada em pequenos serviços independentes que se comunicam entre si, proporcionando maior flexibilidade, escalabilidade e facilidade de manutenção. Essa transição foi facilitada pelo surgimento de plataformas de orquestração como Kubernetes e Docker, que simplificam a implementação e gestão de microserviços.

Paralelamente, frameworks modernos como Node.js, Django, Ruby on Rails e Spring Boot dominam o cenário de desenvolvimento back-end, fornecendo ferramentas robustas que facilitam a construção de aplicações. Esses frameworks oferecem bibliotecas e estruturas pré-construídas que automatizam tarefas comuns, como autenticação de usuários, manipulação de dados e roteamento de requisições, simplificando assim o processo de desenvolvimento.

No que diz respeito aos bancos de dados, o desenvolvimento back-end tem se diversificado significativamente. Embora os bancos de dados relacionais, como MySQL,

PostgreSQL e Oracle, continuam a ser fundamentais, os bancos de dados NoSQL, como MongoDB, Cassandra e Redis, têm ganhado popularidade. Esses bancos de dados NoSQL são particularmente adequados para aplicações que lidam com grandes volumes de dados não estruturados ou semi-estruturados, oferecendo uma flexibilidade e escalabilidade que os bancos de dados relacionais tradicionais não conseguem alcançar facilmente. Além disso, o conceito de Data as a Service (DaaS) está emergindo, permitindo que organizações acessem, gerenciem e analisem dados em tempo real sem a necessidade de manter infraestruturas complexas.

A computação em nuvem tem transformado o desenvolvimento back-end ao permitir que as empresas implantem, escalem e gerenciem suas aplicações de maneira eficiente e com custos otimizados. Plataformas como AWS, Microsoft Azure e Google Cloud oferecem uma vasta gama de serviços gerenciados que suportam diversas necessidades, desde o armazenamento de dados até a inteligência artificial e o aprendizado de máquina. Um avanço significativo neste domínio é o modelo de computação 'serverless', onde os desenvolvedores escrevem código sem a preocupação com a infraestrutura subjacente. Serviços como AWS Lambda e Google Cloud Functions permitem que o código seja executado em resposta a eventos, escalando automaticamente conforme a demanda.

Adicionalmente, a cultura DevOps, que integra desenvolvimento e operações, tornou-se um componente essencial do desenvolvimento back-end moderno. Essa prática promove a automação de todo o ciclo de vida do desenvolvimento de software, desde a integração contínua (CI) até a entrega contínua (CD). Ferramentas como Jenkins, GitLab CI e CircleCI facilitam a integração, testes e implantação de código de forma automatizada, melhorando a eficiência e a qualidade do software.

A segurança também se tornou uma prioridade no desenvolvimento back-end, especialmente com o aumento das ameaças cibernéticas. A implementação de práticas como autenticação multifator (MFA), criptografia de dados e monitoramento contínuo é agora uma necessidade, e frameworks de segurança como OWASP são amplamente seguidos. Além disso, a conformidade com regulamentos como LGPD exige que as aplicações tratem os dados de usuários de forma segura e transparente.

2.4.2 Front-end

O desenvolvimento de front-end, que se refere à construção da interface de usuário (UI) e experiência de usuário (UX) em aplicações web, tem se consolidado como um campo fundamental na engenharia de software, evoluindo rapidamente para atender às demandas crescentes por interfaces interativas, responsivas e acessíveis. As tecnologias de front-end englobam uma ampla gama de ferramentas e frameworks que permitem aos

desenvolvedores criar aplicações web complexas, funcionais e esteticamente agradáveis.

O componente de um aplicativo ou site com o qual os usuários interagem é chamado de Front End ou Lado do Cliente da aplicação. Ele inclui tudo o que os usuários encontram diretamente, como imagens, botões, cores de texto, gráficos, tabelas, etc. HTML e CSS são usados para o design e a estilização, enquanto o JavaScript é utilizado para validações. Desempenho e responsividade são os dois principais objetivos do desenvolvimento de front-end (SHETTY et al., 2020).

Historicamente, as três principais tecnologias que formam a base do desenvolvimento de front-end são HTML, CSS e JavaScript. O HTML estrutura o conteúdo da web, o CSS é responsável pela apresentação visual, e o JavaScript adiciona interatividade e dinamismo às páginas. A partir dessas tecnologias fundamentais, uma série de bibliotecas e frameworks emergiram, permitindo o desenvolvimento mais eficiente e robusto.

Entre os frameworks de front-end mais proeminentes, destacam-se o React, o Angular e o Vue.js. O React, desenvolvido pelo Facebook, é uma biblioteca JavaScript focada na criação de interfaces de usuário por meio de componentes reutilizáveis. Ele introduziu o conceito de Virtual DOM, que melhora significativamente o desempenho das aplicações ao minimizar as operações de atualização da interface. O Angular, mantido pelo Google, é um framework completo que fornece uma solução integrada para o desenvolvimento de aplicações web, incluindo ferramentas para roteamento, gerenciamento de estado e integração com APIs. Já o Vue.js é uma framework progressiva, que pode ser adotada de forma incremental, proporcionando flexibilidade para os desenvolvedores.

A evolução das tecnologias e práticas ligadas ao desenvolvimento front-end é marcado pela adoção de práticas e ferramentas que visam melhorar a performance, escalabilidade e manutenibilidade das aplicações. O uso de pré-processadores CSS, como SASS e LESS, permite a criação de estilos de forma mais organizada e com maior capacidade de reutilização. Além disso, a utilização de ferramentas de build, como Webpack e Vite, otimizam o processo de desenvolvimento, oferecendo funcionalidades como carregamento rápido, divisão de código e integração com diversos plugins.

Outra tendência significativa é a ascensão de frameworks e ferramentas que permitem o desenvolvimento de aplicações de página única (Single Page Applications - SPAs) e aplicações progressivas na web (Progressive Web Applications - PWAs). As SPAs, construídas em frameworks como React e Angular, proporcionam uma experiência de usuário fluida, similar a uma aplicação desktop, ao carregar uma única página HTML e dinamicamente atualizar o conteúdo conforme necessário. As PWAs, por outro lado, combinam as melhores práticas da web e do mobile, permitindo que as aplicações web

funcionem offline, enviem notificações push e sejam instaladas como se fossem aplicativos nativos.

Além disso, o uso de design systems e bibliotecas de componentes tem se tornado cada vez mais comum. Essas ferramentas, como Material UI e Bootstrap, padronizam o desenvolvimento de interfaces, garantindo consistência e qualidade em grandes equipes de desenvolvimento.

No contexto atual, o front-end está cada vez mais integrado com tecnologias de backend através de arquiteturas como JAMstack (JavaScript, APIs, and Markup), que promovem a criação de aplicações rápidas e seguras, desacoplando o front-end do backend. O uso de APIs RESTful e GraphQL facilita a comunicação eficiente entre front-end e backend, permitindo o desenvolvimento de interfaces ricas e interativas com dados dinâmicos.

3 TECNOLOGIAS

Esta seção busca elucidar mais detalhes das tecnologias que poderiam e foram efetivamente utilizadas para o desenvolvimento e a hospedagem da plataforma de visualização, assim como os locais e métodos para armazenamento de dados que alimentam a plataforma.

3.1 AMAZON WEB SERVICES

A Amazon Web Services (AWS) é uma plataforma de computação em nuvem líder mundialmente, oferecida pela Amazon. Lançada em 2006, a AWS oferece uma ampla gama de serviços de computação em nuvem, incluindo armazenamento, banco de dados, análise, inteligência artificial, machine learning, Internet das Coisas (IoT), segurança, desenvolvimento de aplicativos, entre outros. A plataforma fornece uma infraestrutura global de data centers que permite que indivíduos e organizações executem suas cargas de trabalho com eficiência, escalabilidade e segurança, sem precisar gerenciar a infraestrutura física subjacente.

A AWS é amplamente reconhecida por seu modelo de computação em nuvem flexível e altamente escalável, que permite às empresas ajustar recursos computacionais conforme a demanda, eliminando a necessidade de investimento em hardware físico e reduzindo os custos operacionais. A plataforma oferece mais de 200 serviços, categorizados em áreas como computação, armazenamento, banco de dados, redes, análise de dados, aprendizado de máquina, segurança e conformidade. Entre os serviços mais destacados estão o Amazon Elastic Compute Cloud (EC2), que oferece capacidade computacional sob demanda, e o Amazon Simple Storage Service (S3), um serviço de armazenamento de objetos altamente durável e escalável.

A AWS disponibiliza os três modelos de serviço consagrados no mercado: Software as a Service (SaaS), Infrastructure as a Service (IaaS) e Plataforma as a Service (PaaS). Sua abrangência inclui uma ampla gama de serviços de computação, armazenamento, banco de dados, análises, aplicação e implantação, oferecidos como utilidades on-demand, acessíveis em questão de segundos, o que auxilia as organizações a aumentarem sua agilidade, reduzirem custos de TI e escalarem suas aplicações de maneira eficiente (SINHA, 2020).

Um dos avanços mais significativos da AWS é o conceito de computação 'serverless', exemplificado pelo serviço AWS Lambda. Com o Lambda, os desenvolvedores podem executar código em resposta a eventos sem a necessidade de provisionar ou gerenciar

servidores, permitindo que as aplicações escalem automaticamente em resposta às demandas de carga. Este modelo reduz a complexidade operacional e os custos, pois os desenvolvedores só pagam pelo tempo de execução do código.

Além disso, a AWS tem sido pioneira na oferta de serviços gerenciados de banco de dados, suportando tanto bancos de dados relacionais quanto não relacionais. O Amazon RDS (Relational Database Service) permite a fácil configuração, operação e escalabilidade de bancos de dados relacionais na nuvem, enquanto o Amazon DynamoDB oferece um serviço de banco de dados NoSQL totalmente gerenciado, conhecido por sua capacidade de lidar com grandes volumes de dados e tráfego intenso com baixa latência.

Para consolidar-se como um provedor global de nuvem pública, a AWS mantém uma rede mundial de infraestrutura que suporta e gerencia seu crescente portfólio de serviços de nuvem, atendendo clientes em escala global. A AWS é amplamente reconhecida por sua eficiência em termos de economia de tempo, flexibilidade, confiabilidade, segurança e custo-benefício. Este conjunto de atributos faz da AWS uma solução tecnológica robusta para a resolução de questões relacionadas à computação. Além disso, ao utilizar a AWS, as organizações não necessitam se preocupar com a manutenção de centros de dados, uma vez que toda a gestão é realizada pela própria AWS (SINHA, 2020).

A AWS também se destaca no campo de inteligência artificial e aprendizado de máquina, oferecendo serviços como Amazon SageMaker, que facilita a construção, treinamento e implantação de modelos de machine learning em escala. Além disso, a plataforma disponibiliza uma série de APIs de IA, incluindo reconhecimento de imagem, processamento de linguagem natural e previsão de séries temporais, que permitem aos desenvolvedores integrar facilmente capacidades de IA em suas aplicações.

Em termos de segurança e conformidade, a AWS adota uma abordagem rigorosa, com um modelo de responsabilidade compartilhada que assegura que tanto a infraestrutura da nuvem quanto as aplicações dos clientes estejam protegidas. A plataforma suporta mais de 90 padrões de conformidade e certificações de segurança, tornando-se uma escolha atraente para setores regulamentados, como finanças e saúde.

A AWS continua a inovar e expandir seu portfólio de serviços. A adoção de arquiteturas orientadas a eventos, a integração com tecnologias emergentes como blockchain e a oferta de ferramentas avançadas de automação e DevOps são exemplos de como a AWS está moldando o futuro da computação em nuvem. Além disso, a AWS está investindo significativamente em computação quântica com o Amazon Braket, que permite aos pesquisadores e desenvolvedores explorar algoritmos quânticos em um ambiente gerenciado.

3.1.1 Amazon EC2

O Amazon EC2 é um dos serviços centrais da AWS, desempenhando um papel crucial na popularização e consolidação da computação em nuvem. O serviço oferece capacidade de computação escalável na nuvem, permitindo que as organizações provisionem e gerenciem servidores virtuais, conhecidos como 'instâncias', de maneira flexível e sob demanda. Este serviço possibilita que empresas de todos os portes executem cargas de trabalho variadas, desde simples aplicativos web até complexas simulações científicas, sem a necessidade de investir em infraestrutura física.

O Amazon EC2 é projetado para oferecer uma ampla gama de opções de configuração, o que inclui diferentes tipos de instâncias, otimizadas para diversas necessidades de processamento, memória, armazenamento e rede. Isso permite que os usuários escolham a instância mais adequada para sua aplicação específica, otimizando o desempenho e os custos operacionais. As instâncias podem ser iniciadas e encerradas conforme necessário, proporcionando uma elasticidade que é fundamental para gerenciar variações na carga de trabalho.

Os clientes do Amazon EC2 alugam máquinas virtuais chamadas instâncias, de modo que cada instância possui um tipo que descreve seus recursos computacionais da seguinte forma: m1.small, m1.large e m1.xlarge denotam, respectivamente, instâncias 'padrão' pequenas, grandes e extra grandes; m2.xlarge, m2.2xlarge e m2.4xlarge denotam, respectivamente, instâncias 'de alta memória' extra grandes, duplamente extra grandes e quadruplicadamente extra grandes; e c1.medium e c1.xlarge denotam, respectivamente, instâncias 'de alta CPU' médias e extra grandes (BEN-YEHUDA et al., 2013).

Um dos aspectos mais inovadores do Amazon EC2 é sua capacidade de escalabilidade automática, por meio de serviços como o Auto Scaling e o Elastic Load Balancing. O Auto Scaling permite que as instâncias aumentem ou diminuam automaticamente em resposta à demanda, garantindo que as aplicações permaneçam disponíveis e eficientes em momentos de pico de uso, ao mesmo tempo em que ajudam a minimizar custos durante períodos de menor atividade. O Elastic Load Balancing, por sua vez, distribui automaticamente o tráfego de rede entre várias instâncias, assegurando a alta disponibilidade das aplicações.

Além disso, o Amazon EC2 oferece diversas opções de armazenamento, incluindo o Amazon Elastic Block Store (EBS) para armazenamento persistente em nível de bloco, e o Amazon Elastic File System (EFS) para armazenamento de arquivos, proporcionando soluções flexíveis e escaláveis para diferentes tipos de dados e cargas de trabalho. O EC2 também se integra de forma nativa com outros serviços AWS, como Amazon S3, para

armazenamento de objetos, e Amazon RDS, para bancos de dados gerenciados, formando uma infraestrutura coesa e interconectada.

A segurança é outro pilar fundamental do Amazon EC2. O serviço permite que os usuários configurem suas instâncias em Virtual Private Clouds (VPCs), onde podem definir controles detalhados de segurança, como grupos de segurança e listas de controle de acesso de rede (ACLs). Além disso, o EC2 suporta a criptografia de dados em repouso e em trânsito, garantindo que as informações sensíveis sejam protegidas em todas as fases de seu ciclo de vida.

No que diz respeito ao estado da arte, o Amazon EC2 continua a evoluir, introduzindo constantemente novos tipos de instâncias, baseadas em processadores otimizados, como as instâncias baseadas em Graviton, que utilizam arquitetura ARM para oferecer melhor desempenho por dólar para determinadas cargas de trabalho. A plataforma também tem expandido seu suporte a tecnologias emergentes, como machine learning e computação de alto desempenho (HPC), ao disponibilizar instâncias especializadas com GPUs e outros aceleradores de hardware.

3.1.2 Amazon S3

O Amazon S3 é um serviço de armazenamento de objetos amplamente utilizado e um dos pilares da AWS. O serviço foi projetado para fornecer armazenamento escalável, seguro e durável na nuvem, tornando-se a solução preferida para armazenar grandes volumes de dados, incluindo backups, arquivos estáticos, logs, e até mesmo arquivos multimídia como vídeos e imagens. Sua flexibilidade e robustez têm desempenhado um papel crucial na transformação digital de empresas de todos os tamanhos, ao permitir o armazenamento e a recuperação de dados de maneira eficiente e econômica.

O Amazon S3 foi intencionalmente projetado com um conjunto mínimo de funcionalidades para simplificar o gerenciamento de dados. As funcionalidades do S3 incluem operações de escrita, leitura e exclusão de objetos de dados, que são referenciados por uma chave exclusiva atribuída pelo desenvolvedor. Para assegurar a proteção dos dados contra acessos não autorizados, o S3 oferece um mecanismo de autenticação simples e eficaz. Além disso, os objetos armazenados podem ser configurados como privados ou públicos, e permissões de acesso específicas podem ser concedidas a usuários determinados. Dessa forma, um usuário do S3 tem a capacidade de autorizar outros usuários a acessarem seus dados, garantindo flexibilidade e controle sobre o compartilhamento de informações (GARFINKEL, 2007).

O Amazon S3 organiza os dados em "buckets," que são contêineres globais onde os

objetos, ou arquivos, são armazenados. Cada objeto no S3 é identificado por uma chave única dentro de um bucket, permitindo o armazenamento de quantidades praticamente ilimitadas de dados. A arquitetura de armazenamento de objetos do S3 é projetada para ser altamente escalável, o que significa que os usuários podem facilmente expandir seu armazenamento sem se preocupar com a capacidade ou com a complexidade da gestão dos dados.

Uma das características mais notáveis do S3 é sua durabilidade e disponibilidade. O serviço garante uma durabilidade de praticamente 100% para os objetos armazenados, o que é alcançado por meio da replicação automática de dados em várias regiões geográficas, protegendo contra falhas e desastres regionais. Além disso, o S3 oferece diferentes classes de armazenamento, como o S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA (Infrequent Access) e S3 Glacier, que permitem aos usuários otimizar os custos de armazenamento com base na frequência e no tempo de recuperação dos dados.

O Amazon S3 também se destaca em termos de segurança e conformidade, oferecendo uma ampla gama de ferramentas e controles que permitem aos usuários proteger seus dados de maneira eficaz. O S3 suporta criptografia de dados tanto em repouso quanto em trânsito, através de chaves gerenciadas pelo AWS Key Management Service (KMS) ou chaves fornecidas pelo próprio cliente. Além disso, o S3 integra-se com o AWS Identity and Access Management (IAM), permitindo a implementação de políticas de acesso detalhadas que controlam quem pode acessar os dados e em quais condições.

A escalabilidade e a eficiência do S3 também são aprimoradas pela integração nativa com outros serviços AWS, como o Amazon CloudFront para entrega de conteúdo, o AWS Lambda para processamento de dados em resposta a eventos, e o Amazon Athena para análise de dados diretamente no S3 usando SQL. Essa integração permite que o S3 funcione como uma base centralizada para uma ampla gama de aplicações, desde simples armazenamento de arquivos até arquiteturas de big data complexas.

No contexto do estado da arte, o Amazon S3 continua a evoluir para atender às necessidades crescentes de armazenamento e processamento de dados em grandes volumes. A AWS tem introduzido constantemente novos recursos que aprimoram a funcionalidade do S3, como o S3 Select, que permite a extração de subconjuntos de dados diretamente de um objeto, reduzindo a quantidade de dados transferidos e os custos associados. Além disso, o S3 Object Lock oferece proteção imutável para os dados, o que é essencial para atender a requisitos regulatórios e de conformidade.

3.1.3 Amazon SageMaker

O SageMaker é uma plataforma de aprendizado de máquina fornecida como parte da AWS e oferece suporte a treinamento incremental, aprendizado elástico e retomável, bem como otimização automática de hiperparâmetros (LIBERTY et al., 2020). Lançado em 2017, o SageMaker visa democratizar o acesso ao machine learning, permitindo que desenvolvedores e cientistas de dados de todos os níveis de experiência construam modelos preditivos de maneira eficiente e sem a necessidade de gerenciar a infraestrutura subjacente.

O SageMaker oferece um ambiente integrado que simplifica todas as etapas do ciclo de vida do machine learning. Isso inclui a preparação dos dados, a seleção e construção de algoritmos, o treinamento de modelos em grande escala, a otimização de hiperparâmetros, e a implantação dos modelos em produção. A plataforma suporta uma ampla gama de algoritmos prontos para uso, bem como a capacidade de trazer seus próprios modelos personalizados ou scripts de treinamento, proporcionando flexibilidade tanto para iniciantes quanto para usuários avançados.

Uma das principais características do Amazon SageMaker é a sua capacidade de treinamento distribuído, que permite o processamento eficiente de grandes volumes de dados. O SageMaker pode automaticamente distribuir o treinamento de modelos por múltiplas instâncias, reduzindo significativamente o tempo necessário para construir modelos complexos. Além disso, o serviço inclui ferramentas para otimização automática de hiperparâmetros (Amazon SageMaker Automatic Model Tuning), que ajusta os parâmetros de entrada para melhorar o desempenho do modelo de maneira automatizada.

O SageMaker também facilita a implementação de modelos em produção com seu recurso de implantação automatizada e escalável. Os modelos treinados podem ser facilmente implantados como endpoints hospedados em AWS, prontos para inferência em tempo real ou em lote. Esse processo é gerenciado pelo SageMaker, que cuida da escalabilidade, balanceamento de carga, e monitoramento dos endpoints, garantindo que os modelos funcionem de maneira eficiente e contínua.

Além do treinamento e implantação, o Amazon SageMaker inclui ferramentas avançadas de monitoramento e explicabilidade, como o Amazon SageMaker Model Monitor, que permite a detecção automática de desvios de dados que podem afetar a precisão do modelo, e o Amazon SageMaker Clarify, que ajuda a detectar e mitigar vieses nos dados e nos modelos. Essas ferramentas são essenciais para manter a integridade e a confiabilidade dos modelos em ambientes de produção, especialmente em setores onde a transparência e a conformidade são críticas.

O Amazon SageMaker se destaca por sua integração com outros serviços AWS,

como o Amazon S3 para armazenamento de dados, o AWS Glue para preparação de dados, e o Amazon Athena para consultas SQL em grande escala. Essa integração permite que os usuários construam pipelines de machine learning completos e automatizados, que vão desde a ingestão e processamento de dados até a implantação e monitoramento contínuo de modelos.

3.2 GOOGLE EARTH ENGINE

O Google Earth Engine (GEE) é uma plataforma poderosa de computação em nuvem desenvolvida pelo Google, destinada à análise e visualização de grandes quantidades de dados geoespaciais. Lançada em 2010, a ferramenta oferece uma vasta gama de funcionalidades voltadas para o processamento de imagens de satélite, dados geográficos, e análise ambiental, sendo amplamente utilizada por pesquisadores, cientistas, ONGs e órgãos governamentais em todo o mundo.

O principal diferencial do Google Earth Engine é a capacidade de lidar com grandes volumes de dados, particularmente séries temporais de imagens de satélite, de forma eficiente e escalável. O sistema armazena petabytes de dados geoespaciais, incluindo imagens de satélite de diferentes sensores e épocas, o que permite aos usuários acessar informações que abrangem décadas de observação da Terra. Entre as fontes de dados mais importantes estão imagens dos satélites Landsat e Sentinel, além de dados de topografia, uso do solo e clima.

O catálogo de dados abriga um grande repositório de conjuntos de dados geoespaciais disponíveis publicamente, incluindo observações de uma variedade de sistemas de imagens de satélite e aéreas em comprimentos de onda ópticos e não ópticos, variáveis ambientais, previsões e retroprojeções climáticas e meteorológicas, cobertura do solo, dados topográficos e socioeconômicos. Todos esses dados são pré-processados em uma forma pronta para uso, mas que preserva as informações, permitindo um acesso eficiente e removendo muitas barreiras associadas ao gerenciamento de dados (GORELICK et al., 2017).

Uma das características mais relevantes do GEE é o processamento paralelo de dados, que ocorre em sua infraestrutura de computação em nuvem. Isso permite que análises complexas, que exigiriam recursos computacionais elevados em sistemas locais, sejam realizadas de maneira rápida e eficiente. O GEE disponibiliza uma interface de programação (API) baseada em JavaScript e Python, facilitando a criação de scripts para análise de dados, além de oferecer uma interface gráfica que simplifica o uso da ferramenta para usuários com menos experiência em programação.

As aplicações do Google Earth Engine são vastas e abrangem diversas áreas do conhecimento. Ele é amplamente utilizado em estudos de monitoramento ambiental, como desmatamento, mudanças no uso da terra, monitoramento de corpos d'água, análise de emissões de carbono e desastres naturais. Por exemplo, é uma ferramenta essencial para o mapeamento e monitoramento da Amazônia, fornecendo dados em tempo quase real para a detecção de áreas desmatadas ou queimadas. Além disso, a plataforma tem sido utilizada em projetos de sustentabilidade urbana, como a análise de expansão urbana e os seus impactos sobre os ecossistemas locais.

Do ponto de vista técnico, o Google Earth Engine oferece uma variedade de algoritmos prontos para uso, como classificadores de imagens, índices de vegetação, algoritmos de detecção de mudanças e modelos hidrológicos. Estes podem ser aplicados diretamente sobre os dados de entrada, permitindo aos usuários obter rapidamente resultados significativos. Ademais, a integração com outras ferramentas e bibliotecas de código aberto, como o TensorFlow, expande ainda mais as possibilidades analíticas da plataforma.

Em termos de acesso e colaboração, o Google Earth Engine é gratuito para pesquisadores, instituições sem fins lucrativos e o setor público, o que promove uma ampla adoção da ferramenta em projetos de relevância social e científica. A possibilidade de compartilhar scripts e resultados dentro da comunidade também é um fator que fortalece a colaboração entre diferentes usuários e instituições ao redor do mundo.

3.3 PYHTON

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral, amplamente reconhecida por sua simplicidade sintática e versatilidade. Criada por Guido van Rossum e lançada pela primeira vez em 1991, Python foi projetada com a filosofia de facilitar a legibilidade do código, utilizando uma sintaxe que permite aos programadores expressar conceitos de maneira concisa e clara. Essa característica faz de Python uma linguagem ideal tanto para iniciantes quanto para desenvolvedores experientes, favorecendo a manutenção e a colaboração em projetos de software.

Uma das principais vantagens de Python é sua ampla adoção em diversos domínios da computação, como desenvolvimento web, ciência de dados, automação, inteligência artificial e machine learning, análise de dados, e programação de sistemas. Python possui uma biblioteca padrão extensa que oferece módulos e pacotes para uma grande variedade de tarefas, desde manipulação de strings e operações matemáticas até acesso ao sistema de arquivos e protocolos de rede. Além disso, sua comunidade ativa e expansiva contribui regularmente com bibliotecas adicionais, como NumPy, Pandas, Matplotlib, TensorFlow, e Django, que ampliam as capacidades da linguagem para aplicações específicas.

A linguagem é interpretada, o que significa que o código Python é executado linha por linha, sem a necessidade de compilação prévia. Isso acelera o ciclo de desenvolvimento, pois permite testes e iterações rápidas. Python é também dinamicamente tipada, o que significa que os tipos de dados das variáveis são determinados em tempo de execução, proporcionando flexibilidade, embora essa característica possa introduzir desafios de desempenho em comparação com linguagens compiladas e estaticamente tipadas.

Python é altamente portátil, suportado em uma ampla variedade de plataformas, incluindo Windows, macOS e várias distribuições de Linux, bem como em dispositivos móveis e ambientes embarcados. Essa portabilidade, aliada à sua sintaxe clara e rica biblioteca padrão, faz de Python uma escolha popular para desenvolvimento de scripts, automação de tarefas, e prototipagem rápida, além de ser amplamente utilizada na educação como linguagem introdutória de programação.

Devido à sua arquitetura orientada a objetos, o Python deixou de ser apenas uma linguagem de scripting. A linguagem evoluiu além do scripting e avançou para as áreas de desenvolvimento web e big data. Suas vantagens incluem uma abundância de bibliotecas e ferramentas, uma comunidade ampla e ativa, e documentação clara. Adicionalmente, pelo fato de ser escrito em C, aplicações que demandam alto desempenho podem aproveitar a estrutura de dados e funcionalidades centrais do C, aumentando assim a eficiência (GHIMIRE, 2020).

No desenvolvimento web, frameworks como Django e Flask permitiram que Python se destacasse na construção de aplicações web robustas e escaláveis. Django, por exemplo, oferece uma estrutura de desenvolvimento full-stack, que facilita a construção de aplicações complexas com segurança e eficiência. No domínio da ciência de dados e machine learning, Python tornou-se a linguagem de escolha, graças a bibliotecas como NumPy, Pandas, Scikit-learn, TensorFlow e PyTorch, que fornecem ferramentas poderosas para análise de dados, modelagem estatística, e treinamento de modelos de aprendizado de máquina.

Python também desempenha um papel fundamental na automação e scripting, permitindo aos desenvolvedores criar scripts simples para automatizar tarefas repetitivas, bem como construir pipelines de automação mais complexos em ambientes corporativos. Ferramentas como Ansible e frameworks como Selenium utilizam Python para automação de infraestrutura e testes automatizados, respectivamente.

3.3.1 Django

Django é um framework de desenvolvimento web de alto nível, escrito em Python, que facilita a criação de aplicações web robustas, seguras e escaláveis. O framework foi

projetado com a filosofia de 'não repetir a si mesmo' (DRY - Don't Repeat Yourself), enfatizando a reutilização de código, a modularidade e a redução da redundância no desenvolvimento. Este framework oferece uma abordagem estruturada para a construção de aplicações web, permitindo que os desenvolvedores foquem em escrever código limpo e eficiente, enquanto Django cuida de aspectos fundamentais, como a segurança, a autenticação de usuários, e a administração de banco de dados.

Uma das principais características do Django é seu ORM (Object-Relational Mapping), que permite que os desenvolvedores interajam com bancos de dados relacionais de maneira intuitiva, usando objetos Python ao invés de escrever SQL diretamente. O ORM do Django suporta várias bases de dados, como PostgreSQL, MySQL, SQLite e Oracle, e permite a migração fácil entre elas, proporcionando uma camada de abstração que simplifica a gestão e a manipulação de dados. Essa funcionalidade não apenas acelera o desenvolvimento, mas também promove a portabilidade e a manutenção do código.

Django é conhecido por seu foco em segurança, integrando medidas robustas para proteger as aplicações web contra ameaças comuns, como SQL injection, XSS, e CRFS. Além disso, o framework facilita a implementação de práticas de segurança, como o uso de senhas criptografadas, a configuração de políticas de autenticação e autorização de usuários, e a proteção contra ataques de clickjacking. Essa ênfase na segurança torna Django uma escolha popular para a construção de aplicações web que lidam com dados sensíveis, como em setores financeiros e de saúde.

O framework segue o padrão de arquitetura Model-View-Template (MVT), que é uma variante do Model-View-Controller (MVC). Nessa arquitetura, o "Model" representa a estrutura e a lógica de negócios dos dados, o "View" é responsável pela apresentação da informação ao usuário, e o "Template" é a camada que define a interface de usuário através de uma linguagem de template baseada em HTML. Esse design modular permite um desenvolvimento mais organizado e facilita a manutenção e a evolução das aplicações ao longo do tempo.

Outra característica notável do Django é seu sistema de administração automático, gerado a partir dos modelos de dados definidos pelo desenvolvedor. Esse sistema de administração oferece uma interface web pronta para uso que permite a manipulação e gestão dos dados da aplicação sem a necessidade de código adicional, o que é particularmente útil durante o desenvolvimento e para a administração de conteúdo em produção.

Django também se destaca por sua escalabilidade, sendo utilizado para construir desde pequenos sites até grandes plataformas web com milhões de usuários. A flexibilidade do framework permite que ele se adapte a diferentes cargas de trabalho, suportando

o crescimento das aplicações ao longo do tempo. O uso de componentes reutilizáveis, middlewares e a possibilidade de integração com outras tecnologias e serviços, como caches de memória (e.g., Memcached, Redis) e serviços de nuvem, aumentam ainda mais sua capacidade de lidar com aplicações em grande escala.

No contexto do desenvolvimento web moderno, Django continua a ser um dos frameworks mais populares e amplamente adotados, sustentado por uma comunidade ativa e uma rica documentação. A comunidade Django contribui regularmente com pacotes e extensões que expandem as capacidades do framework, facilitando a integração com novas tecnologias e melhorando a eficiência do desenvolvimento.

3.4 HTML

O HyperText Markup Language (HTML) é a linguagem padrão utilizada na criação de documentos que são exibidos em navegadores web. Desde sua criação em 1991 por Tim Berners-Lee, o HTML passou por diversas evoluções, consolidando-se como a base da estruturação de conteúdo na internet. Sua principal função é organizar e estruturar informações textuais e multimídias, permitindo que sejam interpretadas de forma consistente por diferentes navegadores.

O HTML oferece suporte a uma ampla variedade de conteúdos, incluindo textos, planilhas, cliques de vídeo, animações e imagens. A página da web, frequentemente referida como "documento" em termos técnicos, tem seu tipo de documento definido no início do código, o que determina a natureza do documento e como ele deve ser interpretado pelo navegador durante a renderização. Como componente essencial da World Wide Web, o HTML passou por diversas fases de desenvolvimento desde 1991 até o presente, com a versão inicial sendo o HTML e a versão mais recente sendo o HTML5 (GHIMIRE, 2020)

A arquitetura do HTML é baseada em uma série de elementos e tags, que descrevem diferentes partes de um documento, como títulos, parágrafos, links, imagens, tabelas e listas. Cada elemento é definido por uma tag que pode conter atributos adicionais para especificar o comportamento ou aparência deste conteúdo. Um aspecto crucial do HTML é sua semântica, que foi amplamente aprimorada com a introdução do HTML5 em 2014. Essa versão trouxe novas tags semânticas, como `<header>`, `<footer>`, `<article>`, e `<section>`, que permitem uma descrição mais precisa da estrutura e do propósito do conteúdo, melhorando tanto a acessibilidade quanto o SEO (Search Engine Optimization).

A expansão do HTML está intrinsecamente ligado à contínua evolução dos padrões web e ao desenvolvimento colaborativo de organizações como o World Wide Web Consortium (W3C) e a Web Hypertext Application Technology Working Group (WHATWG). Es-

As organizações trabalham em conjunto para manter e evoluir o padrão HTML, adaptando-o às novas necessidades da web moderna, como o suporte a novos formatos de mídia, melhor desempenho em dispositivos móveis e integração com APIs avançadas para aplicações web interativas.

Além disso, o HTML5 introduziu APIs nativas que expandem as capacidades das páginas web, como a API de Geolocalização, a API de Armazenamento Local e a API de Multimídia, que permitem o desenvolvimento de aplicações web ricas sem a necessidade de plugins externos. O HTML5 também melhorou o suporte para gráficos e animações por meio da integração com a API `<canvas>`, que possibilita o desenho de gráficos dinâmicos diretamente no navegador.

Nesse contexto, a contínua introdução de novos padrões e a evolução dos navegadores web garantem que o HTML permaneça relevante, eficiente e capaz de suportar as inovações no desenvolvimento web. A tendência é que o HTML continue a ser um pilar central na construção de experiências digitais, com futuras evoluções focadas na simplificação do desenvolvimento, melhoria da acessibilidade, e integração mais profunda com outras tecnologias emergentes da web, como WebAssembly e Web Components.

3.5 CSS

O Cascading Style Sheets (CSS) é uma linguagem de estilo utilizada na web para definir a apresentação visual de documentos escritos em HTML ou XML. Desde sua introdução em meados da década de 1990, o CSS tornou-se uma ferramenta fundamental para o desenvolvimento web, permitindo separar o conteúdo de um documento de sua apresentação, o que facilita a manutenção, reutilização e personalização de interfaces.

O CSS atua como uma camada de design que especifica como os elementos de uma página devem ser exibidos em termos de layout, cores, fontes, espaçamento, e outros aspectos visuais. Ao trabalhar em conjunto com HTML, o CSS permite criar layouts responsivos e atraentes, capazes de se adaptar a diferentes dispositivos e resoluções de tela, melhorando significativamente a experiência do usuário.

A arquitetura do CSS é baseada em seletores, que identificam os elementos do documento a serem estilizados, e declarações, que especificam os estilos aplicáveis. O CSS define três possíveis fontes para folhas de estilo: autores, leitores e navegadores, sendo capaz de combinar folhas de estilo dessas fontes para formar a apresentação de um documento. O processo de combinar várias folhas de estilo – e resolver conflitos caso ocorram, com base em critérios como especificidade, importância e origem das regras – é conhecido como cascata, característica central da linguagem (LIE, 2006).

O CSS está submetido a um processo de evolução contínua, com a introdução de novos módulos e funcionalidades que expandem suas capacidades. Entre os avanços mais significativos, destaca-se o CSS Grid Layout e o Flexbox, que proporcionam métodos modernos e flexíveis para a criação de layouts complexos e responsivos sem a necessidade de hacks ou soluções alternativas que eram comuns em versões anteriores do CSS. Essas tecnologias permitem um controle preciso sobre o posicionamento, alinhamento e dimensionamento de elementos dentro de uma página.

Outra inovação importante é a introdução de variáveis CSS (ou custom properties), que permitem a reutilização de valores em diferentes partes de uma folha de estilo, facilitando a manutenção e atualização de temas. As variáveis, juntamente com novos recursos como media queries e funções de cálculo, permitem que os desenvolvedores criem interfaces dinâmicas que se adaptam automaticamente a diferentes condições, como tamanhos de tela, preferências de esquema de cores, e outros fatores contextuais.

O CSS também evoluiu para melhorar o suporte à acessibilidade e à internacionalização, com a introdução de novas propriedades que facilitam a criação de interfaces inclusivas para usuários com diferentes necessidades e preferências culturais. Além disso, o CSS moderno incorpora recursos avançados de animação e transição, permitindo a criação de efeitos visuais sofisticados sem a necessidade de JavaScript adicional.

No contexto atual, o CSS continua a ser uma linguagem essencial para o desenvolvimento web, com uma comunidade ativa de desenvolvedores e a constante atualização dos padrões pelo World Wide Web Consortium (W3C). A introdução de novas funcionalidades, como o CSS Houdini, que oferece APIs para estender o comportamento do CSS além de suas capacidades nativas, representa a vanguarda da inovação na linguagem. Esses avanços indicam uma tendência crescente de maior controle e personalização na criação de interfaces, bem como uma maior integração entre o CSS e outras tecnologias da web.

3.6 JAVASCRIPT

Criado originalmente por Brendan Eich em 1995, o JavaScript se consolidou como uma das principais linguagens de programação na web, reconhecida por sua versatilidade e capacidade de adicionar interatividade e dinamismo às páginas web. Esta linguagem de programação do lado do cliente permite a manipulação dinâmica de elementos HTML, a resposta a eventos do usuário e a comunicação assíncrona com servidores, enriquecendo a experiência do usuário e possibilitando a criação de aplicações web altamente interativas.

Através do JavaScript, um script simples pode ser inserido inline no documento HTML ou em um arquivo separado, vinculado por meio de uma tag no cabeçalho do

arquivo. Dessa forma, o documento HTML identifica qual script deve ser carregado e de qual local. A principal característica do JavaScript é a capacidade de carregar ou atualizar o conteúdo da página sem recarregar toda a página, direcionando apenas a tag relevante. Além disso, permite a adição dinâmica de estilos CSS conforme necessário (GHIMIRE, 2020).

O JavaScript é uma linguagem de alto nível, interpretada, e baseada em eventos, o que a torna ideal para manipulação do Document Object Model (DOM), validação de formulários, e criação de animações interativas. Seu papel expandiu-se significativamente com o tempo, especialmente com o advento do AJAX (Asynchronous JavaScript and XML) no início dos anos 2000, que possibilitou o desenvolvimento de aplicações de página única (Single Page Applications - SPAs) e a interação assíncrona com servidores sem a necessidade de recarregar a página. Isso marcou o início de uma nova era de interfaces mais rápidas e responsivas.

O progresso do JavaScript reflete uma trajetória contínua, impulsionada por uma comunidade ativa e por inovações tecnológicas. Um marco significativo nesse desenvolvimento foi a introdução do ECMAScript 6 (ES6) em 2015, que trouxe várias melhorias à linguagem, como a inclusão de classes, módulos, arrow functions e promessas, que facilitaram a escrita de código mais limpo, modular e eficiente. Essas funcionalidades transformaram o JavaScript em uma linguagem mais poderosa e expressiva, adequada para o desenvolvimento tanto de front-end quanto de back-end, especialmente com a popularização do Node.js, que permite a execução de JavaScript no servidor.

No front-end, bibliotecas e frameworks como React, Angular, e Vue.js revolucionaram a forma como aplicações web são construídas. O React, criado pelo Facebook, popularizou o conceito de componentes reutilizáveis e a manipulação eficiente do DOM através do Virtual DOM. O Angular, mantido pelo Google, oferece uma estrutura robusta para a criação de aplicações escaláveis, com uma ênfase na integração e na arquitetura MVVM (Model-View-ViewModel). O Vue.js, por sua vez, combina as melhores características de ambos, oferecendo uma curva de aprendizado suave e flexibilidade para se adaptar a diferentes projetos.

O JavaScript moderno também tem se beneficiado de ferramentas de build como Webpack e Babel, que permitem a transpilação de código e a otimização de desempenho, e de gerenciadores de pacotes como npm e Yarn, que facilitam a gestão de dependências. Além disso, o TypeScript, um superset do JavaScript desenvolvido pela Microsoft, introduziu tipagem estática à linguagem, melhorando a detecção de erros em tempo de desenvolvimento e aumentando a robustez do código.

Por outro lado, as inovações no JavaScript não se limitam ao front-end. No back-end, o Node.js expandiu o uso da linguagem, permitindo que os desenvolvedores utilizem JavaScript em toda a stack de desenvolvimento, facilitando a criação de aplicações web de alta performance, como APIs RESTful e serviços de microserviços. O estado da arte inclui também o uso crescente de arquiteturas orientadas a eventos e programação reativa, exemplificadas por frameworks como RxJS, que permitem o gerenciamento eficiente de fluxos de dados assíncronos.

Além disso, o JavaScript tem desempenhado um papel crescente no desenvolvimento de aplicações móveis híbridas e desktop, através de frameworks como React Native e Electron, respectivamente. Essas tecnologias permitem que os desenvolvedores utilizem um único código base em JavaScript para criar aplicações que funcionam em múltiplas plataformas, reduzindo o tempo e os custos de desenvolvimento.

Em termos de futuro, o JavaScript continua a evoluir, com o ECMA International lançando atualizações anuais que introduzem novas funcionalidades e melhorias de desempenho. A linguagem está cada vez mais integrada com outras tecnologias emergentes, como WebAssembly, que permite a execução de código de baixo nível no navegador com alto desempenho, e o desenvolvimento de Web Components, que promovem a criação de elementos de interface reutilizáveis nativamente suportados pelos navegadores.

4 TRABALHOS RELACIONADOS

4.1 GIUZIO ET AL. (2023)

O projeto de formatura intitulado “Estimation of Socioeconomic Indicators through Satellite Imagery in the Nexus Area” tem como objetivo principal propor uma metodologia para a estimativa de indicadores socioeconômicos no território brasileiro, especificamente na região Nexus, utilizando imagens de satélite e técnicas de aprendizado profundo.

Dado o vasto território brasileiro e a dificuldade de realizar censos frequentes e abrangentes, o trabalho propõe o uso de imagens de satélite como uma alternativa para estimar indicadores socioeconômicos. Embora o IBGE forneça dados censitários, a periodicidade desses censos (a cada 10 anos) e os desafios de acessibilidade em algumas regiões tornam difícil obter uma visão em tempo real da situação socioeconômica. O uso de imagens de satélite e aprendizado de máquina possibilita estimar esses indicadores com maior frequência e menor custo.

O trabalho tem como os seus dois principais objetivos (i) desenvolver uma metodologia reproduzível para estimar indicadores socioeconômicos na área Nexus usando aprendizado profundo, através de modelos de Deep Learning com imagens multiespectrais e de luz noturna; e (ii) propor e desenvolver uma plataforma interativa que permita visualizar e explorar os indicadores estimados para diferentes regiões e períodos de tempo.

A metodologia adotada foi dividida em duas etapas principais: a aquisição de dados e a estimativa dos indicadores. Ao longo da etapa de aquisição de dados, foram utilizados dois tipos principais de dados: (i) indicadores socioeconômicos: Mais de 100 indicadores fornecidos pelo INPE e IBGE, que incluem dados sobre renda, alfabetização e longevidade, baseados no censo de 2010; e (ii) imagens de satélite: Imagens multiespectrais e de luz noturna coletadas através da plataforma Google Earth Engine. Os dados foram organizados de maneira a permitir a segmentação da área de estudo em clusters, que foram posteriormente utilizados no treinamento dos modelos de aprendizado profundo.

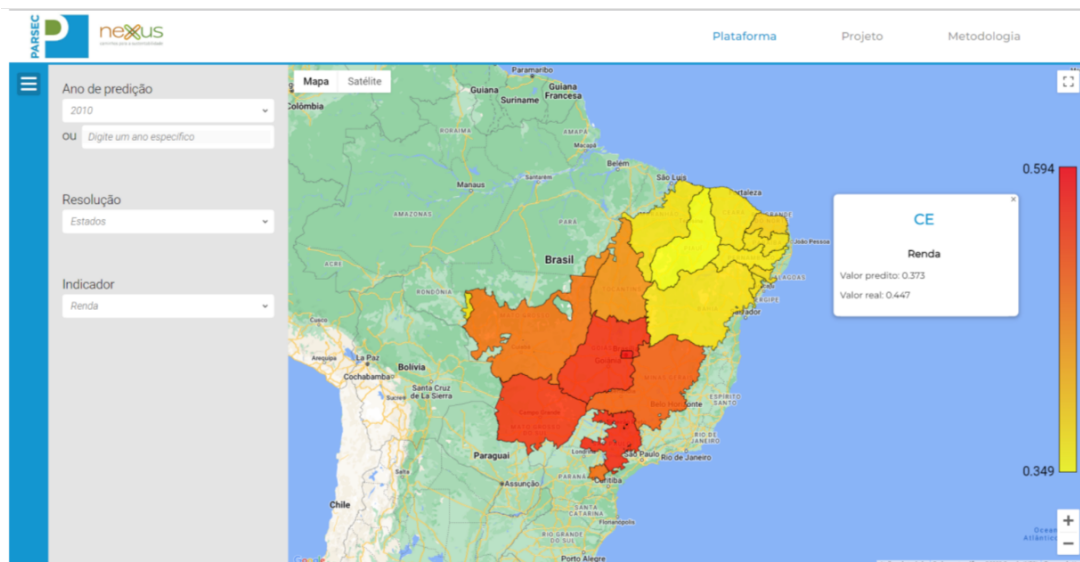
Já na etapa de estimativa de indicadores socioeconômicos, foram treinados modelos de Deep Learning baseados na arquitetura ResNet-18 para estimar os indicadores a partir das imagens de satélite. Modelos separados foram treinados para cada tipo de dado: imagens multiespectrais e imagens de luz noturna. Além disso, foi treinado um modelo combinado, utilizando ambos os tipos de imagens para melhorar a acurácia das estimativas.

Os resultados obtidos indicam que as imagens multiespectrais foram mais eficazes

na previsão dos indicadores do que as imagens de luz noturna, devido a limitações de dados nas imagens noturnas. O melhor modelo alcançou uma correlação de 42% entre os indicadores preditos e os valores reais, um resultado considerado promissor em comparação ao trabalho de referência (70%), consideradno ainda que os modelos podem ser melhorados como proposto ao longo do trabalho.

O trabalho conseguiu criar uma metodologia documentada para a estimativa de indicadores socioeconômicos a partir de imagens de satélite, aplicável ao cenário brasileiro. A plataforma interativa proposta permite que os resultados sejam visualizados em um mapa do Brasil, facilitando a exploração dos dados para diferentes regiões e anos. Além disso, o trabalho contribuiu com a criação de um novo conjunto de dados, resultado da fusão dos indicadores socioeconômicos com as informações geográficas e administrativas dos setores censitários brasileiros.

Figura 4.1 – Plataforma de visualização de desenvolvida por GIUZIO et al. (2022).



Fonte: GIUZIO et al. (2022).

4.2 MEHMOOD ET AL. (2021)

O artigo intitulado “Mapping of Flood Areas Using Landsat with Google Earth Engine Cloud Platform” publicado na revista Atmosphere explora a utilização de imagens Landsat e a plataforma Google Earth Engine (GEE) para o mapeamento de áreas de inundação. Esta pesquisa visa preencher lacunas de dados, especialmente em países em desenvolvimento, onde as informações sobre riscos de inundação são escassas ou desatualizadas.

Os Desastres Relacionados à Água (WRD), como inundações, são responsáveis por cerca de 90% dos desastres naturais globais. Desde 2000, mais de 5300 WRD foram

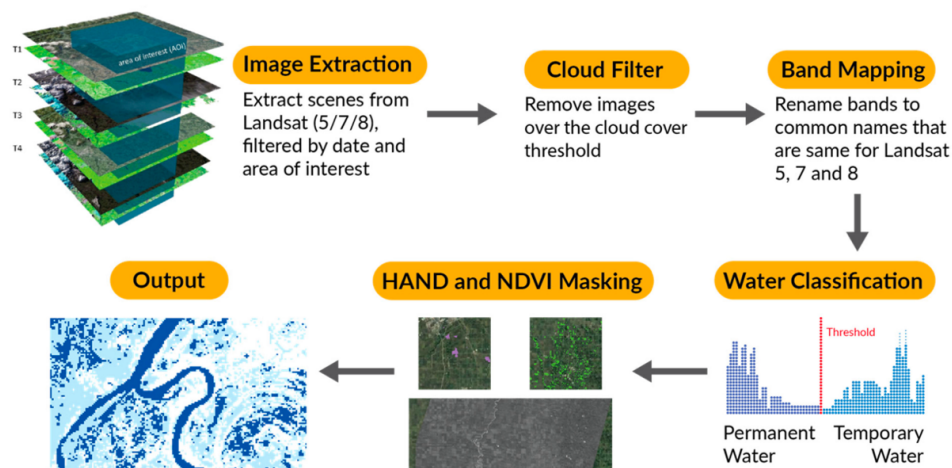
registrados, resultando em mais de 325 mil mortes e perdas econômicas superiores a US\$ 1,7 trilhões. As inundações representam 54% desses desastres. O estudo sublinha a importância de prever a extensão das inundações para mitigar os impactos socioeconômicos, uma vez que essas catástrofes têm crescido em frequência e intensidade devido às mudanças climáticas.

A modelagem de inundações com precisão geralmente requer modelos complexos, que são custosos em termos de tempo e recursos. Em países como o Canadá, a atualização de mapas nacionais de inundações pode custar US\$ 350 milhões e levar uma década. A situação é ainda mais grave em países em desenvolvimento, onde mapas de risco de inundação são muitas vezes inexistentes ou inadequados.

O artigo apresenta o desenvolvimento de um Algoritmo de Mapeamento de Inundações (Flood Mapping Algorithm - FMA), que utiliza dados das imagens de satélite Landsat 5, 7 e 8, processadas na plataforma GEE. O FMA se baseia na criação de um “data cube”, composto por uma pilha de pixels sobrepostos de imagens Landsat capturadas ao longo do tempo. Esse cubo de dados permite a identificação de corpos d’água temporários e permanentes usando o Índice Modificado de Diferença Normalizada da Água (MNDWI), combinado com dados de elevação e uso do solo.

O algoritmo segue cinco etapas principais: (1) extração de imagens; (2) filtragem de nuvens; (3) mapeamento de bandas; (4) classificação da água; e (5) mascaramento com dados HAND (Altura Acima da Drenagem mais Próxima) e NDVI (Índice de Vegetação por Diferença Normalizada). O FMA é projetado para gerar mapas históricos de inundação, com uma resolução espacial de 30 metros, cobrindo eventos globais onde a escassez de dados é significativa.

Figura 4.2 – Representação esquemática do fluxo de trabalho utilizado no artigo.



Fonte: Mehmood et al. (2021).

O FMA foi validado por meio de nove eventos de inundação em várias partes do mundo, incluindo Austrália, Bangladesh, Canadá, Camboja, Índia, Moçambique, Sri Lanka e Tailândia. Os resultados foram comparados com mapas de inundação de agências nacionais e regionais, utilizando métricas como taxa de verdadeiros positivos (TPR), taxa de falsos positivos (FPR) e acurácia geral. A TPR variou de 71% a 90%, com uma acurácia geral entre 74% e 89%, o que indica uma alta precisão na detecção de áreas de inundação.

Os melhores desempenhos foram observados em áreas rurais com pouca cobertura arbórea e estruturas urbanas, como no evento de inundação em Queensland, Austrália, que teve uma TPR de 90% e acurácia de 89%. Em áreas urbanas densas, como em Dhaka e Bangkok, o desempenho foi ligeiramente inferior, com uma TPR em torno de 71% a 74%, devido à dificuldade de distinguir a água em áreas construídas.

O FMA oferece uma solução econômica e eficiente para o mapeamento de inundações, especialmente em países do hemisfério sul, onde a criação de sistemas de alerta e mapas históricos é uma tarefa cara e demorada. O algoritmo pode ser integrado em sistemas de planejamento de uso da terra, serviços de emergência e até em seguros agrícolas, fornecendo dados valiosos para mitigar os impactos de inundações.

O artigo conclui que o FMA pode ser uma ferramenta eficaz para complementar mapas de risco de inundação e auxiliar em sistemas de monitoramento e mitigação. O algoritmo também pode gerar dados de treinamento para modelos de aprendizado de máquina voltados para previsão de inundações. No entanto, reconhece-se que há espaço para melhorias, como a inclusão de dados Sentinel-2 no fluxo de trabalho e a combinação com dados demográficos para criar uma ferramenta mais abrangente.

5 METODOLOGIA

Neste capítulo, será detalhada a metodologia adotada para o desenvolvimento da plataforma web de visualização das veredas. A metodologia descrita busca garantir que a plataforma atenda aos requisitos funcionais e não funcionais definidos.

A metodologia abrange desde a concepção do sistema, passando pela escolha das tecnologias e ferramentas adequadas, até a implementação e validação da plataforma. Serão discutidos os critérios de usabilidade e acessibilidade considerados no design da interface, bem como as estratégias para garantir a precisão e a atualização contínua das informações geoespaciais apresentadas.

5.1 FLUXO DE TRABALHO

O fluxo de trabalho, conforme disponibilizado graficamente no apêndice A, para o desenvolvimento e implementação da plataforma web destinada à visualização de veredas em um mapa interativo trabalho está estruturado em cinco etapas principais: treinamento do modelo, validação do modelo, predição, armazenamento de dados e visualização na plataforma.

A primeira etapa é destinada ao treinamento do algoritmo de inteligência artificial. O processo é iniciado com a inserção de imagens de treinamento, que são processadas para a geração de rótulos (labels), os quais são utilizados para validar a resposta do modelo. O modelo é, então, avaliado com base nas respostas obtidas, e, caso não atinja o desempenho desejado, ele é atualizado, e o processo é repetido iterativamente até que os critérios de adequação sejam satisfeitos, momento em que o treinamento é considerado concluído.

Na etapa subsequente, é realizada a validação do modelo treinado. Para que seja feita a validação, são inseridas as imagens do conjunto de validação para avaliar o desempenho do modelo em um conjunto de dados que ele nunca viu durante o treinamento. As imagens de validação são processadas e rotuladas, e o modelo treinado é carregado para que seja avaliado quanto à sua adequação. Após o processamento das imagens e a comparação com as saídas reais esperadas, realiza-se a verificação de parâmetros para a avaliação de presença problemas no modelo treinado como overfitting ou underfitting. Caso o modelo não seja considerado adequado, o fluxo retorna para a etapa de treinamento, onde novas iterações podem ocorrer até que o modelo atenda aos requisitos estabelecidos. Se o modelo for considerado satisfatório, a etapa de validação é finalizada.

Na terceira etapa, o modelo é carregado com as imagens do conjunto de predição.

Através das imagens, o modelo realiza a classificação dos elementos visuais como veredas e a predição de como serão as veredas das áreas inseridas. Os resultados obtidos então são salvos para serem utilizados nas etapas subsequentes do fluxo de trabalho.

Em seguida, na etapa de armazenamento de dados, as imagens processadas durante a etapa de predição são transferidas e armazenadas em um bucket do Amazon S3. Este processo assegura que todas as imagens e resultados das predições estejam disponíveis para uso posterior na plataforma de visualização.

Finalmente, com todas as etapas de ciência de dados devidamente finalizadas, a última etapa consiste na definição do fluxo de dados e resultados obtidos para abastecimento da Plataforma de Visualização, o qual foi desenvolvido valendo-se das tecnologias como HTML, CSS e JavaScript. As imagens obtidas como resultado das etapas anteriores, armazenadas na Amazon S3, abastecem a plataforma e os dados são implantados na plataforma web, onde um mapa dinâmico é gerado com base nos dados obtidos do Google Maps. Este mapa interativo é então disponibilizado na plataforma, permitindo a visualização das veredas de forma navegável e interativa.

5.2 REQUISITOS

Nesta seção, são delineados os requisitos funcionais e não funcionais essenciais para o desenvolvimento da plataforma web de visualização de veredas em mapa interativo do Brasil.

Os requisitos funcionais descrevem as funcionalidades específicas que a plataforma deve oferecer para atender às necessidades dos usuários. Esses requisitos incluem as operações básicas, como a capacidade de exibir veredas em um mapa interativo, filtrar e pesquisar locais por critérios específicos, acessar informações detalhadas sobre cada vereda, e atualizar os dados geoespaciais em tempo real. Esses aspectos definem as funcionalidades essenciais que o sistema deve proporcionar para cumprir seu objetivo principal de visualização e análise geoespacial.

Por outro lado, os requisitos não funcionais tratam das características que influenciam a qualidade e a performance da plataforma. Esses requisitos incluem critérios como a escalabilidade, garantindo que a plataforma suporte um número crescente de usuários e dados sem comprometer o desempenho; a segurança, assegurando a proteção dos dados sensíveis e a privacidade dos usuários; a usabilidade, garantindo que a interface seja intuitiva e fácil de navegar; e a disponibilidade, garantindo que a plataforma esteja acessível e operacional em tempo integral. Também são considerados aspectos como a compatibilidade com diferentes dispositivos e navegadores, e a eficiência do sistema em

termos de tempo de resposta e consumo de recursos.

A definição clara e precisa desses requisitos é fundamental para orientar o processo de desenvolvimento, garantindo que a plataforma não só atenda às necessidades funcionais dos usuários, mas também ofereça uma experiência de uso robusta, segura e de alta qualidade.

5.2.1 Requisitos Funcionais

A seguir, são apresentados os requisitos funcionais para o desenvolvimento da plataforma web, considerando as necessidades de navegação, exibição de dados geoespaciais e manipulação de imagens. Estes requisitos foram elaborados para garantir que a plataforma atenda às necessidades dos usuários, fornecendo um ambiente eficiente e acessível para a visualização, análise e download de informações sobre veredas e suas respectivas imagens.

- 1. Visualização Interativa de Mapa com Divisões Administrativas:** A plataforma deverá exibir um mapa interativo do Brasil, incluindo todas as suas divisões administrativas até o nível municipal. O mapa será navegável, com funcionalidades de zoom e pan, permitindo que o usuário explore as diferentes regiões. Ao passar o cursor sobre uma divisão administrativa (estados e municípios), o nome da área será exibido, e, ao clicar, o sistema deverá listar todas as imagens coletadas no local, facilitando o acesso a dados específicos (GIUZIO et al., 2022).
- 2. Exibição e Navegação de Imagens:** A plataforma deve possibilitar a visualização de todas as imagens já coletadas para uma determinada divisão administrativa. Além disso, os usuários terão a capacidade de navegar entre essas imagens de forma cronológica ou com base em parâmetros específicos. Deve ser possível ampliar ou reduzir o tamanho das imagens exibidas, proporcionando uma melhor análise visual dos dados disponíveis.
- 3. Pesquisa Avançada de Imagens:** A plataforma deverá oferecer uma funcionalidade de pesquisa avançada, permitindo que o usuário filtre as imagens por critérios como data da coleta, local de coleta (região ou coordenadas geográficas), intervalo de coordenadas (latitude e longitude), e resultado da análise de identificação de veredas (positivo ou negativo). Esta funcionalidade é essencial para a busca eficiente de imagens que atendam a necessidades específicas de análise e monitoramento.
- 4. Exibição de Resultados de Estimções de Veredas:** A plataforma deverá exibir os resultados de todas as estimções realizadas com as imagens coletadas até o momento, permitindo que o usuário visualize e compare os dados de diferentes

períodos. Esses resultados serão exibidos juntamente com as respectivas imagens, facilitando a análise das mudanças no ambiente das veredas ao longo do tempo.

- 5. Baixar Imagens e Resultados:** O usuário terá a possibilidade de baixar qualquer imagem exibida na plataforma, bem como os dados e resultados de todas as estimações realizadas até aquele momento. Esses dados podem incluir informações geoespaciais, coordenadas, e métricas relacionadas às veredas identificadas nas imagens, proporcionando uma base para análises posteriores.

5.2.2 Requisitos Não Funcionais

Os requisitos não funcionais estabelecem as características de qualidade e desempenho que a plataforma deve apresentar para garantir sua eficiência e adequação ao propósito de visualização de veredas em um mapa interativo. A seguir, são descritos os principais requisitos não funcionais para a plataforma, considerando aspectos de desempenho, usabilidade, disponibilidade, escalabilidade, manutenibilidade e extensibilidade.

- 1. Desempenho:** A plataforma deve garantir um desempenho elevado, sendo responsiva e rápida ao carregar imagens, mesmo em situações de uso simultâneo por diversos usuários. A latência de carregamento deve ser minimizada, assegurando uma experiência fluida, particularmente no acesso e manipulação de grandes volumes de dados geoespaciais e imagens de alta resolução. O desempenho deve ser otimizado para funcionar eficientemente, mesmo quando o sistema estiver sob carga elevada em momentos de maior demanda.
- 2. Usabilidade:** A interface da plataforma deve ser intuitiva e de fácil utilização, mesmo para usuários com pouca experiência em sistemas de georreferenciamento ou manipulação de dados geoespaciais. O design da interface deve seguir princípios de usabilidade centrados no usuário, proporcionando uma navegação clara e acessível. Além disso, deve facilitar a execução das principais tarefas, como a visualização de veredas, navegação pelo mapa e download de imagens, sem exigir conhecimento técnico especializado.
- 3. Disponibilidade:** A plataforma deve estar disponível de maneira contínua para consulta por pesquisadores e instituições da área. Contudo, considerando o público-alvo restrito e a expectativa de uma demanda de uso moderada, não é necessária uma disponibilidade extremamente elevada (GIUZIO et al., 2022). A plataforma deve atender adequadamente as requisições dos usuários durante horários de pico, sem a necessidade de consumir recursos excessivos em momentos de inatividade.
- 4. Escalabilidade:** O sistema deve ser escalável o suficiente para lidar com variações ocasionais no número de usuários e no volume de dados a serem processados. Embora

o uso da plataforma possa ser esporádico e variável, ela deve ser capaz de atender de forma eficiente às demandas em horários de pico e acomodar o crescente volume de imagens e dados geoespaciais que podem ser carregados no sistema. A escalabilidade deve garantir que a plataforma se adapte a um aumento gradual de usuários e dados ao longo do tempo, sem comprometer seu desempenho (GIUZIO et al., 2022).

- 5. Manutenibilidade e Extensibilidade:** A plataforma deve ser altamente manutível e extensível, permitindo futuras adaptações e melhorias. Como se trata de um projeto vinculado ao INPE, é esperado que o sistema seja projetado de forma modular, facilitando a incorporação de novas funcionalidades ou ajustes necessários para atender às demandas de pesquisas futuras. A arquitetura deve permitir a fácil atualização de componentes e a integração de novos métodos de análise geoespacial ou processamento de imagens, sem a necessidade de grandes reestruturações do sistema (GIUZIO et al., 2022).
- 6. Segurança:** O sistema deve garantir a proteção de dados sensíveis e assegurar a privacidade dos usuários e das instituições que utilizarem a plataforma. Medidas de segurança, como autenticação de usuários, controle de acesso e criptografia de dados, devem ser implementadas para prevenir acessos não autorizados e proteger as informações geoespaciais coletadas.
- 7. Compatibilidade Multidispositivo e Multiplataforma:** A plataforma deve ser compatível com diversos dispositivos e navegadores, garantindo uma experiência de uso consistente e eficiente, tanto em desktops quanto em dispositivos móveis, como tablets e smartphones. O sistema deve ser otimizado para funcionar nos principais navegadores (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge), sem comprometer a usabilidade ou o desempenho.
- 8. Eficiência de Rede:** Considerando a possibilidade de uso da plataforma em áreas com conectividade limitada, o sistema deve ser otimizado para consumir o mínimo de largura de banda possível, sem comprometer a qualidade da experiência do usuário. Deve haver um equilíbrio entre o tamanho dos dados carregados e o desempenho da rede, garantindo que a plataforma seja utilizável em diferentes condições de acesso à internet.

6 DESENVOLVIMENTO BACK-END

Este capítulo é dedicado à descrição do desenvolvimento do componente back-end da plataforma web interativa projetada para a visualização dinâmica das veredas no Cerrado brasileiro. A importância deste componente reside na sua capacidade de gerenciar eficientemente a lógica de construção dos mapas, a autenticação de usuários, e a integração e comunicação contínua com modelos avançados de aprendizado de máquina. Este segmento do sistema é crucial, pois ele processa e disponibiliza dados essenciais obtidos e refinados a partir de extensas bases de dados geoespaciais e imagens de satélite, utilizando para isso algoritmos de Deep Learning para classificar e prever mudanças na paisagem das veredas ao longo do tempo.

A seleção do framework Django, escrito em Python, para o desenvolvimento do back-end foi estratégica. Python é amplamente reconhecido por sua forte aplicação em tarefas de computação científica e análise de dados, além de possuir uma comunidade robusta e uma extensa biblioteca de pacotes, o que facilita a implementação de operações complexas de processamento de dados. Django, especificamente, oferece um alto nível de eficiência no gerenciamento de modelos de dados, segurança, e fluxos de autenticação, além de permitir uma escalabilidade eficaz. Essas características são fundamentais para suportar o volume e a complexidade dos dados manipulados na plataforma.

Além disso, a arquitetura de back-end foi projetada para ser robusta e escalável, utilizando uma infraestrutura de nuvem fornecida pelos serviços da AWS. A plataforma da Amazon, além de já ser utilizada por outras frentes do projeto Nexus Biodiversity, foi escolhida por sua capacidade comprovada de fornecer soluções escaláveis e seguras para hospedagem em nuvem, essenciais para o armazenamento de grandes conjuntos de dados e para o processamento intensivo necessário para os modelos de aprendizado de máquina. A flexibilidade da AWS em termos de escalabilidade computacional permite que o sistema se adapte às demandas variáveis de carga de processamento, garantindo assim uma entrega de dados eficiente e em tempo real para os usuários finais da plataforma.

6.1 ESCOLHA E JUSTIFICATIVAS DAS TECNOLOGIAS UTILIZADAS

6.1.1 Python e Django

A escolha de Python e do framework Django para o desenvolvimento do back-end da plataforma de visualização de veredas no Cerrado brasileiro é fundamentada por suas capacidades singulares de gestão de aplicativos complexos e desenvolvimento ágil. Esta seleção é estratégica no contexto deste trabalho, onde a eficiência na manipulação de

grandes volumes de dados geoespaciais e a robustez na segurança são primordiais.

O Python desempenha um papel fundamental no projeto, agindo como um pilar de robustez, particularmente por meio de suas bibliotecas especializadas em integração via APIs e framework Django para o desenvolvimento back-end.

A biblioteca “requests“ de Python simplifica significativamente a integração com plataformas externas, como o Google Earth Engine. Esta integração é vital para a automatização do fluxo de dados, que sustenta as funcionalidades analíticas avançadas da plataforma. Com “requests“, a plataforma pode eficientemente consultar e recuperar dados de sensoriamento remoto, permitindo que análises ambientais sejam realizadas com dados atualizados e precisos. Essa capacidade de integração facilita a implementação de modelos de aprendizado de máquina e outras análises estatísticas, que são fundamentais para prever e monitorar as dinâmicas das veredas no contexto do Cerrado brasileiro.

O Django, com sua arquitetura sofisticada, desempenha um papel vital no desenvolvimento ágil e eficaz do back-end, simplificando a administração de componentes essenciais do sistema. A implementação deste framework é particularmente benéfica para a gestão de usuários na plataforma, uma vez que seu módulo de autenticação integrado assegura um controle seguro e eficiente. Isso é crucial em um contexto onde múltiplos usuários, incluindo pesquisadores e formuladores de políticas, dependem de acesso confiável a informações sensíveis relativas às veredas.

Adicionalmente, a interface administrativa gerada automaticamente por Django facilita uma administração eficiente do conteúdo e das atividades na plataforma. Esta funcionalidade é indispensável para monitorar a integridade e precisão dos dados apresentados, garantindo que as visualizações das veredas sejam precisas e confiáveis.

A utilização do Object-Relational Mapping (ORM) do Django também é fundamental, pois minimiza a susceptibilidade a erros na manipulação de modelos de dados. Isso simplifica as operações com os extensos conjuntos de dados provenientes do Google Earth Engine, que são vitais para conduzir análises ambientais rigorosas.

O suporte do Django a middlewares é outro recurso estratégico, permitindo a implementação de funções personalizadas para processar requisições. Este aspecto é essencial para a filtragem e processamento específicos dos dados geoespaciais das veredas, facilitando intervenções precisas e contextualizadas no fluxo de dados.

Finalmente, a robustez do Django em incorporar proteções contra vulnerabilidades comuns da web reforça a segurança da plataforma, salvaguardando a integridade dos

dados sobre as veredas. Este nível de segurança é crucial para manter a credibilidade das pesquisas e análises disponibilizadas à comunidade científica e ao público em geral, assegurando que as informações são protegidas contra ameaças externas.

6.1.2 Google Earth Engine

O Google Earth Engine é uma plataforma robusta para o processamento geoespacial que desempenha um papel crucial no nosso projeto, especialmente na manipulação visual dos dados das veredas. Esta plataforma não apenas fornece acesso a uma vasta quantidade de dados de sensoriamento remoto, mas também possui capacidades avançadas de visualização e análise que são essenciais para a nossa investigação das veredas no Cerrado brasileiro.

Utilizando o GEE, podemos extrair características visuais específicas dos mapas, que são fundamentais para identificar e monitorar as veredas. A plataforma permite aplicar filtros avançados sobre os dados de imagem, facilitando a separação de características relevantes como vegetação, corpos d'água e alterações antrópicas no terreno. Esta filtragem é crucial para isolar as áreas de veredas das demais características do Cerrado, permitindo uma análise mais precisa e focada.

Além da filtragem, o GEE possibilita a utilização de máscaras para melhorar a visualização e disposição dos dados sobre as veredas. As máscaras podem ser aplicadas para destacar regiões específicas ou para ocultar áreas menos relevantes, melhorando assim a clareza dos mapas gerados. Essa técnica é particularmente útil para sobrepor dados de veredas sobre imagens de satélite, permitindo que pesquisadores e tomadores de decisão visualizem as informações de forma intuitiva e acessível.

A capacidade do GEE de manipular visualmente os dados de sensoriamento remoto transforma a forma como as informações sobre as veredas são apresentadas e interpretadas. Ao aplicar filtros e máscaras, transformamos dados brutos em representações visuais claras que podem ser facilmente compreendidas por cientistas, gestores ambientais e o público. Essas representações visuais não apenas facilitam o monitoramento das condições atuais das veredas, mas também ajudam na previsão de tendências de mudança, essenciais para a conservação e o manejo sustentável do bioma Cerrado.

Portanto, o uso do Google Earth Engine amplia significativamente nossas capacidades de análise visual e interpretação de dados geoespaciais, tornando-se uma ferramenta indispensável para a nossa plataforma de visualização de veredas. Esta integração não só enriquece nosso trabalho de formatura com insights valiosos e representações detalhadas, mas também reforça o impacto científico e aplicado do projeto na conservação ambiental e

na pesquisa.

6.2 ARQUITETURA DO SISTEMA

A arquitetura do sistema para a plataforma de visualização de veredas no Cerrado brasileiro foi planejada seguindo o padrão Model-View-Controller (MVC), uma metodologia bem estabelecida utilizada pelo Django para organizar o código de forma lógica e modular. Este padrão divide o sistema em três componentes principais, cada um responsável por uma aspecto distinto da aplicação, o que facilita a manutenção e a escalabilidade do projeto.

6.2.1 *Models* (Modelos)

Os modelos no Django são a espinha dorsal da estrutura de dados do sistema. Eles definem a organização lógica dos dados, cada modelo representando uma tabela no banco de dados. No contexto deste projeto, os modelos foram criados para capturar todos os dados essenciais relacionados às veredas, incluindo:

- **Metadados das Imagens:** Cada conjunto de dados de imagens de satélite armazenado tem seu modelo correspondente que inclui atributos como data da captura, resolução, e parâmetros específicos do sensor.
- **Informações das Áreas de Veredas:** Dados geoespaciais que definem a localização e as características específicas das veredas, como tipo de vegetação, status de conservação, e histórico de alterações ambientais.
- **Registros de Análises Temporais:** Modelos que armazenam os resultados das análises feitas sobre as séries temporais dos dados capturados, permitindo o acompanhamento de mudanças nas veredas ao longo do tempo.

Utilizando o ORM (Object-Relational Mapping) do Django, estes modelos abstraem as operações do banco de dados, permitindo que os desenvolvedores interajam com a base de dados usando Python, sem a necessidade de escrever consultas SQL complexas. Essa abstração também facilita a portabilidade da aplicação para diferentes sistemas de banco de dados, oferecendo uma flexibilidade significativa no manejo dos dados.

6.2.2 *Views* (Visões)

As visões em Django funcionam como o componente que coordena a lógica de negócios, processando os dados recebidos das requisições dos usuários e enviando respostas

apropriadas. No projeto em questão, as visões interagem diretamente com o Google Earth Engine para:

- **Solicitar Dados de Satélite:** Comandos são enviados para o Google Earth Engine para recuperar imagens e dados específicos baseados em critérios como datas, localização geográfica ou outros filtros ambientais relevantes.
- **Processar Dados Recebidos:** As visões aplicam algoritmos de processamento e análise para interpretar os dados de satélite, extrair características relevantes das veredas e realizar comparações temporais.

Estas visões são implementadas como classes ou funções em Python que lidam com as requisições HTTP, executando a lógica necessária e retornando a resposta apropriada ao cliente, na forma de dados HTML para visualização no navegador.

6.2.3 *Controllers* (Contraladores)

Embora o Django não utilize explicitamente o termo “controlador” no mesmo sentido que outros frameworks MVC, essa funcionalidade é incorporada dentro do sistema de roteamento de URLs, que configura quais visões são chamadas com base na URL acessada.

O sistema de roteamento permite que o desenvolvedor defina claramente quais padrões de URL (endereços) correspondem a quais visões, organizando o fluxo de entrada e saída de dados dentro da aplicação. Este método direciona as requisições dos usuários para as visões corretas, garantindo que a lógica de negócios seja processada eficientemente e que os dados corretos sejam retornados.

6.3 IMPLEMENTAÇÃO

A implementação deste projeto foca na integração da inteligência artificial com a plataforma de visualização geoespacial, resultando em um sistema robusto para análise de veredas. Inicialmente, a IA é treinada em instâncias Amazon EC2 para identificar contornos de veredas, com os resultados armazenados no Amazon S3.

Utilizando a API do Google Static Maps, imagens de satélite das áreas detectadas são obtidas e convertidas do formato PNG para TIFF, necessário para o Google Earth Engine que assegura a precisão geoespacial das imagens.

Em seguida, essas imagens são carregadas e sobrepostas no GEE, permitindo a integração com outras camadas georreferenciadas e uma visualização detalhada dos

contornos sobre as imagens de satélite. Finalmente, o sistema oferece uma interface interativa onde os usuários podem explorar as veredas e acessar informações detalhadas.

Cada etapa deste processo será abordada com mais detalhes nas próximas seções, evidenciando a estrutura e os mecanismos de integração que tornam essa plataforma de análise ambiental eficiente e precisa.

6.3.1 Aquisição de Arquivos do Amazon S3

Esta subseção detalha o funcionamento do script desenvolvido para acessar e recuperar arquivos armazenados no Amazon S3, uma solução essencial para o gerenciamento e o controle de imagens no projeto. O código implementado usa a biblioteca boto3 para a comunicação direta com o bucket S3, aproveitando as funcionalidades de criação de URLs temporárias para o acesso seguro aos arquivos.

6.3.1.1 Processo de Conexão e Aquisição

O primeiro passo do script é a configuração de um cliente AWS S3 utilizando a biblioteca boto3. Esse cliente é configurado com as credenciais de acesso da conta AWS associada ao projeto, que incluem uma chave de acesso e uma chave secreta. Com essas credenciais, o cliente é autorizado a interagir com o bucket específico onde os arquivos de mídia são armazenados. Esse processo garante que o script tenha a permissão necessária para listar, ler e gerar links temporários para os arquivos no bucket.

Com o cliente configurado, o script solicita ao bucket uma lista de todos os arquivos armazenados. Esse retorno fornece informações detalhadas sobre cada arquivo, como o nome, o tamanho e a data de modificação. A listagem é processada de forma a verificar se o tipo de cada arquivo corresponde a uma imagem. A verificação é baseada nas extensões comuns de imagens, como .jpg, .jpeg e .png. Esta etapa é crucial para filtrar os arquivos de interesse e preparar os objetos de controle que serão manipulados pelo sistema.

Após a listagem e identificação das imagens, o script utiliza uma função do cliente boto3 para gerar URLs temporárias para cada imagem. Essas URLs são configuradas para expirar após um período determinado, o que limita o tempo em que os arquivos estão acessíveis diretamente pela internet, reduzindo significativamente os riscos de segurança. A URL gerada permite que o arquivo seja acessado como se estivesse disponível publicamente, mas apenas dentro do período de validade da URL. Essa abordagem fornece uma camada de segurança importante, pois elimina a necessidade de tornar os arquivos públicos no bucket S3.

Cada imagem identificada no bucket é associada a um objeto de controle criado

pelo script, que armazena detalhes como o nome do arquivo e o caminho completo dentro do bucket S3. Esses objetos de controle são instâncias de uma classe desenvolvida especificamente para armazenar e manipular informações de imagem no projeto. Eles são projetados para manter a integridade das referências entre o arquivo original no S3 e seu representante interno no sistema, facilitando a recuperação e a manipulação das imagens em etapas subsequentes do projeto.

6.3.1.2 Integração com Outros Módulos do Sistema

Esses objetos de controle de imagem podem ser facilmente acessados e manipulados por outros módulos do projeto. Por exemplo, caso um módulo necessite exibir ou processar uma imagem, ele pode utilizar o objeto de controle correspondente para obter as informações necessárias de acesso. Esta arquitetura modular permite que o sistema interaja com as imagens sem necessidade de reconfigurar a conexão com o S3, o que otimiza o desempenho e minimiza o número de requisições ao bucket.

6.3.2 Extração e Conversão de Coordenadas Geográficas de Imagens .tif

Esta subseção descreve o processo implementado para extrair e converter as coordenadas geográficas das extremidades de imagens no formato .tif, uma funcionalidade essencial para o correto georreferenciamento e manipulação das imagens no projeto. Esse processo inclui a verificação do tipo de arquivo, a extração dos limites espaciais da imagem e a conversão das coordenadas para um formato mais acessível, utilizando as bibliotecas rasterio e pyproj.

6.3.2.1 Validação e Extração Inicial de Coordenadas

O processo de extração de coordenadas inicia-se com uma verificação do tipo de arquivo, assegurando que ele possui a extensão .tif. Esse formato é amplamente utilizado para dados espaciais e permite o armazenamento de informações de georreferenciamento. A validação evita que arquivos incompatíveis sejam processados, o que otimiza o desempenho do sistema e reduz a ocorrência de erros durante a execução.

Após a verificação, o script utiliza a biblioteca rasterio para acessar os metadados geográficos da imagem. Esta biblioteca possibilita a leitura das coordenadas de suas extremidades mínima e máxima (x e y), que representam os limites de abrangência espacial da imagem. Esses valores fornecem a base para o posicionamento geográfico da imagem e são fundamentais para a conversão subsequente.

6.3.2.2 Conversão das Coordenadas para Graus

As coordenadas extraídas com rasterio podem estar em um sistema de projeção específico, como UTM, o que limita sua usabilidade em visualizações geográficas padrão. Para resolver isso, o script utiliza a biblioteca pyproj, que converte as coordenadas para o sistema de latitude e longitude em graus. Este formato é amplamente reconhecido e utilizado em ferramentas de mapeamento, o que facilita a integração com outras plataformas e melhora a interoperabilidade do sistema.

Após a conversão, as coordenadas mínimas e máximas (latitude e longitude) são armazenadas em objetos de controle dentro do sistema. Esses objetos preservam a referência geográfica de cada imagem, permitindo que outros módulos acessem essas informações de maneira simplificada. A estrutura de armazenamento das coordenadas facilita o uso das imagens em visualizações de mapa, cálculos de distância e outras análises espaciais.

6.3.2.3 Aplicação e Integração das Coordenadas Geográficas

Com as coordenadas geográficas devidamente estruturadas e armazenadas, o sistema está preparado para integrar as imagens com módulos de visualização. Cada imagem pode ser posicionada com precisão em mapas interativos, utilizando as coordenadas de latitude e longitude para assegurar um alinhamento correto. Essa funcionalidade é essencial para operações que dependem da localização geográfica precisa, como sobreposição de camadas de dados e análises espaciais interativas.

Além disso, o uso das bibliotecas rasterio e pyproj contribui para a precisão e confiabilidade dos dados extraídos. A escolha dessas ferramentas para geoprocessamento garante a qualidade das coordenadas geográficas e facilita a interoperabilidade com outras plataformas de visualização e análise de dados geográficos, o que é um diferencial importante para o projeto.

6.3.3 Aquisição e Conversão de Imagens de Satélite para o Google Earth Engine

Esta subseção descreve o processo de obtenção e preparação de imagens de satélite para integração com a plataforma Google Earth Engine (GEE). A abordagem inclui a requisição das imagens, o tratamento de erros na aquisição, a conversão do formato da imagem e a preparação final para disponibilização no GEE.

6.3.3.1 Requisição de Imagens de Satélite com a API do Google Static Maps

Para adquirir imagens de satélite que serão usadas no GEE, o processo inicia com a criação de uma requisição HTTP através da API do Google Static Maps. Utilizando

a biblioteca requests em Python, a solicitação é estruturada com parâmetros essenciais, como latitude e longitude da área de interesse, além da chave de API para autenticação. Esses parâmetros garantem que a imagem obtida represente com precisão a região de estudo selecionada.

A imagem retornada pela API está no formato PNG, escolhido por sua visualização direta e fácil manipulação. No entanto, devido à incompatibilidade do formato PNG com a plataforma GEE, é necessário realizar uma conversão para um formato adequado que permita o processamento e análise geoespacial da imagem dentro do GEE.

6.3.3.2 Monitoramento e Tratamento de Erros na Aquisição de Imagens

Durante o processo de requisição, o script monitora a resposta da API para assegurar a aquisição correta da imagem. O código de resposta HTTP é verificado para confirmar o sucesso da solicitação. Em caso de falhas ou erros na resposta, são aplicados mecanismos de tratamento de exceções para gerenciar esses problemas, assegurando que o processo de aquisição continue de forma robusta. Este tratamento de erros é essencial para a continuidade do fluxo de trabalho e para a estabilidade do sistema, prevenindo interrupções causadas por falhas de comunicação com a API.

6.3.3.3 Conversão de Imagens de PNG para TIFF

Após a obtenção da imagem em formato PNG, segue-se a conversão para o formato .tif, que é compatível com o Google Earth Engine. Esta conversão é realizada com a biblioteca rasterio, amplamente utilizada para manipulação de dados geoespaciais em Python. Durante essa etapa, são ajustados parâmetros cruciais, incluindo o sistema de referência de coordenadas (CRS), que define a projeção espacial da imagem.

Além da conversão, são aplicadas transformações geoespaciais para garantir que a imagem mantenha a escala e o posicionamento corretos da área de interesse. Este ajuste é fundamental para assegurar que a imagem .tif esteja alinhada com precisão a outros dados georreferenciados, o que é essencial para a sobreposição de camadas e análises espaciais no GEE.

6.3.3.4 Preparação e Disponibilização da Imagem para o Google Earth Engine

Na etapa final, a imagem convertida para .tif é preparada para análise e carregada na plataforma Google Earth Engine. Após o carregamento, a imagem passa a fazer parte do conjunto de mapas e camadas geoespaciais disponíveis no sistema. Isso permite que a imagem de satélite seja visualizada e analisada em conjunto com outras camadas de dados, possibilitando uma visão abrangente e detalhada dos dados ambientais.

A integração das imagens de satélite no GEE potencializa a capacidade analítica do sistema, permitindo a sobreposição de diferentes camadas geográficas e contribuindo para a interpretação de aspectos ambientais específicos. A imagem, agora processada e georreferenciada, torna-se uma ferramenta central para estudos ambientais e análises de mudanças na paisagem, promovendo uma abordagem robusta para o monitoramento de áreas de interesse e a exploração de variáveis ambientais ao longo do tempo.

6.3.4 Transferência de Imagens ao Google Earth Engine

Essa subseção descreve o processo de organização e envio das imagens ao Google Earth Engine, essencial para visualização e análise geoespacial. O fluxo inclui a estruturação das imagens em coleções específicas, sua transferência via Google Cloud Storage e a conversão em ativos do GEE, garantindo acesso eficiente e estruturado aos dados.

6.3.4.1 Organização das Coleções de Imagens no Google Earth Engine

A transferência de imagens para o Google Earth Engine exige uma estrutura de organização que facilita o acesso e manipulação dos dados. Cada coleção de imagens, conhecida como *imageCollection*, agrupa dados de acordo com a área de interesse, chamada de “nexusArea“, e a data de captura das imagens. A partir desses dados, duas pastas são criadas para cada *imageCollection*, permitindo a separação entre as imagens de contorno e as imagens de satélite.

A pasta destinada às imagens de contorno é nomeada com a palavra *outlines*, seguida pela área de interesse e pela data da coleção no formato “ano-mês-dia“. Por exemplo, uma coleção de imagens relacionada a uma área chamada “Norte“ e datada de 15 de maio de 2023 geraria uma pasta chamada “*outlines_Norte_2023-05-15*“. De forma similar, a pasta que armazena as imagens de satélite é nomeada com a expressão *satelliteimages*, acompanhada pela área de interesse e a mesma data, resultando em “*satelliteimages_Norte_2023-05-15*“.

Dentro de cada pasta, as imagens são nomeadas com base na chave primária, ou *primary key* (pk), do objeto armazenado no banco de dados do sistema. Para as imagens de contorno, o nome segue o padrão “*outline-pk*“, enquanto para as imagens de satélite o padrão é “*satelliteimage-pk*“. Esse modelo de nomeação permite a identificação única de cada imagem, facilitando a rastreabilidade e integridade dos dados no sistema.

6.3.4.2 Fluxo de Transferência de Imagens para o Google Earth Engine

O Google Earth Engine não permite o upload direto de arquivos, por conta disso, o fluxo de transferência requer uma etapa intermediária de armazenamento no Google

Cloud Storage (GCS). O processo de envio inicia com o upload das imagens para o GCS e, em seguida, ocorre a conversão desses arquivos em ativos do GEE, tornando-os acessíveis e manipuláveis dentro do ambiente Earth Engine. Essa transferência e organização dos dados ocorrem de maneira automatizada para garantir a continuidade do processo e a organização dos arquivos de forma consistente.

6.3.4.3 Bibliotecas Utilizadas no Processo

Para a implementação desse fluxo de transferência e armazenamento, duas bibliotecas principais são utilizadas: “*storage*” e “*ee*”. A biblioteca *storage* é responsável pelo gerenciamento dos arquivos no Google Cloud Storage, cuidando da organização e categorização nas pastas de contorno e satélite. A biblioteca *ee*, do Google Earth Engine, é usada para manipular os ativos no próprio GEE, permitindo a conversão das imagens previamente armazenadas no Google Cloud Storage em ativos utilizáveis no Earth Engine, mantendo o padrão de organização estabelecido para as coleções de imagens.

6.3.4.4 Integração com o Sistema Django e Identificação dos Arquivos

Cada imagem transferida é identificada e organizada de acordo com a chave primária do objeto, definida no banco de dados Django. Este identificador único facilita a atualização e manutenção das imagens no Google Earth Engine, dispensando a necessidade de renomeação ou reorganização manual. A chave primária serve como um identificador fixo, permitindo que cada imagem carregada seja facilmente associada à sua coleção e área de interesse específica.

Essa integração com o Django também facilita o monitoramento do status de cada imagem no fluxo de transferência e armazenamento, adicionando uma camada de controle e segurança ao processo.

6.3.5 Aplicação do Google Earth Engine

O Google Earth Engine exerce uma função crucial na estrutura do back-end deste projeto, fornecendo uma plataforma poderosa para visualização e manipulação de dados geoespaciais. A aplicação utiliza o GEE por meio de um script em JavaScript que recebe parâmetros via URL, especificamente a região de interesse (*nexusArea*) e uma data (*data*), para determinar quais dados exibir. A partir desses parâmetros, o sistema localiza e carrega os dados geoespaciais apropriados a partir de pastas previamente configuradas.

Ao iniciar o processamento, o script acessa as pastas de “*outlines*” e “*satelliteImages*”, onde estão armazenadas as geometrias de contorno das áreas e as imagens de satélite correspondentes. Essas geometrias de contorno, representadas como *outlines*, são exibidas

diretamente no mapa, delineando as fronteiras da região especificada. Paralelamente, as imagens de satélite correspondentes aparecem em um painel de controle à parte, permitindo ao usuário visualizar as imagens específicas da área de interesse. Esse painel, além de exibir a imagem de satélite, inclui controles de navegação que possibilitam ao usuário avançar ou retroceder entre diferentes outlines, enquanto o mapa é automaticamente centralizado na região do contorno selecionado. Dessa forma, o sistema facilita a navegação e permite uma análise detalhada das informações visuais.

A interatividade do GEE é um aspecto fundamental do sistema. O usuário pode clicar diretamente em qualquer outline no mapa para ver a imagem de satélite correspondente, que é imediatamente carregada no painel de visualização. Como alternativa, o usuário pode explorar as imagens usando as setas de navegação no próprio painel, que permitem percorrer os outlines em sequência. Cada interação, seja por meio de cliques nos contornos ou pelo uso das setas, faz com que o mapa centralize automaticamente na área de interesse atual, oferecendo uma experiência de uso contínua e intuitiva.

Na interface da aplicação, diversos componentes são implementados para aprimorar a navegação. Os botões de navegação alteram a imagem exibida no painel, e um contador indica a posição da imagem atual em relação ao total de outlines carregados, orientando o usuário quanto ao progresso de visualização. A imagem de satélite é apresentada em uma miniatura, que oferece uma visão detalhada e organizada das informações visuais, reforçando a utilidade da interface.

7 DESENVOLVIMENTO FRONT-END

Este capítulo é dedicada à descrição do desenvolvimento do componente front-end da plataforma web interativa projetada para a visualização e análise das veredas no Cerrado brasileiro. A importância deste componente reside em sua capacidade de fornecer uma interface intuitiva e responsiva, que facilita o acesso e a interação dos usuários com dados complexos, como mapas dinâmicos e visualizações de dados geoespaciais detalhados. O front-end é o ponto de contato direto com o usuário, permitindo que ele visualize de forma clara e acessível as informações processadas pelo back-end e tenha uma experiência enriquecida com os dados fornecidos pela plataforma.

Para o desenvolvimento desta interface, foram utilizadas as tecnologias HTML e JavaScript, que fornecem uma base sólida para a estrutura e funcionalidade das páginas. HTML é responsável pela estruturação do conteúdo exibido, enquanto o JavaScript possibilita a interatividade e manipulação dinâmica dos elementos da página. A simplicidade e a ampla compatibilidade dessas tecnologias foram consideradas estratégicas para garantir o carregamento rápido e o acesso universal, independentemente do dispositivo ou navegador utilizado.

Além disso, a arquitetura de front-end foi planejada para garantir uma integração eficiente com o back-end, exibindo dados em tempo real e oferecendo uma experiência de navegação fluida e intuitiva. A escolha por HTML e JavaScript permite que a plataforma atenda de forma eficaz aos requisitos de visualização de dados, ao mesmo tempo em que facilita futuras expansões e ajustes necessários para a melhoria contínua da experiência do usuário.

7.1 ESCOLHA E JUSTIFICATIVAS DAS TECNOLOGIAS UTILIZADAS

A escolha das tecnologias para o desenvolvimento do front-end da plataforma visou criar uma interface acessível, responsiva e de fácil manutenção, priorizando a compatibilidade e o carregamento eficiente em diferentes dispositivos. Neste contexto, HTML e JavaScript foram as principais ferramentas selecionadas para a construção do front-end, garantindo a estruturação adequada dos conteúdos e uma interação dinâmica com os elementos da interface, fundamentais para a usabilidade da plataforma.

7.1.1 HTML

O HTML foi adotado para estruturar e organizar o conteúdo exibido na plataforma. Esta linguagem de marcação é amplamente reconhecida e compatível com todos

os navegadores modernos, o que assegura que a interface seja carregada corretamente e esteja acessível a uma ampla gama de usuários. A escolha do HTML baseou-se em sua simplicidade e na capacidade de definir a estrutura dos dados de forma semântica, facilitando tanto o desenvolvimento quanto futuras manutenções e expansões da plataforma. Além disso, a natureza padronizada do HTML permite que a plataforma ofereça um layout consistente, melhorando a experiência do usuário final.

7.1.2 JavaScript

JavaScript foi a linguagem escolhida para a programação de funcionalidades interativas, possibilitando uma experiência dinâmica e fluida. Com o uso do JavaScript, foi possível implementar funcionalidades de navegação e manipulação de conteúdo sem a necessidade de recarregar a página, proporcionando uma interação mais imediata e intuitiva para o usuário. A linguagem permite um alto nível de controle sobre os elementos da interface, viabilizando a criação de componentes interativos que enriquecem a experiência de navegação, especialmente em relação à visualização dos mapas e dados dinâmicos.

7.1.3 CSS

CSS foi a tecnologia escolhida para a estilização da interface e definição do layout visual da plataforma, garantindo uma apresentação atraente e funcional. Com o uso do CSS, foi possível criar elementos visuais consistentes e responsivos, assegurando que a aplicação seja acessível em diferentes dispositivos e tamanhos de tela.

A tecnologia permitiu implementar estilos personalizados em componentes como menus, campos de entrada e botões, destacando-se pela flexibilidade na customização. Elementos como a barra de navegação e os formulários interativos foram estilizados para proporcionar uma experiência visual coerente, integrando design moderno com facilidade de uso.

Além disso, CSS desempenhou um papel crucial na organização e reutilização de estilos, otimizando o desenvolvimento e a manutenção do código. Por meio de classes e seletores bem estruturados, foi possível unificar a aparência dos elementos em toda a aplicação, promovendo uniformidade e economizando esforços futuros em ajustes e aprimoramentos.

Sua integração com JavaScript permitiu adicionar interatividade ao design, como animações e transições suaves, enriquecendo a experiência do usuário e tornando a navegação na plataforma mais envolvente. Assim, CSS se mostrou útil para a criação de uma interface visualmente sofisticada e funcional.

7.1.4 Justificativa da Escolha

A escolha por HTML, CSS e JavaScript para o desenvolvimento da plataforma foi guiada pela necessidade de criar uma aplicação funcional, responsiva e acessível, alinhada às boas práticas de desenvolvimento web. Essas tecnologias, amplamente adotadas e suportadas por todos os navegadores modernos, foram essenciais para atender aos requisitos de estruturação, estilização e interatividade do sistema.

O HTML foi empregado como base estrutural da aplicação, permitindo organizar o conteúdo e os elementos de interface de maneira lógica e semântica. Essa escolha garantiu que a plataforma fosse compatível com diversos dispositivos e navegadores, além de facilitar a manutenção e a futura expansão do sistema. A utilização de uma estrutura modular possibilitou a reutilização de componentes, promovendo consistência visual e operacional em todas as páginas da plataforma.

O CSS desempenhou um papel crucial na definição do design e do layout da interface, contribuindo para a criação de uma experiência visual coerente e agradável. Com o uso de técnicas modernas de estilização, como design responsivo e gerenciamento de variáveis, a aplicação foi adaptada para diferentes tamanhos de tela, garantindo acessibilidade em dispositivos móveis e desktops. Elementos como barras de navegação, menus interativos e transições suaves foram implementados para proporcionar uma interação fluida e intuitiva, alinhada às expectativas dos usuários.

Já o JavaScript foi essencial para adicionar interatividade e dinamismo à plataforma. A linguagem foi utilizada para manipular os elementos da interface em tempo real, possibilitando funcionalidades como menus suspensos, validação de entradas de dados e carregamento dinâmico de conteúdo. Essas implementações tornaram a interação do usuário mais imediata e eficiente, eliminando a necessidade de recarregamento constante da página e promovendo uma experiência mais fluida.

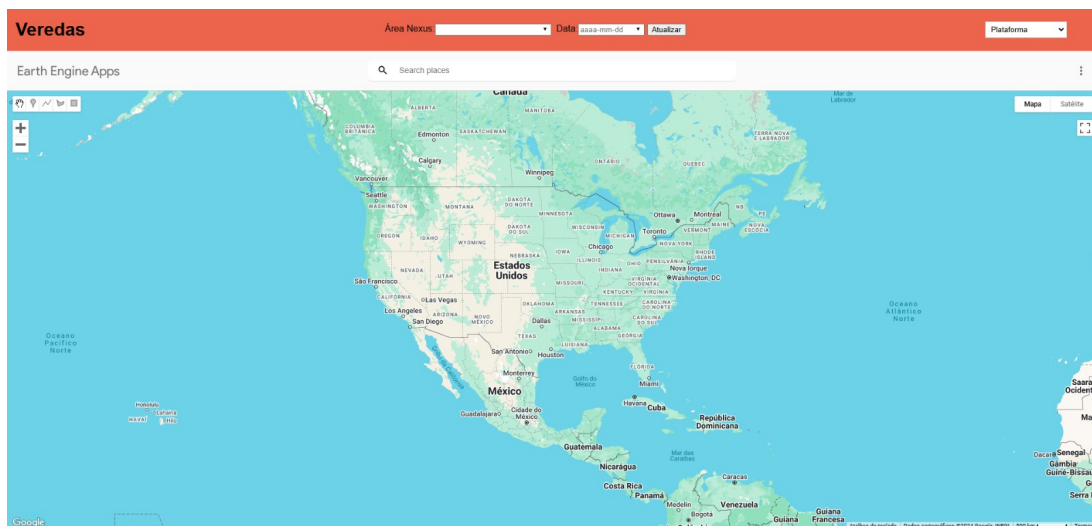
A integração entre HTML, CSS e JavaScript resultou em uma solução robusta e escalável. O uso combinado dessas tecnologias permitiu criar uma plataforma que não apenas atende às necessidades atuais, mas também oferece flexibilidade para futuras atualizações e adaptações. Essa escolha foi orientada pela busca de eficiência, compatibilidade e simplicidade, garantindo que a aplicação pudesse evoluir conforme as demandas do projeto, sem comprometer a estabilidade ou a usabilidade do sistema.

8 CONCLUSÃO

8.1 RESULTADOS

O desenvolvimento deste trabalho resultou na criação de uma plataforma web interativa voltada para a visualização de veredas no Cerrado brasileiro, demonstrando o potencial de integração entre tecnologias avançadas de sensoriamento remoto e análise geoespacial. Apesar de, atualmente, a plataforma operar com imagens fictícias devido à indisponibilidade das imagens reais geradas pela etapa de ciência de dados, ela já se apresenta como uma ferramenta funcional e promissora para a exploração de dados ambientais.

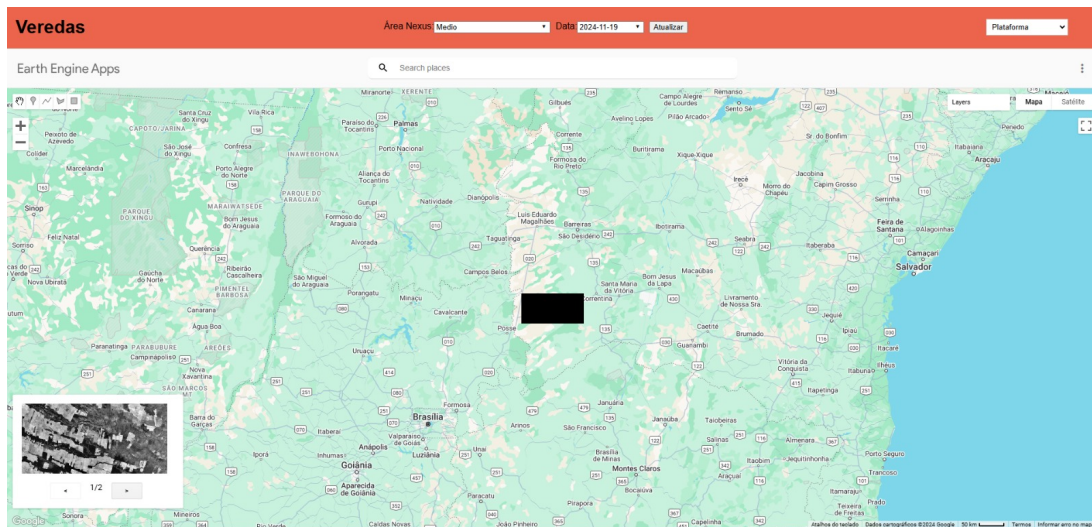
Figura 8.1 – Plataforma de visualização de desenvolvida ao longo do projeto.



Fonte: Elaborado pelos autores.

A plataforma oferece recursos como mapas dinâmicos, que permitem a navegação interativa e a seleção de áreas de interesse por meio de interfaces intuitivas. A funcionalidade de filtragem por áreas e datas, juntamente com a exibição visual de regiões mapeadas, demonstra a viabilidade do sistema para o acesso e análise de informações espaciais, mesmo em sua configuração inicial com dados simulados.

Figura 8.2 – Seleção de veredas na plataforma.



Fonte: Elaborado pelos autores.

Além disso, o projeto foi concebido com um design modular e escalável, utilizando frameworks e tecnologias que permitem a rápida adaptação a novos cenários. Essa arquitetura garante que a plataforma possa facilmente incorporar imagens reais no futuro, sem a necessidade de reestruturações significativas. Essa flexibilidade tecnológica assegura que a solução possa evoluir para atender às demandas de monitoramento ambiental em larga escala.

A navegação intuitiva, os elementos visuais bem estruturados e a responsividade demonstram que a plataforma é capaz de atender a diferentes tipos de usuários, desde pesquisadores até gestores ambientais e formuladores de políticas públicas. Assim, embora a funcionalidade esteja parcialmente limitada pela ausência de dados reais, a plataforma já é capaz de ilustrar seu potencial para transformar dados geoespaciais em informações acessíveis e acionáveis, consolidando uma base sólida para futuras expansões e melhorias.

8.2 CONTRIBUIÇÕES

O presente trabalho apresenta contribuições relevantes em múltiplos aspectos, tanto no âmbito da ciência de dados aplicada ao meio ambiente quanto no desenvolvimento de tecnologias de suporte à gestão ambiental. A plataforma proposta destaca-se como um exemplo prático de integração entre análise geoespacial avançada e visualização interativa, demonstrando um modelo que pode ser expandido para outros biomas e contextos ecológicos.

No campo da ciência de dados, o projeto ilustra como algoritmos avançados

de aprendizado de máquina e ferramentas de sensoriamento remoto, como o Google Earth Engine, podem ser utilizados para tratar grandes volumes de dados geoespaciais. A integração dessas tecnologias permite gerar representações visuais que facilitam a compreensão de fenômenos ambientais complexos. Embora a plataforma atual funcione com dados fictícios, sua estrutura modular e escalável foi projetada para incorporar dados reais, oferecendo uma base sólida para análises futuras mais robustas.

A escolha de ferramentas tecnológicas como serviços de nuvem da AWS, combinada com frameworks de desenvolvimento web, possibilitou a construção de uma aplicação escalável e eficiente. Essa arquitetura assegura não apenas a flexibilidade para futuras expansões, mas também a capacidade de atender a diferentes demandas de usuários, desde acadêmicos e pesquisadores até gestores ambientais e formuladores de políticas públicas. A modularidade do sistema garante que ele possa ser atualizado e adaptado de maneira ágil, permitindo a inclusão de novas funcionalidades e integrações com outras plataformas ou bancos de dados.

Além do avanço técnico, o projeto reforça a importância da tecnologia como uma aliada estratégica na conservação ambiental. A plataforma foi concebida como uma ferramenta acessível, capaz de disseminar dados e informações críticas sobre ecossistemas, contribuindo para a conscientização sobre a preservação ambiental. Ao disponibilizar uma solução voltada para a gestão sustentável, o trabalho também colabora com os esforços de planejamento e desenvolvimento de políticas públicas baseadas em dados.

Por fim, ao unir aspectos de ciência de dados, tecnologia web e preservação ambiental, este trabalho demonstra como soluções interdisciplinares podem ser criadas para enfrentar desafios ambientais complexos. As contribuições apresentadas consolidam a plataforma como uma base inicial robusta, com grande potencial de expansão e impacto em iniciativas de monitoramento e conservação ambiental no Cerrado e em outros biomas.

8.3 TRABALHOS FUTUROS

Embora a plataforma desenvolvida tenha alcançado seu objetivo inicial, ainda existem desafios e oportunidades de aprimoramento que podem expandir significativamente seu impacto e utilidade. A principal melhoria esperada é a integração de imagens reais geradas pela etapa de ciência de dados. Essa integração permitirá que a plataforma transcenda sua funcionalidade atual baseada em dados fictícios, viabilizando análises precisas e fundamentadas sobre a distribuição e evolução das veredas no Cerrado brasileiro. Essa atualização deverá envolver a incorporação de algoritmos de aprendizado profundo ajustados às características das imagens reais, aumentando a confiabilidade dos resultados.

Outra oportunidade de melhoria está na ampliação das funcionalidades da plataforma, incluindo a adição de ferramentas que possibilitem a análise comparativa de dados entre diferentes períodos e regiões. Essa funcionalidade será essencial para identificar tendências de degradação ou recuperação das veredas ao longo do tempo. Além disso, a geração de relatórios customizados, baseados em consultas específicas dos usuários, poderá fornecer informações detalhadas para subsidiar decisões estratégicas de gestores ambientais e pesquisadores.

Para fortalecer a aplicabilidade da plataforma, recomenda-se a integração com bases de dados regionais e globais, como aquelas mantidas por instituições governamentais e organizações internacionais. Essa integração permitirá enriquecer os dados disponíveis, oferecendo uma visão mais abrangente sobre o contexto ambiental das veredas e sua relação com outros fatores, como uso do solo e mudanças climáticas. A conexão com iniciativas governamentais poderá facilitar a adoção da plataforma como ferramenta oficial de monitoramento ambiental, ampliando seu alcance e relevância.

Adicionalmente, melhorias na interface e na experiência do usuário podem ser exploradas, garantindo que a plataforma seja intuitiva e acessível a públicos diversos, incluindo comunidades locais, educadores e estudantes. A introdução de elementos de gamificação e visualizações interativas mais avançadas pode aumentar o engajamento e ampliar o impacto educativo da aplicação.

Por fim, é recomendável explorar parcerias com instituições acadêmicas e de pesquisa para validar os dados e análises geradas pela plataforma, garantindo a precisão científica dos resultados. A implementação de mecanismos de feedback dos usuários também pode orientar futuras iterações, assegurando que a plataforma evolua de forma alinhada às necessidades de seus diferentes públicos. Esses avanços consolidarão a aplicação como uma ferramenta indispensável para o monitoramento, a preservação e o planejamento sustentável das veredas no Cerrado brasileiro.

REFERÊNCIAS

- ARAÚJO, G. M.; BARBOSA, A. A.; ARANTES, A. A.; AMARAL, A. F. Composição florística de veredas no Município de Uberlândia, MG. **Revista Brasileira de Botânica**, 2002.
- BEN-YEHUDA, O. A.; BEN-YEHUDA, M.; SCHUSTER, A.; TSAFRIR, D. Deconstructing amazon ec2 spot instance pricing. **ACM Trans. Econ. Comput.**, Association for Computing Machinery, New York, NY, USA, v. 1, n. 3, sep 2013. ISSN 2167-8375. Disponível em: <<https://doi.org/10.1145/2509413.2509416>>.
- BI, Q.; GOODMAN, K. E.; KAMINSKY, J.; LESSLER, J. What is machine learning? a primer for the epidemiologist. **American Journal of Epidemiology**, 2019.
- CDB. **Convention on Biological Diversity**. 2024. Acessado em 25 de novembro de 2024. Disponível em: <<https://www.cbd.int/countries/profile?country=br>>.
- DINKU, Z. **React.js vs. Next.js**. Dissertação (Bacharelado) — Metropolia University of Applied Sciences, 2022.
- GARFINKEL, S. An Evaluation of Amazon's Grid Computing Services: EC2, S3, and SQS. **Harvard Computer Science Group Technical Report TR-08-07**, 2007.
- GHIMIRE, D. **Comparative study on Python web frameworks: Flask and Django**. Dissertação (Bacharelado) — Metropolia University of Applied Sciences, 2020.
- GIUZIO, C. M. O.; ZOELLER, I. V.; COELHO, R. A. **Estimation of socioeconomic indicators through satellite imagery in the NEXUS area**. Dissertação (Bacharelado) — Escola Politécnica da Universidade de São Paulo, 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**, 2017.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, 2015.
- LIBERTY, E.; KARNIN, Z.; XIANG, B.; ROUESNEL, L.; COSKUN, B.; NALLAPATI, R.; DELGADO, J.; SADOUGHI, A.; ASTASHONOK, Y.; DAS, P.; BALIOGLU, C.; CHAKRAVARTY, S.; JHA, M.; GAUTIER, P.; ARPIN, D.; JANUSCHOWSKI, T.; FLUNKERT, V.; WANG, Y.; GASTHAUS, J.; STELLA, L.; RANGAPURAM, S.; SALINAS, D.; SCHELTER, S.; SMOLA, A. Elastic machine learning algorithms in amazon sagemaker. In: **Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: Association for Computing Machinery, 2020. (SIGMOD '20), p. 731–737. ISBN 9781450367356. Disponível em: <<https://doi.org/10.1145/3318464.3386126>>.
- LIE, H. W. **Cascading Style Sheets**. Tese (Doutorado) — University of Oslo, 2006.
- MEHMOOD, H.; CONWAY, C.; PERERA, D. Mapping of flood areas using landsat with google earth engine cloud platform. **Atmosphere**, 2021.

NEXUS. **Nexus - Caminhos para a Sustentabilidade**. 2024. Acessado em 25 de novembro de 2024. Disponível em: <<https://nexus.ccst.inpe.br/projeto/>>.

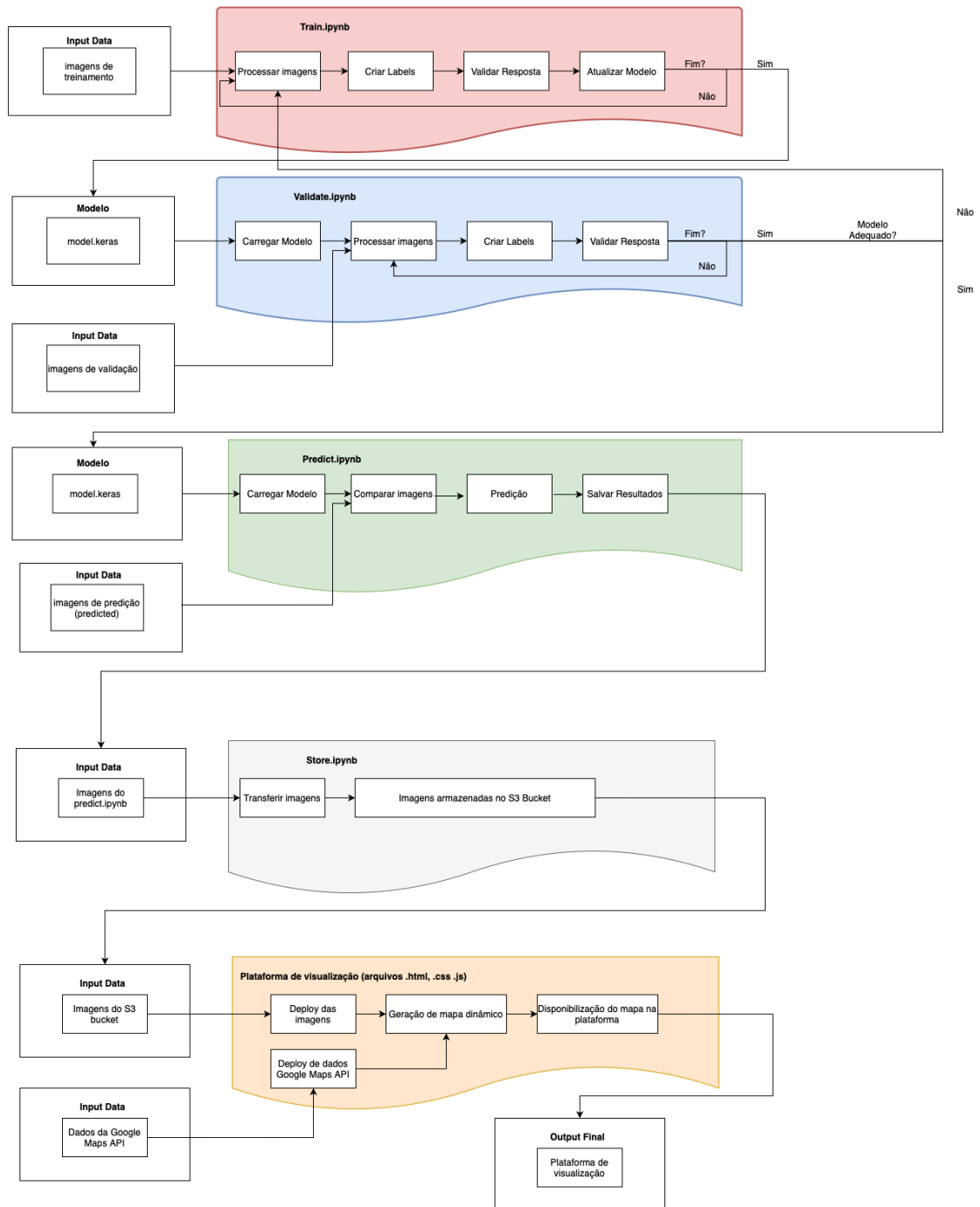
SHETTY, J.; DASH, D.; JOISH, A. K.; C., G. Review Paper on Web Frameworks, Databases and Web Stacks. **International Research Journal of Engineering and Technology (IRJET)**, 2020.

SHINDE, P. P.; SHAH, S. A Review of Machine Learning and Deep Learning Applications. In: **2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)**. [S.l.: s.n.], 2018. p. 1–6.

SINHA, P. Cloud Computing Using aws: An Analysis. **Indian Journal of Computer Science**, 2020.

APÊNDICE A – FLUXO DE TRABALHO DO PROJETO

Figura A.1 – Representação ilustrativa do fluxo de trabalho.



Fonte: Elaborado pelos autores.