

PROTÓTIPO DE SISTEMA DE INFORMAÇÃO GEOGRÁFICA ONLINE OPEN SOURCE

Davi Nesi Destro ¹ Marcel Campos Inocencio ²

Resumo: A análise e interpretação de informações georreferenciadas são fundamentais para o desenvolvimento sustentável e a tomada de decisões em áreas como gestão municipal, meio ambiente e agronegócio. Apesar das soluções robustas oferecidas por SIGs de código aberto, como QGIS e GRASS, sua limitação ao ambiente desktop restringe o acesso e dificulta o compartilhamento de dados. Este trabalho apresenta o desenvolvimento de um protótipo de SIG online de código aberto, permitindo a criação, edição e visualização de informações geográficas diretamente no navegador. Utilizando tecnologias como Django, React, PostGIS e OpenLayers, a solução adota uma abordagem iterativa com testes de usabilidade para garantir eficiência, acessibilidade e desempenho. Os resultados demonstram um sistema funcional e escalável, capaz de atender às necessidades de manipulação e compartilhamento de dados geoespaciais. A interface intuitiva e responsiva facilita o uso por diversos públicos, com potencial para futuras melhorias, como suporte a dados raster e funcionalidades analíticas mais avançadas.

Palavras-chave: Sistema de Informação Geográfica, SIG online, PostGIS, Open Source, OpenLayers.

¹davi_nesi@unesb.net

²marcel.inocencio@gmail.com

ABSTRACT: The analysis and interpretation of georeferenced information are fundamental for sustainable development and decision-making in areas such as municipal management, environmental conservation, and agribusiness. Despite the robust solutions provided by open-source GIS tools like QGIS and GRASS, their limitation to desktop environments restricts access and complicates data sharing. This study presents the development of an open-source online GIS prototype, enabling the creation, editing, and visualization of geographic information directly through a web browser. Leveraging technologies such as Django, React, PostGIS, and OpenLayers, the solution employs an iterative approach with usability testing to ensure efficiency, accessibility, and performance. The results demonstrate a functional and scalable system capable of addressing the needs of geospatial data manipulation and sharing. Its intuitive and responsive interface accommodates a diverse user base, with potential for future enhancements, such as support for raster data and advanced analytical functionalities.

Keywords: Geographic Information Systems, SIG online, PostGIS, Open Source, OpenLayers.

1 INTRODUÇÃO

Reconhece-se amplamente que a capacidade de gerar, analisar e compreender informações georreferenciadas é um dos elementos mais relevantes para o desenvolvimento dos países. Existe um consenso de que a informação é um recurso estratégico fundamental para orientar decisões (Máximo, 2004).

Neste contexto, no Brasil, observa-se uma demanda crescente pela incorporação de geoprocessamento, principalmente nas áreas de gestão municipal, meio ambiente e agronegócio (Rosa, 2011).

Na gestão municipal, o crescimento acelerado das cidades exige uma administração eficiente do espaço urbano e a otimização de recursos, melhorando a qualidade de vida dos cidadãos (Lima; Lopes; Façanha, 2019).

No meio ambiente, a necessidade de monitorar e preservar vastas áreas florestais, como a Amazônia, e de gerenciar recursos naturais de forma sustentável, torna essencial o uso de tecnologias de geoprocessamento (Cunha, 2013).

No agronegócio, a competição global e a busca por aumento de produtividade impulsionam a adoção de agricultura de precisão e a otimização do uso de insumos agrícolas (Cunha, 2013).

A consolidação desse movimento depende de diversos fatores, incluindo o acesso a dados georreferenciados relevantes, a disponibilidade de softwares licenciados, além da capacitação de profissionais e o contínuo desenvolvimento de técnicas para análise espacial (Barcelos, 2002).

Um SIG (Sistema de informação geográfica) é composto por um conjunto de ferramentas especializadas que desempenham funções específicas, como a aquisição, armazenamento, recuperação, transformação e emissão de informações georreferenciadas (Câmara, 2002).

Esses elementos são essenciais para permitir a utilização efetiva do geoprocessamento como uma ferramenta estratégica no planejamento e na tomada de decisões, contribuindo assim para o desenvolvimento sustentável do país (Câmara, 2002).

Os principais SIG's disponíveis de forma gratuita no mercado são o QGIS e o Grass, no entanto estes sistemas são disponibilizados apenas para desktop, o que obriga seus usuários a baixarem e instalarem a ferramenta, restringindo assim seu acesso, fazendo que suas informações fiquem armazenadas no disco do sistema, dificultando o compartilhamento entre diferentes usuários (Furquim; Furquim, 2007).

Além disso, ao manter as informações localmente, existe o risco de perda de dados em caso de falha no disco rígido. Sem um backup eficiente os usuários estão vulneráveis a perda de seu trabalho, o que pode resultar em interrupções e prejuízos financeiros (Fernandes; Fernandes, 2011).

Portanto, o objetivo desta pesquisa é facilitar a visualização e manipulação de informações georreferenciadas por meio de um protótipo online. Visando estabelecer uma base comparativa com estudos já publicados, outros trabalhos foram considerados para avaliar os resultados obtidos.

O projeto baseado em JavaServer Faces (JSF) proposto por Wallick (2017) visa simplificar o desenvolvimento de sistemas de informações geográficas, integrando OpenLayers 3 e GeoServer, com foco em abstrair a complexidade do uso de JavaScript e facilitar o trabalho dos desenvolvedores. Por outro lado, o sistema de informações geográficas web desenvolvido, também utilizando OpenLayers, oferece uma abordagem mais interativa, permitindo que os usuários visualizem, editem e compartilhem informações geográficas diretamente no ambiente web. Enquanto o framework JSF prioriza a modularidade e a reutilização de componentes, o SIG web proposto coloca ênfase na experiência do usuário, possibilitando a

manipulação de dados geoespaciais de forma mais dinâmica. Além disso, o uso de PostGIS, em vez de GeoServer, permite a realização de operações geoespaciais complexas, como cálculos de distâncias e interseções, diretamente no banco de dados, otimizando o desempenho da aplicação.

O projeto Autarquia Livre de Henriques (2016) estruturado em uma arquitetura de três camadas e baseado em ferramentas open source como GeoServer e OpenLayers, visa oferecer uma alternativa flexível e acessível para SIGs em administrações locais, promovendo autonomia tecnológica e interoperabilidade com sistemas geoespaciais de padrões abertos. Em comparação, o SIG web proposto adota um framework moderno e escalável, com foco na performance e integração, favorecendo o processamento avançado de dados geoespaciais e uma interface dinâmica e interativa. Enquanto o Autarquia Livre prioriza a replicabilidade entre municípios, o SIG web proposto concentra-se em fornecer uma solução mais robusta e adaptável.

A dissertação de Gomes (2013) apresenta o SIAGF, um sistema que centraliza dados geoespaciais para apoiar a gestão florestal, utilizando tecnologias open source e priorizando ferramentas simples, acessíveis a usuários com pouca experiência em SIG, como gestores florestais e proprietários. Em contraste, o SIG web proposto adota uma abordagem modular e escalável, visando otimizar a experiência do usuário em diversos contextos e dispositivos. Ambos utilizam o OpenLayers para visualização de mapas interativos; porém, enquanto o SIAGF facilita o acesso direto à informação florestal e à interação com gestores, o SIG web prioriza performance e capacidade de adaptação a cenários mais complexos e customizáveis.

2 MATERIAIS E MÉTODOS

A pesquisa é de natureza aplicada, tecnológica e explicativa. O desenvolvimento do aplicativo foi realizado em um computador com 32GB de RAM, processador Ryzen 5 5600X (6 núcleos, 12 threads, 3.7GHz) e placa de vídeo NVIDIA RTX 3060 Ti de 8GB, rodando o sistema operacional Linux.

Para o desenvolvimento do código, foi utilizada a IDE Visual Studio Code, amplamente adotada por desenvolvedores de software. Suas funcionalidades, como autocompletar e indentação automática, facilitaram e agilizaram o processo de desenvolvimento, mantendo o código organizado.

A escolha das tecnologias Python com Django, Go para requisi-

ção de vector tiles, e React combinado com OpenLayers resultou em uma arquitetura robusta e eficiente para a aplicação.

Django, o framework web de alto nível para Python, foi selecionado para gerenciar a lógica de backend e o processamento dos dados geoespaciais. Com sua capacidade de integração eficiente com bibliotecas geoespaciais GeoDjango e GDAL, Django oferece uma infraestrutura sólida para a manipulação e armazenamento de dados geográficos. Além disso, sua estrutura modular e organizada facilita o desenvolvimento de uma API consistente e segura para gerenciar as interações com os dados geoespaciais.

Go foi utilizado para o gerenciamento e recuperação dos vector tiles, que são essenciais para a renderização eficiente de mapas. O desempenho altamente otimizado de Go é vantajoso para o processamento rápido e simultâneo de requisições de tiles, o que reduz o tempo de resposta e melhora a fluidez da navegação pelo mapa, especialmente com grandes volumes de dados geográficos e requisições simultâneas.

Na camada de frontend, o uso de React e OpenLayers possibilitou uma experiência de usuário avançada e interativa para visualização e manipulação de dados geoespaciais. OpenLayers foi integrado para gerenciar e exibir mapas de forma detalhada e interativa, oferecendo suporte a operações avançadas de visualização e customização de dados geográficos. Essa combinação permite que a aplicação apresente dados de forma fluida, com alta capacidade de resposta, oferecendo uma experiência de usuário intuitiva e altamente informativa.

Além das tecnologias escolhidas, o uso do PostGIS como extensão do banco de dados PostgreSQL acrescenta uma camada essencial de capacidade espacial à aplicação. PostGIS é amplamente reconhecido por sua eficiência e robustez no armazenamento, manipulação e consulta de dados geoespaciais, tornando-o uma escolha natural para aplicações que lidam com grandes volumes de informações espaciais.

PostGIS permite que dados geoespaciais sejam armazenados diretamente no banco de dados, oferecendo suporte a tipos de dados geométricos e funções espaciais avançadas. Através de suas funções, é possível realizar operações como cálculos de distâncias, interseções e buffer diretamente no banco de dados, o que otimiza a performance ao minimizar a necessidade de processamento adicional no backend.

No contexto desta aplicação SIG, Django é configurado para integração ao PostGIS, por meio do GeoDjango, o que permite que consultas

complexas e manipulações espaciais sejam executadas de forma mais eficiente. Essa integração entre Django e PostGIS facilita o desenvolvimento, fornecendo dados prontos para serem consumidos pela aplicação, sem a necessidade de complexas operações intermediárias.

Com PostGIS, Django, Go e o frontend em React com Next.js e OpenLayers, a aplicação atinge um alto nível de desempenho e interatividade. Essa arquitetura integrada facilita o gerenciamento de dados espaciais complexos, proporcionando ao usuário final uma experiência rica e eficiente na exploração e análise de informações geoespaciais.

2.1 PROCESSO DE DESENVOLVIMENTO

Foi adotada uma abordagem iterativa e flexível no desenvolvimento do sistema, permitindo ajustes rápidos e frequentes conforme novas necessidades surgiam. Essa metodologia garantiu a entrega contínua de resultados funcionais, mantendo o foco na agilidade e na evolução constante do projeto.

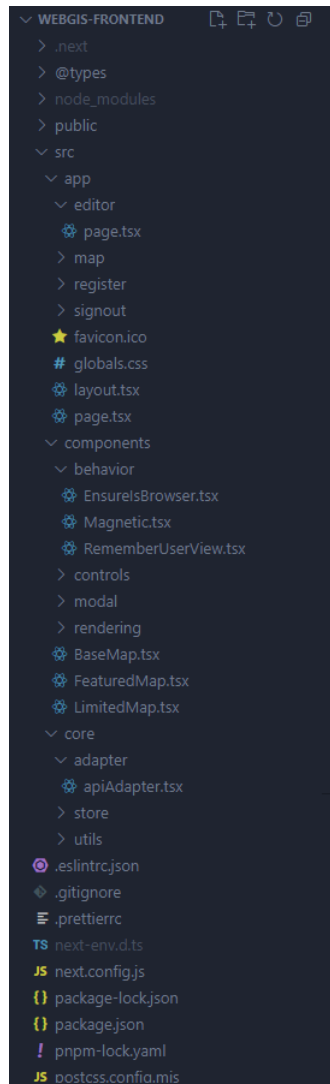
A continuidade do projeto foi conduzida de forma ágil e prática, com a criação de um protótipo inicial diretamente no ambiente de desenvolvimento. As etapas de testes e correção de erros ocorreram em paralelo ao desenvolvimento, sem uma divisão rígida entre as fases. Isso permitiu ajustes e melhorias conforme o projeto avançava, garantindo flexibilidade e a entrega contínua de resultados funcionais.

2.1.1 FRONT-END

No front-end do projeto, o desenvolvimento se deu por meio do meta-framework Next.js, que combina o potencial do React com a tipagem estática fornecida pelo TypeScript. Esta abordagem aumenta a confiabilidade do código e facilita futuras manutenções. A estrutura do código-fonte é organizada em diferentes subdiretórios, cada um com uma funcionalidade específica, o que melhora a modularidade e a escalabilidade da aplicação.

Na Figura 1 o diretório projeto front-end é representado por meio do software de edição Visual Studio Code. Nele é possível observar a organização das pastas.

Figura 1 - Diretório front-end



Fonte: Elaborado pelo autor.

O diretório `src` serve como a raiz para todo o código não gerado pelo framework, enquanto `src/app` contém os subdiretórios que representam as rotas principais e páginas da aplicação, como a edição de mapas (`editor`), registro de usuário (`register`) e logout (`signout`). Essa organização facilita a navegação do usuário e mantém um fluxo de interação consistente.

No diretório `components`, são encontrados componentes reutilizáveis que são consumidos em várias páginas, aumentando a coesão e reusabilidade do código. O subdiretório `behavior` concentra componentes que tratam de comportamentos específicos de interação com o usuário, enquanto `controls` implementa controles de interface que auxiliam na manipulação do mapa e suas camadas, e `rendering` agrupa componentes dedi-

cados à renderização visual.

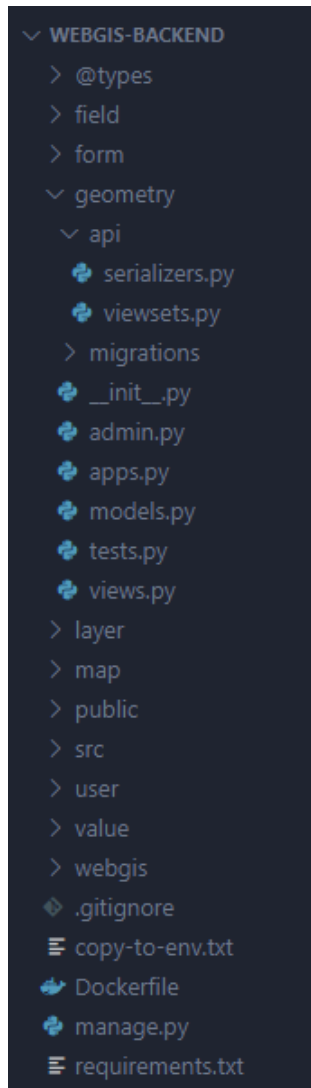
A pasta core abriga scripts e funções principais da aplicação, incluindo o adapter, responsável pela adaptação de dados e configuração das APIs para uma comunicação consistente com o backend. A subpasta store implementa uma arquitetura de estado centralizada, controlando os dados da aplicação e funcionalidades de mapa, enquanto utils contém funções utilitárias para abstrair tarefas repetitivas, como debounce para otimizar chamadas de função e gettoken para gerenciar o acesso ao token de autenticação, essencial para segurança e controle de acesso.

2.1.2 BACK-END

A implementação do projeto foi realizada utilizando o Django como framework principal. A organização modular do código é dividida em diretórios que representam cada componente principal do sistema: field, form, geometry, layer, map, user, value, facilitando a manutenção e a escalabilidade do projeto. O diretório api concentra a lógica de interação da aplicação com o Django Rest Framework, responsável por estruturar a interface de programação de aplicações que permite a comunicação entre o cliente e o servidor.

Na Figura 2 o diretório projeto back-end é representado por meio do software de edição Visual Studio Code. Nele é possível observar a organização das pastas.

Figura 2 - Diretório back-end



Fonte: Elaborado pelo autor.

Dentro de api, os arquivos `serializers.py` e `viewsets.py` contêm a definição de serializadores e viewsets, que são essenciais para transformar as instâncias dos modelos em formatos JSON, adequados para transferência de dados via HTTP, e para organizar as operações CRUD nos dados de forma padronizada. O arquivo `models.py` define a estrutura de dados do banco em modelos Django, refletindo a arquitetura relacional utilizada.

As migrações, localizadas no subdiretório `migrations`, são geradas para aplicar alterações na estrutura do banco de dados conforme o projeto evolui, garantindo integridade e consistência dos dados. Arquivos como `admin.py` e `apps.py` configuram a integração da aplicação com o pai-

nel administrativo do Django e a inicialização do módulo, respectivamente. O DjangoRestFramework (DRF) permite que as funcionalidades de manipulação de dados espaciais e a aplicação de regras de negócio sejam expostas de forma segura e eficiente.

2.1.3 BACK-END VECTOR TILE

Desenvolvido em Go, o serviço recupera tiles vetoriais, permitindo a visualização eficiente de dados geoespaciais. Cada tile é definido por coordenadas X, Y e um nível de zoom Z, especificando uma área do mapa que será representada no formato MVT. O sistema converte essas coordenadas em uma área delimitadora, realizando consultas no banco de dados para extrair as geometrias que intersectam essa área. O resultado é um tile vetorial que é retornado ao cliente, permitindo a renderização interativa e segmentada do mapa. O serviço expõe essa funcionalidade por meio de um endpoint RESTful que utiliza parâmetros de rota para especificar a camada e as coordenadas do tile, enquanto um middleware CORS garante que o sistema possa ser acessado de diversas origens, facilitando sua integração com aplicações web de mapas dinâmicos.

2.1.4 DEVOPS

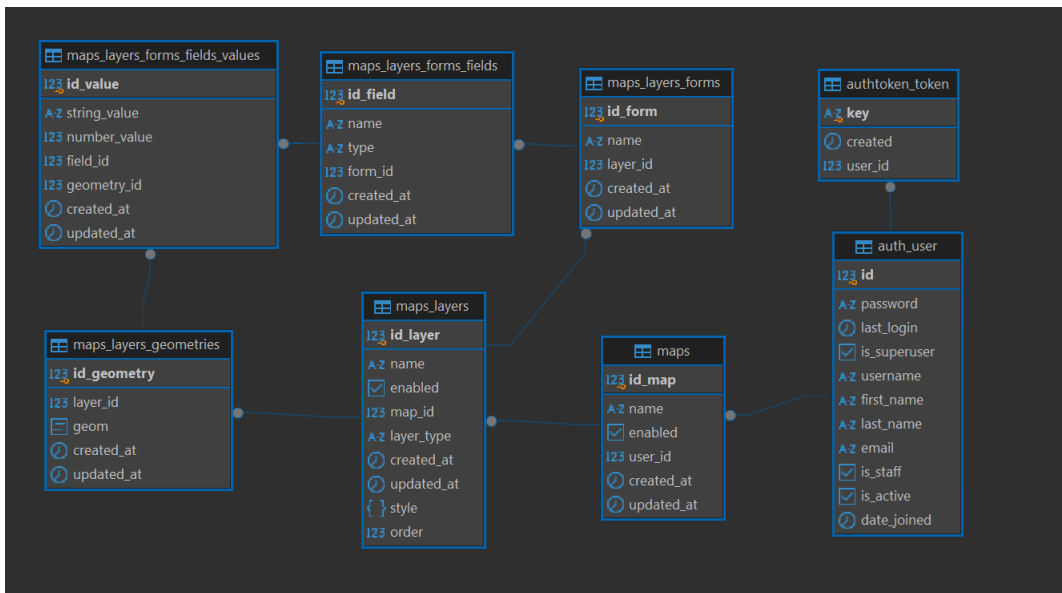
A arquitetura do projeto back-end em Django e o serviço em Go para recuperação de tiles vetoriais, foi containerizada utilizando Docker e distribuído em uma máquina virtual privada, facilitando a implantação e escalabilidade do sistema. Esses serviços estão organizados em contêineres separados, garantindo isolamento e gerenciamento eficiente dos recursos. Para centralizar o acesso e melhorar a segurança, o serviço de Nginx atua como proxy reverso, redirecionando as requisições para os serviços corretos com base no endpoint requisitado. O Nginx distribui o tráfego entre o backend em Django e o serviço de tiles em Go.

O front-end é hospedado e distribuído pela plataforma Vercel, garantindo uma entrega rápida e confiável para os usuários. Essa escolha facilita o deployment contínuo do front-end, permitindo atualizações rápidas e um ambiente de desenvolvimento ágil. Integrado ao backend containerizado, o front-end na Vercel interage com os serviços de backend em Django e o serviço de tiles vetoriais em Go, proporcionando uma interface fluida e responsiva para o usuário final, com alto desempenho e escalabilidade em um ambiente distribuído.

2.1.5 BANCO DE DADOS

Na Figura 3 o banco de dados do projeto é representado em um diagrama e organiza-se em uma estrutura relacional que permite a gestão de mapas, camadas, geometrias e usuários de forma integrada.

Figura 3 - Diagrama banco de dados



Fonte: Elaborado pelo autor.

A tabela principal de mapas armazena dados como o nome do mapa, estado de ativação e referência ao usuário criador, servindo como ponto de ancoragem para as camadas associadas. Cada camada, identificada individualmente, possui informações sobre tipo, estilo e ordem de exibição, além de uma referência ao mapa a que pertence, garantindo a hierarquia entre mapas e suas respectivas camadas. As geometrias, associadas a camadas específicas, contêm a estrutura geométrica dos dados espaciais, sendo fundamentais para o mapeamento espacial no sistema.

Para a coleta de dados específicos, o banco permite a criação de formulários para cada camada, possibilitando a adição de campos configuráveis. Estes campos são representados na tabela de valores, que armazena as informações preenchidas pelos usuários, associando-as tanto ao campo de entrada quanto à geometria correspondente. Esse esquema permite a coleta e organização de dados de forma precisa e contextualizada com as camadas e geometrias no ambiente.

A segurança e o controle de acesso são gerenciados pela tabela

de usuários, que armazena informações dos perfis cadastrados, e pela tabela de tokens de autenticação, essencial para o controle de sessão e validação de acesso. Desta forma, essa estrutura oferece uma base robusta e escalável para o funcionamento, facilitando o armazenamento e a manipulação de dados espaciais e atributivos de forma segura e eficiente.

2.2 INTERFACE DA APLICAÇÃO

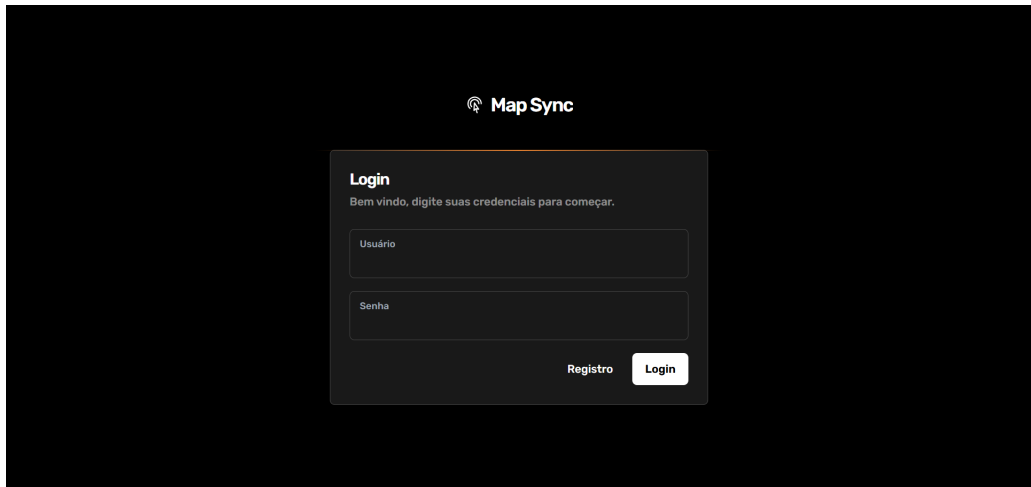
Para alcançar o objetivo de construir uma ferramenta que permita aos usuários criar, visualizar e compartilhar informações geográficas, o projeto foi desenvolvido como uma aplicação web no formato de Single Page Application. A escolha desse modelo se baseou principalmente na acessibilidade, permitindo que os usuários acessem a plataforma de qualquer dispositivo com um navegador, sem a necessidade de instalações adicionais.

No desenvolvimento da aplicação, foi priorizada a criação de uma interface intuitiva e de fácil utilização, voltada para a manipulação de dados geoespaciais. Para isso, reduziu-se o número de telas e interações, simplificando a navegação e permitindo que os usuários visualizassem e editassem informações diretamente no mapa.

Como resultado, a aplicação foi desenvolvida com seções específicas, como telas de autenticação, uma tela principal com o mapa contendo camadas geográficas e suas geometrias, além de ferramentas para edição de dados geoespaciais. A aplicação também conta com barras laterais que oferecem funcionalidades adicionais e visualização detalhada de dados das camadas. Essas funcionalidades foram integradas para maximizar a eficiência na manipulação de dados geoespaciais e aprimorar a experiência do usuário.

A tela de login exibida na Figura 4 permite que o usuário acesse sua conta no aplicativo, oferecendo os campos de e-mail e senha. Também há um botão que direciona para a opção de cadastro, caso o usuário ainda não tenha uma conta.

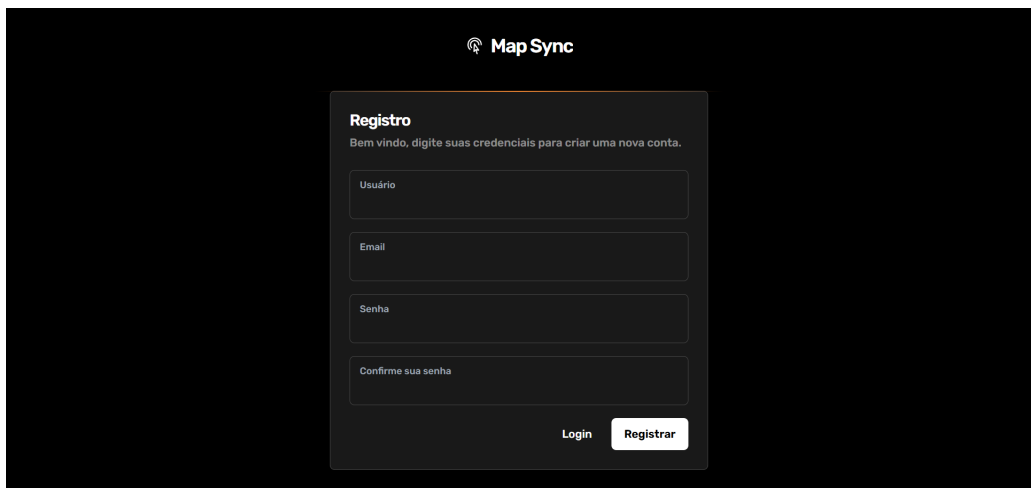
Figura 4 - Tela de login



Fonte: Elaborado pelo autor.

A tela de cadastro, na Figura 5, permite que novos usuários se registrem. Ela contém campos para nome de usuário, e-mail, senha e confirmação de senha. Após pressionar o botão de cadastro, se os dados forem validados, o usuário é registrado e redirecionado para a tela principal.

Figura 5 - Tela de cadastro

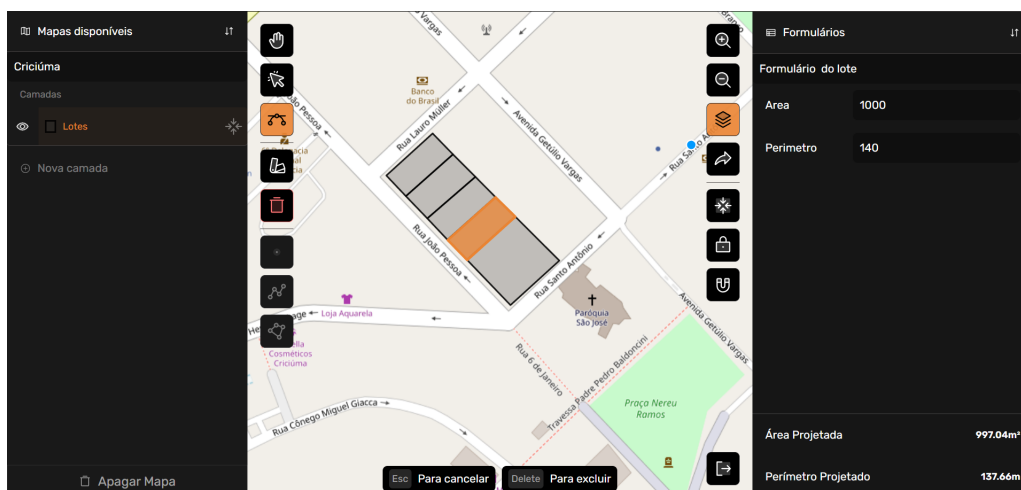


Fonte: Elaborado pelo autor.

A tela principal da aplicação representada na Figura 6 permite aos usuários visualizar e interagir com um mapa. Sobreposto à esquerda, há um painel contendo as ferramentas responsáveis pelo gerenciamento e controles de visualização das camadas do mapa, permitindo que o usuário

habilite, desabilite, crie ou remova camadas. Enquanto o painel a direita fica com a responsabilidade de gerenciar os formulários, atributos e seus valores.

Figura 6 - Tela principal



Fonte: Elaborado pelo autor.

A interface do mapa apresentada conta com ferramentas organizadas na lateral esquerda, que permitem uma interação ampla com os elementos geográficos. Essas ferramentas incluem opções para edição, personalização de camadas e exclusão das mesmas, oferecendo controle direto sobre os dados exibidos no mapa.

O mapa central é interativo, facilitando a visualização detalhada, edição e exploração de informações geográficas em tempo real, permitindo que o usuário navegue pelos dados de forma intuitiva e dinâmica. À direita, encontram-se opções adicionais para controlar a visualização e comportamento do mapa, como ajustes ao campo de visão, compartilhamento e encaixe de cursor.

Com essa organização funcional e visual, a interface proporciona uma experiência prática e eficiente, permitindo a manipulação de dados geoespaciais diretamente no navegador, com agilidade e facilidade. A aplicação é compatível com navegadores amplamente utilizados, como Google Chrome, Mozilla Firefox, Microsoft Edge e Safari, garantindo acessibilidade para diversos usuários.

2.3 ARMAZENAMENTO DE DADOS

O armazenamento de dados geoespaciais foi um aspecto crítico no desenvolvimento da aplicação. A escolha de uma tecnologia segura e robusta foi essencial para garantir a integridade, escalabilidade e segurança das informações.

A autenticação de usuários foi realizada por meio do sistema integrado do Django, garantindo que dados sensíveis, como e-mails e senhas, fossem armazenados de forma segura e criptografada, seguindo as melhores práticas de segurança. Isso assegurou a proteção e privacidade dos dados confidenciais dos usuários.

2.4 FUNCIONALIDADES DA APLICAÇÃO

A aplicação desenvolvida tem como principal objetivo permitir que os usuários visualizem, editem e compartilhem informações geográficas de forma interativa. A manipulação de dados geoespaciais é feita diretamente no navegador, através de um sistema de mapas interativos que exibe camadas vetoriais de informações.

O compartilhamento dessas informações é facilitado pela visualização no mapa, que permite aos usuários acessar detalhes de cada camada ou geometria. Após a inclusão ou atualização de dados, o sistema reflete as modificações no mapa principal, onde cada geometria pode ser selecionada para exibir detalhes, como atributos espaciais e temporais, além de oferecer opções de edição ou exclusão diretamente na interface.

Essas funcionalidades promovem a colaboração entre usuários que precisam compartilhar ou acessar informações geoespaciais de maneira prática e eficiente, utilizando uma interface visual intuitiva e interativa.

3 RESULTADOS E DISCUSSÃO

A partir da conclusão do desenvolvimento do Sistema de Informação Geográfica Online, constatou-se que as escolhas tecnológicas e de plataformas foram adequadas e eficientes para os objetivos do projeto.

Para avaliar o desempenho do sistema, foram realizados testes específicos que mediram o tempo de resposta para o carregamento de dados e a execução de consultas espaciais em diferentes cenários. Os testes consideraram fatores como o volume de dados geoespaciais, a complexidade das operações e as condições de infraestrutura, incluindo largura de banda da internet e carga no servidor.

Os resultados estão sintetizados na Tabela 1, onde são apresen-

tados os tempos médios de resposta para diferentes tipos de operações:

Tipo de Operação	Volume de Dados	Tempo Médio de Resposta (ms)	Condições de Teste
Consulta simples	<1.000 registros	100	Rede estável; servidor em baixa carga
Consulta complexa	10.000 registros	300	Rede estável; servidor em média carga
Renderização de camadas	5 camadas simultâneas	200	Rede estável; servidor em baixa carga
Consulta em alta carga	>100.000 registros	500	Rede instável; servidor em alta carga

Tabela 1: Resultados dos testes de desempenho

Os resultados indicam que o sistema mantém uma experiência fluida e responsiva na maioria das operações, com tempos de resposta variando de 100ms a 500ms, dependendo da complexidade e do volume de dados. Em operações mais intensivas, a largura de banda da internet e a carga no servidor mostraram impacto significativo no desempenho.

A integração com o PostGIS permitiu realizar operações geoespaciais avançadas, como interseções de polígonos e cálculos de distâncias, diretamente no banco de dados, otimizando o desempenho e simplificando a manipulação de dados. A visualização de camadas no mapa, através do OpenLayers, foi eficiente, garantindo uma boa experiência de usuário.

4 CONCLUSÃO

Com base no desenvolvimento e nos resultados obtidos, pode-se afirmar que o protótipo do Sistema de Informação Geográfica online atendeu de forma satisfatória aos objetivos propostos. A combinação de tecnologias como PostgreSQL, PostGIS, Django, Go e OpenLayers demonstrou ser eficaz na construção de uma plataforma robusta e escalável, capaz de manipular grandes volumes de dados geoespaciais e fornecer respostas rápidas a consultas espaciais complexas.

A aplicação permitiu aos usuários realizar operações de visualização, edição e compartilhamento de informações geográficas de maneira intuitiva e eficiente, contribuindo para a acessibilidade e colaboração. A abordagem iterativa no desenvolvimento garantiu flexibilidade na implementação de funcionalidades e na resolução de problemas.

Com base nos conhecimentos adquiridos, algumas recomendações podem ser feitas para futuros projetos. Primeiramente, seria benéfico otimizar o sistema para dispositivos de baixo desempenho. Apesar de o sistema ter funcionado bem em dispositivos mais modernos, há espaço para melhorar o uso de memória e processamento, aprimorando o desempenho em dispositivos com recursos limitados.

Além disso, a expansão de funcionalidades de análise espacial pode aumentar a utilidade do sistema para setores específicos, como agri-

cultura de precisão e planejamento urbano. Adicionar ferramentas avançadas, como análise de proximidade, criação de rotas e simulações geoespaciais, enriqueceria o sistema e o tornaria ainda mais versátil.

Por fim, a inclusão de suporte para dados raster, como imagens de satélite e mapas de calor, tornaria a aplicação mais flexível e aplicável em uma gama ainda maior de contextos, ampliando as possibilidades de uso em diversas áreas que exigem dados geoespaciais de diferentes formatos.

REFERÊNCIAS

CUNHA, B. P. d. **Sustentabilidade ambiental: estudos jurídicos e sociais**. [S.l.]: Educs, 2013.

CÂMARA, G. **Sistemas de informação geográfica para aplicações ambientais e cadastrais: Uma visão geral**. 2002. Disponível em: <<https://www.dpi.inpe.br/gilberto/papers/analise.pdf>>.

FERNANDES, R. R.; FERNANDES, A. P. L. M. **Gestão na Segurança de Dados Adotado no Instituto de Tecnologia em Informática e Informação do Estado de Alagoas – Itec**. 2011. Disponível em: <<https://www.aedb.br/seget/arquivos/artigos11/32614346.pdf>>.

FURQUIM, A.; FURQUIM, M. **Principais características e diferenças entre sistemas sig desktop e sig web**. Esteio, 2007. Disponível em: <https://www.esteio.com.br/downloads/2007/SIG-Desktop_e_SIG-Web.pdf>.

GOMES, P. M. M. Desenvolvimento de um sistema de informação e apoio à gestão florestal baseado em tecnologia open source. Disponível em: <<http://hdl.handle.net/10348/2810>>.

HENRIQUES, P. M. F. **Desenvolvimento de uma solução WebSIG Opensource: autarquia livre**. Tese (masterThesis), jun. 2016. Disponível em: <<https://run.unl.pt/handle/10362/18347>>.

LIMA, S. M. S. A.; LOPES, W. G. R.; FAÇANHA, A. C. **Desafios do planejamento urbano na expansão das cidades: entre planos e realidade**. 2019. e20180037 p. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2175-33692019000100253&lng=pt>.

MÁXIMO, A. A. **A importância do mapeamento da criminalidade utilizando-se tecnologia de sistema de informação geográfica para auxiliar a segurança pública no combate à violência**. 2004.

ROSA, R. O uso de tecnologias de informação geográfica no Brasil. **Revista Geográfica de América Central**, v. 2, n. 47E. ISSN 2215-2563. Disponível em: <<https://www.revistas.una.ac.cr/index.php/geografica/article/view/2018>>.

WALTICK, E. F. Protótipo de um framework de componentes javaserver faces para criação de sistemas de informações geográficas utilizando openlayers 3. Disponível em: <<http://tcc.kironunes.net.br/arquivos/trabalhos/497.pdf>>.