

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
TRIÂNGULO MINEIRO – *CAMPUS* PARACATU**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**SARAH LEAL FRITSCHÉ**

**PROTEGE MEU CERRADO: PLATAFORMA WEB PARA DENÚNCIA  
DE CRIMES AMBIENTAIS E MONITORAMENTO DO CERRADO**

**PARACATU - MG  
2025**

**SARAH LEAL FRITSCHÉ**

**PROTEGE MEU CERRADO: PLATAFORMA WEB PARA DENÚNCIA  
DE CRIMES AMBIENTAIS E MONITORAMENTO DO CERRADO**

Trabalho de Conclusão de Curso  
apresentado ao curso Tecnologia em  
Análise e Desenvolvimento de Sistemas,  
do Instituto Federal de Educação, Ciência  
e Tecnologia do Triângulo Mineiro,  
Campus Paracatu, como parte dos  
requisitos para obtenção do título de  
Analista de Sistemas.

Orientador: Prof. Me. Josimar Viana Silva

**PARACATU - MG  
2025**

Ficha Catalográfica elaborada pelo Setor de Referência do IFTM –  
Campus Paracatu

F919p Sarah Leal Fritsche -

Protege Meu Cerrado: Plataforma Web para denúncia de crimes ambientais e monitoramento do cerrado / Sarah Leal Fritsche - 2025.  
67 f. : il.

Orientador: Josimar Viana Silva.

Trabalho de conclusão de curso (graduação) – Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Paracatu, MG, 2025.

1. Software para denúncia ambiental. 2. Frontend utilizando Next.js. 3. Tecnologia web. I. Josimar Viana Silva. II. Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro - Campus Paracatu. III. Título.

CDD 005

## **TERMO DE APROVAÇÃO**

**SARAH LEAL FRITSCHÉ**

### **PROTEGE MEU CERRADO: PLATAFORMA WEB PARA DENÚNCIA DE CRIMES AMBIENTAIS E MONITORAMENTO DO CERRADO**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Campus Paracatu, como exigência para obtenção do diploma de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Me. Josimar Viana Silva.

Aprovado em 19 de março de 2025.

IFTM (Prof. Me. Josimar Viana Silva - Orientador)

---

IFTM (Prof. Dr. Claiton Luiz Soares)

---

IFTM (Prof. Me. Pedro Henrique Tomás)

---

**PARACATU - MG  
2025**



JOSIMAR VIANA SILVA  
PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO



Documento assinado eletronicamente por JOSIMAR VIANA SILVA, PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO, em 28/03/2025, às 11:14, conforme horário oficial de Brasília, com fundamento no art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).

CLAITON LUIZ SOARES  
PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO



Documento assinado eletronicamente por CLAITON LUIZ SOARES, PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO, em 28/03/2025, às 17:26, conforme horário oficial de Brasília, com fundamento no art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).

PEDRO HENRIQUE TOMÁS  
PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO



Documento assinado eletronicamente por PEDRO HENRIQUE TOMÁS, PROFESSOR DO ENSINO BÁSICO, TÉCNICO E TECNOLÓGICO, em 28/03/2025, às 18:41, conforme horário oficial de Brasília, com fundamento no art. 4º, § 3º, do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site <https://iftm.edu.br/autenticacao/> informando o código verificador **7C7E896** e o código CRC **45009B32**.

Referência: NUP: 23203.002342/2025-40

DOCS nº 0000705592

Dedico este Trabalho de Conclusão de Curso, com profunda gratidão, aos meus pais, Chris Manuel Fritsche e Talita Roquete Leal Fritsche. Desde o início da minha jornada acadêmica até a conclusão desta graduação, seu apoio incondicional e presença constante foram fundamentais, proporcionando-me força e motivação para seguir em frente.

## **AGRADECIMENTOS**

Gostaria de expressar minha profunda gratidão a todas as pessoas que contribuíram para a realização deste trabalho. Primeiramente, agradeço à minha família, em especial aos meus pais, que sempre foram a base de todo o meu apoio e incentivo. A sua presença constante, com amor e confiança, foi essencial para que eu superasse todos os desafios ao longo dessa jornada. Agradeço imensamente por estarem ao meu lado, acreditando em mim e me ajudando a seguir em frente.

Agradeço também aos meus professores e à instituição, que proporcionaram um ambiente de aprendizado e crescimento ao longo da minha trajetória acadêmica. A dedicação e os ensinamentos dos professores foram fundamentais para o desenvolvimento deste trabalho e para o meu crescimento pessoal e profissional. Por fim, quero agradecer a todos que estiveram comigo e me acompanharam durante essa jornada. Seja com palavras de incentivo, conselhos, ou mesmo pela presença, cada um de vocês teve um papel importante na realização deste trabalho. Muito obrigada por todo o apoio e por estarem ao meu lado em mais essa etapa da minha vida.

*"A persistência é o caminho do êxito."*  
- *Charles Darwin*

## RESUMO

O avanço das tecnologias digitais tem possibilitado o desenvolvimento de soluções inovadoras para a preservação ambiental, especialmente no contexto de biomas ameaçados, como o Cerrado. Este trabalho apresenta o desenvolvimento da plataforma *web* Protege Meu Cerrado, um sistema voltado para a denúncia de crimes ambientais, com o objetivo de conectar a comunidade às autoridades responsáveis pela fiscalização. A plataforma permite o registro de ocorrências relacionadas a desmatamento, queimadas ilegais e poluição, viabilizando uma comunicação mais eficiente e promovendo o engajamento social na proteção ambiental.

O projeto foi proposto no âmbito da disciplina Desenvolvimento de Projeto Aplicado, ministrada pelo professor Josimar Viana, no 6º período do curso de Análise e Desenvolvimento de Sistemas, e idealizado por Pedro Henrique. A implementação do sistema envolveu a utilização de tecnologias modernas para o desenvolvimento do *frontend*, sendo *Next.js* e *React* as principais ferramentas adotadas. O desenvolvimento da interface, do qual fui integrante responsável, seguiu princípios de usabilidade e responsividade, garantindo uma navegação intuitiva e eficiente para os usuários. Além disso, foram aplicadas boas práticas de programação, incluindo componentização, otimização de carregamento e escalabilidade do sistema.

Diante da crescente degradação do Cerrado e da necessidade de fortalecer a fiscalização ambiental, a plataforma Protege Meu Cerrado se apresenta como uma alternativa tecnológica para facilitar a participação cidadã no monitoramento e denúncia de crimes ambientais. A solução desenvolvida demonstra o potencial das aplicações *web* na promoção de iniciativas voltadas à sustentabilidade e preservação do meio ambiente.

**Palavras-chave:** Cerrado, denúncia ambiental, tecnologias *web*, *frontend*, *Next.js*, *React*, usabilidade.

## ABSTRACT

The advancement of digital technologies has enabled the development of innovative solutions for environmental preservation, especially in the context of threatened biomes such as the Cerrado. This study presents the development of the *Protege Meu Cerrado* platform, a system designed for reporting environmental crimes, aiming to connect the community with the authorities responsible for environmental oversight. The platform allows users to report occurrences related to deforestation, illegal fires, and pollution, facilitating more efficient communication and promoting social engagement in environmental protection.

The project was proposed within the *Applied Project Development* course, taught by Professor Josimar Viana, during the 6th semester of the *Analysis and Systems Development* program, and was conceived by Pedro Henrique. The system's implementation involved the use of modern technologies for *frontend* development, with *Next.js* and *React* as the main tools adopted. The interface development, for which I was the responsible member, followed usability and responsiveness principles, ensuring intuitive and efficient navigation for users. Additionally, best programming practices were applied, including componentization, load optimization, and system scalability.

Given the increasing degradation of the Cerrado and the need to strengthen environmental oversight, the *Protege Meu Cerrado* platform presents itself as a technological alternative to facilitate citizen participation in monitoring and reporting environmental crimes. The developed solution demonstrates the potential of web applications in promoting initiatives focused on sustainability and environmental conservation.

**Keywords:** Cerrado, environmental reporting, web technologies, *frontend*, *Next.js*, *React*, usability.

## LISTA DE FIGURAS

Figura 1 – Diagrama de caso de uso.	31
Figura 2 – Etapa 1 da tela de denúncia.	36
Figura 3 – Etapa 2 da tela de denúncia.	37
Figura 4 – Etapa 3 da tela de denúncia.	38
Figura 5 – Tela de <i>login</i> .	40
Figura 6 – Tela de cadastro de usuário.	41
Figura 7 – Tela de recuperação de conta.	42
Figura 8 – Tela de listagem de denúncias.	43
Figura 9 – Tela de edição de perfil.	44
Figura 10 – Tela de acompanhamento de denúncias.	46
Figura 11 – Tela de aceitar denúncias.	47
Figura 12 – Tela de acompanhar denúncias em tratamento.	48
Figura 13 – Tela de adicionar uma tratativa à denúncia.	49
Figura 14 – Tela de cadastrar autoridade.	50
Figura 15 – Tela de vincular usuário.	51
Figura 16 – Tela de informações de autoridades.	52
Figura 17 – Tela de deletar denúncias.	53
Figura 18 – Tela inicial - parte 1.	54
Figura 19 – Tela inicial - parte 2.	55
Figura 20 – Tela inicial - parte 3.	55
Figura 21 – Tela inicial - parte 4.	56
Figura 22 – Tela de perguntas frequentes.	56
Figura 23 – Seção de termos de uso.	57
Figura 24 – Seção de política de privacidade.	58
Figura 25 – Inauguração do Protege Meu Cerrado..	64

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 TEMA.....	11
1.2 PROBLEMA.....	12
1.3 OBJETIVO GERAL.....	13
1.4 OBJETIVOS ESPECÍFICOS.....	13
1.5 JUSTIFICATIVA.....	15
1.6 DELIMITAÇÃO.....	17
<b>2 REFERENCIAL TEÓRICO.....</b>	<b>19</b>
2.1 NEXT.JS.....	19
2.2 REACT.JS.....	20
2.3 TYPESCRIPT.....	21
2.4 BULMA.....	22
2.5 REACT GOOGLE MAPS API.....	23
2.6 REACT INPUT MASK.....	24
2.7 SWIPER.....	25
2.8 DATE-FNS.....	25
2.9 GIT E GITHUB.....	26
2.10 INSOMNIA E POSTMAN.....	27
<b>3 DESENVOLVIMENTO.....</b>	<b>28</b>
3.1 DEFINIÇÃO DOS REQUISITOS E ARQUITETURA DO FRONTEND.....	28
3.2 CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	31
3.3 DESENVOLVIMENTO POR PACOTES.....	33
3.3.1 Pacote 1: tela de denúncias.....	34
3.3.2 Pacote 2: Cadastro de usuário e listagem de ocorrências.....	39
3.3.3 Pacote 3: Dashboard para usuário comum.....	43
3.3.4 Pacote 4: Dashboard para usuário vinculados à autoridades.....	46
3.3.5 Pacote 5: Dashboard para usuário administrador.....	50
3.3.6 Pacote 6: Criação da tela inicial e perguntas frequentes.....	53
3.3.7 Pacote 7: Melhoria na estilização e identidade visual.....	58
3.4 DESAFIOS ENCONTRADOS.....	60
<b>4 CONCLUSÃO.....</b>	<b>63</b>
<b>REFERÊNCIAS.....</b>	<b>65</b>



## 1 INTRODUÇÃO

O Cerrado brasileiro, considerado um dos biomas mais ricos em biodiversidade, desempenha um papel crucial na regulação climática e na manutenção dos recursos hídricos do país. No entanto, nas últimas décadas, esse ecossistema tem enfrentado uma crescente degradação devido à ação humana, particularmente a expansão das atividades agrícolas e pecuárias. O desmatamento, a ocorrência de queimadas ilegais e a ocupação desordenada do solo são fatores que aceleram a destruição do Cerrado, colocando em risco tanto sua biodiversidade quanto os serviços ecossistêmicos essenciais que ele fornece.

De acordo com dados do MapBiomas, entre 1985 e 2020, o Cerrado perdeu aproximadamente 26,5 milhões de hectares de vegetação nativa, o que representa uma redução significativa da cobertura original do bioma. Em 2023, as taxas de desmatamento no Cerrado superaram, pela primeira vez, as da Amazônia, atingindo 1.110.326 hectares de área desmatada, um aumento de 67,7% em relação ao ano anterior. Esse processo de degradação compromete a fauna e a flora local, além de afetar diretamente a segurança hídrica de diversas regiões do Brasil, uma vez que o Cerrado é responsável pela alimentação das principais bacias hidrográficas do país, incluindo as bacias do São Francisco, do Prata e a bacia Amazônica.

Diante desse cenário alarmante, torna-se urgente o fortalecimento das políticas públicas de fiscalização ambiental, além de se buscar novas formas de engajamento social para o combate aos crimes ambientais. A tecnologia, especialmente as plataformas digitais, surge como uma ferramenta eficaz para a melhoria da comunicação entre a população e as autoridades responsáveis pela fiscalização. Essas soluções tecnológicas podem facilitar a denúncia de infrações ambientais, amplificando a participação cidadã e, consequentemente, a atuação dos órgãos competentes. O uso de ferramentas digitais para o bem-estar social e ambiental tem sido uma tendência crescente, como demonstrado por Sachs et al. (2020), que discute como inovações tecnológicas têm sido aplicadas para o monitoramento e combate de crimes ambientais, promovendo iniciativas de sustentabilidade global.

Neste contexto, o presente trabalho propõe o desenvolvimento da plataforma *web* Protege Meu Cerrado, uma ferramenta digital destinada à denúncia de crimes

ambientais no Cerrado, com o objetivo de conectar a comunidade às autoridades responsáveis pela fiscalização e pela proteção ambiental. A plataforma foi proposta no âmbito da disciplina Desenvolvimento de Projeto Aplicado, ministrada pelo professor Josimar Viana, no 6º período do curso de Análise e Desenvolvimento de Sistemas, sendo idealizada por Pedro Henrique. Integrante da equipe de desenvolvimento frontend da plataforma web, fui responsável pelo desenvolvimento do *frontend* do sistema, utilizando as tecnologias *Next.js* e *React*, com foco na criação de uma interface intuitiva e responsiva, que garantisse uma navegação eficiente para os usuários.

O sistema desenvolvido busca proporcionar uma solução ágil e eficaz para o registro de denúncias de crimes ambientais, como desmatamento ilegal, queimadas e poluição. A plataforma também visa promover o engajamento da população na proteção do Cerrado, tornando mais eficiente a comunicação entre cidadãos e autoridades, além de contribuir para o fortalecimento da fiscalização ambiental.

## 1.1 TEMA

O presente trabalho aborda o desenvolvimento de soluções tecnológicas voltadas para a denúncia e fiscalização de crimes ambientais, com foco na criação de uma plataforma digital eficiente. O tema está inserido no contexto da preservação ambiental, mais especificamente na proteção do bioma Cerrado, um dos mais ameaçados no Brasil devido ao desmatamento, queimadas ilegais e ocupação desordenada do solo.

A proposta está alinhada à crescente demanda por ferramentas digitais que facilitem a participação cidadã na proteção ambiental, permitindo que qualquer pessoa possa registrar denúncias de maneira rápida, intuitiva e segura. Além disso, o estudo insere-se no campo da tecnologia da informação aplicada ao meio ambiente, explorando como a utilização de frameworks modernos, metodologias ágeis e boas práticas de desenvolvimento frontend podem contribuir para a criação de plataformas eficientes.

Dessa forma, este trabalho se situa na interseção entre tecnologia e sustentabilidade, analisando como o desenvolvimento web pode ser utilizado como uma ferramenta para promover o engajamento social, otimizar a comunicação

entre cidadãos e autoridades e fortalecer as políticas públicas de fiscalização ambiental.

## 1.2 PROBLEMA

Este trabalho busca investigar como o desenvolvimento do frontend de uma plataforma digital pode facilitar a denúncia de crimes ambientais no Cerrado, criando uma experiência de uso que seja ágil e intuitiva a todos os cidadãos. O Cerrado, um dos biomas mais ricos em biodiversidade, está sendo devastado por atividades como desmatamento ilegal, queimadas e a ocupação desordenada do solo, o que exige uma resposta urgente tanto das autoridades quanto da sociedade. Para que a fiscalização ambiental seja mais eficiente, é fundamental aumentar o engajamento da população na denúncia desses crimes, de maneira que a participação cidadã seja estimulada e não desestimulada.

A criação de uma plataforma acessível é essencial para garantir que qualquer pessoa, independentemente do seu nível de conhecimento tecnológico, consiga registrar denúncias de maneira simples e rápida. Nesse sentido, é preciso desenvolver uma interface *frontend* que permita uma navegação fluida, seja responsiva para diferentes dispositivos e ofereça uma experiência que incentive os usuários a interagir com a plataforma de forma contínua. A plataforma não deve apenas ser funcional, mas também agradável de usar, de modo que a denúncia de crimes ambientais se torne algo fácil e sem frustrações para os cidadãos.

Além disso, ao longo do tempo, é necessário que a plataforma ganhe visibilidade, ampliando seu alcance e engajamento, o que contribui diretamente para a intensificação da fiscalização ambiental. Ao adotar tecnologias modernas como *Next.js* e *React*, a plataforma tem o potencial de se tornar uma solução escalável, otimizada e com alta performance, que possa atender uma grande base de usuários de forma eficiente. A questão central deste trabalho é: como o uso de tecnologias como *Next.js* e *React* pode contribuir para o desenvolvimento de uma interface *frontend* que seja simples, eficiente e capaz de promover a participação ativa da sociedade na preservação do Cerrado, garantindo uma experiência que incentive as denúncias e se torne mais visível no futuro.

### 1.3 OBJETIVO GERAL

O objetivo geral deste trabalho é o desenvolvimento do *frontend* da plataforma digital Protege Meu Cerrado, utilizando tecnologias *web* modernas, como *Next.js* e *React*, com foco na criação de uma interface responsiva e intuitiva. A proposta visa proporcionar aos usuários uma experiência de navegação eficiente, que facilite o registro e a comunicação de denúncias de crimes ambientais no Cerrado, promovendo o engajamento da sociedade na fiscalização e preservação desse bioma.

Para garantir que a plataforma seja robusta, eficiente e alinhada às exigências do mercado tecnológico atual, foi necessário recorrer a tecnologias amplamente utilizadas na indústria de desenvolvimento *web*, o que exigiu um aprofundamento no aprendizado e na aplicação dessas ferramentas. Durante a implementação do *frontend*, buscou-se compreender e empregar as melhores práticas de desenvolvimento, incluindo a utilização de *frameworks* modernos, otimização de performance, componentização e estratégias de renderização. Esse processo não apenas permitiu o aprimoramento técnico, mas também reforçou a importância da adaptação contínua às inovações tecnológicas, um fator essencial para profissionais da área de desenvolvimento de sistemas.

### 1.4 OBJETIVOS ESPECÍFICOS

- Definir os requisitos para uma interface *frontend* eficiente
  - Realizar um levantamento das principais necessidades dos usuários da plataforma, considerando fatores como usabilidade e responsividade, a fim de garantir que a ferramenta atenda ao seu propósito sem criar barreiras que desestimulem o ato de denunciar crimes ambientais.
- Projetar e desenvolver uma interface visual intuitiva e responsiva
  - Elaborar um *design* de interface que proporcione uma experiência de navegação fluida e coerente, garantindo que o usuário possa registrar denúncias de forma rápida e eficaz, tanto em dispositivos móveis quanto em *desktops*, utilizando princípios modernos de *UI* (*User Interface*) e *UX* (*User Experience*).
- Implementar boas práticas de desenvolvimento

- Aplicar padrões de desenvolvimento *frontend*, como componentização, reutilização de código, otimização de carregamento e escalabilidade do sistema, garantindo que a plataforma tenha um desempenho otimizado e esteja alinhada às exigências do mercado tecnológico atual.
- Utilizar tecnologias modernas para maximizar a eficiência do *frontend*
  - Explorar as vantagens oferecidas por tecnologias como *Next.js* e *React*, empregando suas funcionalidades para proporcionar uma melhor experiência ao usuário, incluindo *SSG (Static Site Generation)* para páginas dinâmicas e *React Hooks* para maior eficiência na manipulação de estados.
- Testar e validar a usabilidade da plataforma
  - Realizar testes de usabilidade com usuários reais e aplicar métodos de avaliação para identificar possíveis pontos de melhoria na interface, garantindo que a experiência do usuário seja intuitiva, satisfatória e alinhada ao objetivo de facilitar a denúncia de crimes ambientais.
- Assegurar a escalabilidade e o crescimento da plataforma
  - Desenvolver o *frontend* com foco na escalabilidade, permitindo que a plataforma suporte um número crescente de usuários sem comprometer sua performance, possibilitando futuras expansões e integrações com novas funcionalidades.
- Aprimorar conhecimentos em ferramentas de versionamento de código
  - Utilizar e consolidar o aprendizado no uso do *Git* e plataformas de versionamento, como *GitHub*, garantindo o controle eficiente das alterações no código, o trabalho colaborativo e a rastreabilidade do desenvolvimento.
- Compreender e aplicar ferramentas de teste e consumo de *APIs*
  - Utilizar ferramentas como *Insomnia* e *Postman* para testar e validar as requisições feitas ao *backend*, garantindo que a comunicação entre *frontend* e *backend* ocorra de maneira eficiente e segura.
- Desenvolver habilidades na integração e comunicação com o *backend*
  - Aprender a consumir e integrar *APIs REST*, compreendendo conceitos como requisições *HTTP*, autenticação, manipulação de dados e tratamento de erros, assegurando a funcionalidade adequada da plataforma.

- Atender às demandas do *Product Owner* e atuar dentro de um fluxo de desenvolvimento ágil
  - Compreender o papel do *Product Owner* no desenvolvimento do sistema, acompanhar reuniões de definição de requisitos e priorização de tarefas, garantindo que as entregas estejam alinhadas às expectativas do projeto.
- Aprender a trabalhar em equipe em um ambiente de desenvolvimento colaborativo
  - Desenvolver habilidades de comunicação e colaboração no ambiente de desenvolvimento, utilizando metodologias ágeis, como o *Scrum*, para a organização de tarefas e alinhamento com os demais membros da equipe.
- Garantir uma abordagem orientada à manutenção e evolução contínua do código
  - Desenvolver um código limpo, modular e bem documentado, facilitando futuras manutenções e permitindo que outros desenvolvedores possam compreender e expandir a plataforma com facilidade.
- Explorar a importância da experiência prática para o desenvolvimento profissional
  - Consolidar o aprendizado adquirido ao longo do projeto, refletindo sobre a relevância da experiência prática para a atuação no mercado de tecnologia, especialmente na área de desenvolvimento *frontend*.

## 1.5 JUSTIFICATIVA

A justificativa deste trabalho se baseia na crescente demanda por soluções tecnológicas que possam contribuir para a preservação ambiental e o monitoramento de crimes ambientais, especialmente em biomas críticos como o Cerrado. O Cerrado, um dos maiores biomas brasileiros, é responsável pela regulação climática e pela manutenção dos recursos hídricos do país. No entanto, sua constante degradação, impulsionada principalmente pelo desmatamento, queimadas ilegais e ocupação desordenada do solo, compromete gravemente seus serviços ecossistêmicos e coloca em risco a biodiversidade local. Nesse cenário, o desenvolvimento de plataformas digitais que facilitem a denúncia e a

fiscalização de crimes ambientais torna-se uma necessidade urgente, visando fortalecer as políticas públicas e engajar a sociedade na proteção deste bioma.

O projeto Protege Meu Cerrado surge como uma resposta a essa necessidade, oferecendo uma plataforma digital voltada para a denúncia de crimes ambientais, com o objetivo de conectar a comunidade às autoridades responsáveis pela fiscalização e pela proteção ambiental. Ao integrar a equipe de desenvolvimento *frontend*, fui desafiada a aplicar conhecimentos adquiridos ao longo da graduação e a aprender novas tecnologias de ponta, como *Next.js* e *React*, ferramentas amplamente utilizadas no desenvolvimento de sistemas modernos e escaláveis. Essa experiência foi extremamente enriquecedora, pois me forçou a aprofundar o conhecimento sobre essas tecnologias e a explorar suas capacidades, como a criação de interfaces intuitivas e responsivas que proporcionam uma melhor experiência ao usuário, além de garantir a otimização do desempenho do sistema.

Durante o desenvolvimento da plataforma, tive a oportunidade de colocar em prática boas práticas de programação, como a componentização do código, o uso de *hooks* e o gerenciamento eficiente do estado da aplicação. Além disso, trabalhei com conceitos de *design* responsivo, o que garantiu que a plataforma fosse acessível a diferentes dispositivos e proporcionasse uma navegação fluida. Todo esse processo foi enriquecedor tanto no aspecto técnico quanto no aspecto social, pois pude contribuir para um projeto que, além de agregar valor ao meu desenvolvimento profissional, também tem um impacto direto na sociedade ao promover a participação cidadã na proteção do Cerrado.

Além disso, também se justifica pelo valor significativo que o aprendizado de tecnologias atuais tem no mercado de trabalho para analistas de sistemas. As ferramentas e abordagens modernas, como *Next.js* e *React*, são altamente demandadas no setor de desenvolvimento *web*, que busca profissionais capazes de criar aplicações eficientes, escaláveis e com boas práticas de usabilidade e performance. Ao aprender e aplicar essas tecnologias em um projeto real, sinto que estou me preparando de maneira sólida para os desafios do mercado, especialmente na área de desenvolvimento *frontend*, que tem apresentado um crescimento considerável nos últimos anos. Além disso, o conhecimento adquirido vai além da simples aplicação de ferramentas tecnológicas, pois envolve também a compreensão de como a tecnologia pode ser usada para o bem, neste caso,

contribuindo para a preservação ambiental e o fortalecimento da fiscalização de crimes ambientais.

A combinação do aprendizado técnico com a aplicação de uma solução que visa a conscientização e o engajamento da população na proteção do Cerrado é um aspecto que torna este trabalho particularmente relevante e valioso. Por meio da experiência de integrar a equipe de *frontend* da plataforma Protege Meu Cerrado, fui capaz de expandir meus conhecimentos, desenvolver novas habilidades e, ao mesmo tempo, contribuir para um projeto com impacto social positivo. Esse aprendizado técnico, aliado à consciência de seu impacto ambiental, certamente fortalecerá minha atuação como analista de sistemas e me permitirá enfrentar com mais confiança os desafios que o mercado de trabalho exige.

Em resumo, a escolha deste projeto se justifica pela sua relevância no combate aos crimes ambientais no Cerrado, bem como pela oportunidade de aplicar e expandir meus conhecimentos em tecnologias modernas de desenvolvimento *frontend*. Ao colaborar com a criação de uma plataforma digital de denúncia ambiental, contribuí para um projeto que, além de ampliar minha experiência técnica, também promove a conscientização ambiental e a participação ativa da sociedade na proteção de um dos biomas mais importantes do Brasil.

## 1.6 DELIMITAÇÃO

Este estudo está inserido na área da tecnologia da informação aplicada ao desenvolvimento *web*, com ênfase na construção de interfaces digitais intuitivas. A pesquisa concentra-se no desenvolvimento do *frontend web* da plataforma Protege Meu Cerrado, utilizando tecnologias modernas, como *Next.js* e *React*, para proporcionar uma experiência fluida e responsiva aos usuários que desejam denunciar crimes ambientais.

O espaço geográfico de abrangência da pesquisa refere-se ao bioma Cerrado brasileiro, um dos mais ameaçados do país devido ao avanço do desmatamento e das queimadas ilegais. Embora a plataforma tenha sido projetada para atender a todo o bioma do Cerrado, sua fase inicial de testes e implementação está concentrada na região de Paracatu-MG. No futuro, pretende-se expandir sua abrangência para outras áreas do Cerrado, ampliando o impacto da ferramenta na



proteção ambiental.

Dessa forma, a pesquisa delimita-se ao estudo e à aplicação de ferramentas de desenvolvimento *frontend* para a criação de uma plataforma de denúncia ambiental, considerando princípios de *design*, usabilidade e boas práticas para aplicações *web* modernas.

## 2 REFERENCIAL TEÓRICO

O desenvolvimento de aplicações *web* modernas exige a utilização de tecnologias e ferramentas que garantam desempenho e uma experiência intuitiva para os usuários. Neste contexto, o projeto Protege Meu Cerrado foi implementado com base em uma série de *frameworks*, bibliotecas e metodologias amplamente utilizadas no mercado de desenvolvimento *frontend*. Esta seção apresenta o embasamento teórico das principais ferramentas adotadas.

### 2.1 NEXT.JS

O *Next.js* é um dos *frameworks* mais populares para o desenvolvimento de aplicações *web* modernas, sendo amplamente adotado por sua capacidade de oferecer uma experiência de usuário otimizada, combinando alto desempenho e flexibilidade. Criado pela *Vercel*, o *Next.js* é baseado no *React* e se destaca por fornecer suporte nativo a diferentes estratégias de renderização, como SSR (*Server-Side Rendering*), SSG (*Static Site Generation*) e ISR (*Incremental Static Regeneration*), permitindo que as páginas da aplicação sejam processadas e entregues de forma otimizada, reduzindo o tempo de carregamento e melhorando a indexação por mecanismos de busca (*Hybrid Heroes*, 2023). Além disso, sua arquitetura facilita a construção de interfaces escaláveis e bem estruturadas, tornando-se uma escolha ideal para projetos que exigem alta performance e modularidade.

Outro diferencial importante do *Next.js* é seu sistema de roteamento automático baseado em arquivos, o que simplifica a criação e organização das páginas dentro do projeto. Diferente do *React* tradicional, onde o roteamento precisa ser configurado manualmente através de bibliotecas como *React Router*, o *Next.js* permite que os desenvolvedores simplesmente adicionem arquivos dentro da pasta *pages/* para definir novas rotas. Além disso, ele conta com *API Routes*, uma funcionalidade que possibilita a criação de *endpoints* diretamente no *backend* do próprio projeto *Next.js*, eliminando a necessidade de um servidor separado para a manipulação de dados em pequenas aplicações (*Karimov*, 2023). Esse recurso é especialmente útil para prototipagem rápida e desenvolvimento de *APIs* simples.

A otimização de desempenho é outro aspecto fundamental do *Next.js*, que

conta com funcionalidades integradas como otimização automática de imagens, *Code Splitting* e *Lazy Loading*. A otimização de imagens, por exemplo, permite o carregamento progressivo e o uso de formatos modernos, como *WebP*, garantindo uma experiência visual mais fluida e rápida. O *Code Splitting*, por sua vez, carrega apenas os componentes necessários para cada página, reduzindo o tamanho inicial dos arquivos e tornando a navegação mais eficiente. Além disso, o suporte ao *Prefetching* de páginas permite que o *Next.js* antecipe quais páginas o usuário pode acessar a seguir, carregando-as previamente para uma experiência de navegação praticamente instantânea (Eko Pradana, 2023).

No contexto do projeto Protege Meu Cerrado, a utilização do *Next.js* foi crucial para garantir um *frontend* ágil e eficiente, atendendo aos requisitos de desempenho, escalabilidade e boa experiência do usuário. O *framework* foi utilizado para implementar a interface de forma responsiva e otimizada, permitindo que os usuários registrem denúncias ambientais de maneira rápida e intuitiva.

## 2.2 REACT.JS

O *React* é uma das bibliotecas *JavaScript* mais amplamente utilizadas para o desenvolvimento de interfaces de usuário interativas e dinâmicas. Criado pelo *Facebook* (atual *Meta*) em 2013, o *React* revolucionou o desenvolvimento *web* ao introduzir o conceito de componentização, permitindo que a interface seja dividida em partes reutilizáveis e independentes. Essa abordagem modular facilita a manutenção do código, melhora a escalabilidade do projeto e promove um desenvolvimento mais eficiente (*React Documentation*, 2024). Além disso, o *React* adota um modelo de programação declarativa, no qual os desenvolvedores descrevem como a interface deve se comportar com base no estado da aplicação, reduzindo a complexidade do gerenciamento de interações dinâmicas na tela (*Revelo Community*, 2023).

Um dos recursos mais inovadores do *React* é o seu *Virtual DOM* (*Document Object Model Virtual*), que melhora significativamente o desempenho da renderização da interface. Em aplicações *web* tradicionais, qualquer atualização na interface pode exigir uma re-renderização completa da página, tornando o processo lento e ineficiente. O *Virtual DOM*, no entanto, cria uma cópia leve do *DOM* real e apenas atualiza os elementos que sofreram alterações, minimizando o número de

operações no navegador e melhorando a performance da aplicação (GeeksforGeeks, 2024). Essa tecnologia é essencial para proporcionar uma experiência de usuário mais fluida e responsiva, especialmente em plataformas que exigem atualizações frequentes de dados, como o Protege Meu Cerrado.

Outro fator que torna o *React* uma escolha estratégica para o desenvolvimento *frontend* é seu ecossistema rico e ativo, com diversas bibliotecas e ferramentas complementares. Recursos como o *React Hooks* permitem o gerenciamento do estado da aplicação de maneira mais simples e eficiente, sem a necessidade de classes, tornando o código mais limpo e legível (Brasil Code, 2024). Além disso, o *React* possui compatibilidade com diversos gerenciadores de estado, como *Redux*, *Context API* e *Zustand*, possibilitando a manipulação de dados de forma centralizada e organizada (React Documentation, 2024). No contexto do Protege Meu Cerrado, essa característica foi fundamental para gerenciar informações como o estado das denúncias, dados do usuário e interações na interface, garantindo um fluxo contínuo e bem estruturado.

No desenvolvimento do Protege Meu Cerrado, o *React* foi utilizado para criar uma interface dinâmica e responsiva, garantindo que os usuários pudessem registrar denúncias ambientais de forma intuitiva e sem complicações. A modularidade proporcionada pelo *React* possibilitou a criação de componentes reutilizáveis, como botões, formulários e *cards* informativos, tornando a construção da plataforma mais eficiente e escalável. Além disso, a integração do *React* com outras tecnologias, como o *Next.js* e bibliotecas auxiliares (por exemplo, *React Input Mask* para formulários), permitiu a implementação de uma experiência de navegação fluida (Brasil Code, 2024). Dessa forma, a escolha pelo *React* se mostrou essencial para garantir a qualidade e o desempenho do *frontend* da plataforma, contribuindo diretamente para a usabilidade do sistema.

## 2.3 TYPESCRIPT

O *TypeScript* é uma linguagem de programação desenvolvida pela *Microsoft* que se baseia no *JavaScript*, adicionando suporte à tipagem estática. Esse recurso permite que os desenvolvedores definam tipos para variáveis, funções e objetos, o que reduz erros durante a fase de desenvolvimento e aumenta a previsibilidade do código. Diferentemente do *JavaScript* puro, onde os tipos são dinâmicos e podem

levar a falhas inesperadas em tempo de execução, o *TypeScript* possibilita a detecção desses erros já na fase de compilação, evitando problemas que poderiam comprometer a estabilidade da aplicação. Essa abordagem favorece um código mais seguro e confiável, especialmente em projetos de grande porte (DIO, 2024). Além disso, a linguagem oferece suporte a recursos modernos, como interfaces, *generics* e *enums*, que tornam o desenvolvimento mais organizado e a manutenção do código mais eficiente (AppMaster, 2024).

Outro grande benefício do *TypeScript* é a melhoria na legibilidade e produtividade do código, algo especialmente relevante em projetos colaborativos. Como a linguagem permite a definição clara dos tipos de dados e suas estruturas, o código se torna mais intuitivo para os desenvolvedores, facilitando sua manutenção e evolução ao longo do tempo. Além disso, ferramentas como o *IntelliSense*, presentes em editores como o *Visual Studio Code*, auxiliam os programadores ao sugerirem automaticamente trechos de código, tornando o processo de desenvolvimento mais ágil e eficiente (DIO, 2024). No contexto do Protege Meu Cerrado, a utilização do *TypeScript* foi essencial para estruturar corretamente os dados das denúncias, usuários e interações na plataforma, garantindo maior consistência e segurança no desenvolvimento do *frontend*.

Além disso, o *TypeScript* desempenhou um papel fundamental na comunicação entre o *frontend* e o *backend*, reduzindo a possibilidade de erros ao enviar ou receber dados. A tipagem estática possibilitou a definição de contratos claros entre as diferentes partes da aplicação, garantindo que os dados fossem manipulados de maneira previsível e coerente. Essa característica é especialmente útil quando integrada a tecnologias como *React* e *Next.js*, permitindo a construção de componentes robustos e reutilizáveis (LogRocket, 2024). Dessa forma, a escolha pelo *TypeScript* não apenas aprimorou a qualidade do código, mas também fortaleceu a escalabilidade da plataforma, tornando-a mais preparada para futuras expansões e melhorias.

## 2.4 BULMA

O *Bulma* é um framework CSS baseado em *Flexbox*, projetado para facilitar o desenvolvimento de interfaces responsivas e modernas. Diferente de *frameworks* mais tradicionais, como o *Bootstrap*, o *Bulma* se destaca por sua abordagem

puramente CSS, sem a necessidade de *JavaScript* para funcionamento básico. Isso o torna mais leve e eficiente na integração com bibliotecas e *frameworks frontend*, como *React* e *Next.js*. O *Bulma* fornece uma série de componentes prontos, como *grids*, botões, formulários e *cards*, permitindo que os desenvolvedores criem *layouts* elegantes e organizados de maneira ágil, com sintaxe intuitiva e modularidade que favorecem um desenvolvimento mais dinâmico e estruturado (Bulma, 2024).

Um dos principais diferenciais do *Bulma* é sua forte ênfase em *design* responsivo, garantindo que as interfaces se adaptem automaticamente a diferentes tamanhos de tela, sem a necessidade de ajustes manuais excessivos. Com classes bem definidas para espaçamentos, alinhamentos e cores, o *Bulma* simplifica a construção de interfaces acessíveis e visualmente coerentes. No desenvolvimento do Protege Meu Cerrado, o *framework* foi fundamental para criar uma interface clara, objetiva e intuitiva, assegurando que os usuários pudessem navegar e registrar denúncias ambientais de forma simples e eficiente. Sua compatibilidade com customizações via variáveis CSS também permitiu ajustes personalizados na identidade visual da plataforma (Andrade, 2023).

Outro aspecto relevante do *Bulma* é sua compatibilidade com práticas modernas de desenvolvimento *frontend*, como o uso de pré-processadores CSS, o que possibilita maior organização e reaproveitamento de estilos (Caputo, 2023). No contexto do Protege Meu Cerrado, o *Bulma* foi utilizado para padronizar a interface e manter um *design* coeso entre os diferentes componentes da aplicação, reduzindo o tempo de desenvolvimento e facilitando a manutenção do código. Isso proporcionou um código mais limpo e modular, sem a necessidade de dependências pesadas, garantindo uma experiência de usuário fluida e agradável.

## 2.5 REACT GOOGLE MAPS API

A biblioteca *@react-google-maps/api* é uma das principais soluções para a integração do *Google Maps* em aplicações *React*, permitindo que desenvolvedores incorporem mapas interativos e personalizáveis de maneira eficiente. Criada como uma alternativa mais moderna e otimizada ao antigo *react-google-maps*, essa biblioteca melhora o desempenho ao utilizar uma abordagem baseada em *React Hooks* e *Lazy Loading*, garantindo que os recursos do *Google Maps* sejam carregados apenas quando necessário. Isso resulta em tempos de carregamento

reduzidos e melhor desempenho, algo essencial para aplicações que dependem de mapas dinâmicos, como o Protege Meu Cerrado (*React Google Maps*, 2024).

A capacidade de exibir mapas, adicionar marcadores e interagir com elementos geoespaciais torna o *@react-google-maps/api* uma ferramenta essencial para aplicações que envolvem localização. No contexto do Protege Meu Cerrado, essa biblioteca foi utilizada para permitir que os usuários visualizassem e registrassem denúncias diretamente em um mapa, facilitando a identificação de áreas afetadas por crimes ambientais. Além disso, recursos como geocodificação, busca por localização e ajuste de *zoom* dinâmico foram empregados para tornar a experiência do usuário mais fluida e intuitiva, garantindo que as informações fossem exibidas de forma clara e acessível (*Developer*, 2023).

Outro ponto de destaque do *@react-google-maps/api* é sua flexibilidade e compatibilidade com outras bibliotecas de *React*, permitindo integrações avançadas, como o uso de eventos para captura de coordenadas e customização da aparência do mapa (*Google Maps*, 2023).

No desenvolvimento do Protege Meu Cerrado, essa funcionalidade foi essencial para que os usuários pudessem posicionar com precisão os locais das infrações ambientais, contribuindo para a eficácia do sistema de denúncias. A escolha dessa biblioteca se mostrou estratégica, pois além de otimizar o desempenho da aplicação, garantiu uma interface interativa e moderna para a geolocalização das ocorrências registradas na plataforma.

## 2.6 REACT INPUT MASK

O *React-Input-Mask* é uma biblioteca voltada para a aplicação de máscaras em campos de entrada de formulários em aplicações *React*. Essa funcionalidade é essencial para garantir que os dados sejam inseridos no formato correto, reduzindo erros e melhorando a usabilidade. A biblioteca permite definir padrões de entrada para informações como números de telefone, CPF, datas e coordenadas geográficas, tornando a interação do usuário mais intuitiva e evitando a necessidade de validações excessivas após o envio do formulário. Sua implementação é simples e eficiente, sendo compatível com componentes controlados do *React*, o que facilita sua integração em aplicações modernas (*React Input Mask*, 2023).

No desenvolvimento do Protege Meu Cerrado, o *React-Input-Mask* foi

utilizado para padronizar a entrada de dados em campos estratégicos, como telefone para contato e CEP. Isso garantiu que os usuários pudessem inserir informações de maneira estruturada e sem ambiguidades, reduzindo erros no envio de dados. Além disso, a utilização de máscaras proporcionou uma experiência mais profissional e consistente, ajudando a tornar a interface da plataforma mais acessível e funcional.

## 2.7 SWIPER

O *Swiper* é uma das bibliotecas *JavaScript* mais populares para a criação de carrosséis e *slides* interativos. Sua flexibilidade e desempenho otimizado o tornam uma excelente escolha para aplicações modernas que necessitam exibir conteúdos dinâmicos de forma visualmente atrativa. O *Swiper* permite a implementação de transições suaves, navegação por gestos e suporte a diferentes *layouts* responsivos, sendo altamente personalizável através de suas diversas opções de configuração. Além disso, sua compatibilidade com *React* e *Next.js* facilita sua integração em aplicações que utilizam esses *frameworks*, garantindo um carregamento eficiente dos componentes visuais (Swiper, 2023).

Sua implementação garantiu uma experiência mais agradável para os usuários, permitindo uma navegação fluida e intuitiva entre os conteúdos. Além disso, o *Swiper* possibilitou ajustes responsivos automáticos, assegurando que o carrossel se adaptasse corretamente a diferentes tamanhos de tela, tanto em dispositivos móveis quanto em *desktops* (Swiper, 2023).

Outro aspecto importante do *Swiper* é sua otimização de desempenho, pois a biblioteca oferece suporte a *lazy loading*, garantindo que as imagens e outros conteúdos sejam carregados apenas quando necessários (Swiper, 2023). Isso foi fundamental para manter a eficiência do sistema, evitando carregamentos desnecessários que poderiam impactar a velocidade e a usabilidade da plataforma. Com a implementação do *Swiper* no Protege Meu Cerrado, a interface se tornou mais dinâmica e interativa, agregando valor à experiência do usuário e tornando a navegação mais envolvente.

## 2.8 DATE-FNS

O *date-fns* é uma biblioteca *JavaScript* amplamente utilizada para



manipulação e formatação de datas. Diferente da *API* nativa de datas do *JavaScript*, que pode ser complexa e pouco intuitiva, o *date-fns* oferece funções modulares e simplificadas para lidar com operações comuns, como formatação, cálculos de tempo e manipulação de fusos horários. Além disso, a biblioteca é projetada para ser eficiente, ocupando menos espaço e garantindo um melhor desempenho em comparação a outras alternativas, como o *Moment.js* (*Date-fns*, 2024).

No desenvolvimento do Protege Meu Cerrado, o *date-fns* foi empregado para formatar e exibir corretamente as datas de denúncias registradas na plataforma, garantindo que os usuários pudessem visualizar as informações de forma clara e organizada.

## 2.9 GIT E GITHUB

O *Git* é um dos sistemas de controle de versão mais utilizados no desenvolvimento de *software*, permitindo o acompanhamento de alterações no código e possibilitando que múltiplos desenvolvedores trabalhem simultaneamente no mesmo projeto. Através do *Git*, é possível criar *branches* (ramificações) para testar novas funcionalidades sem comprometer a versão principal do *software*, além de permitir o controle de histórico de alterações, facilitando a reversão de mudanças quando necessário. Sua utilização é essencial para o desenvolvimento ágil e organizado de aplicações modernas (*Chacon & Straub*, 2021).

O *GitHub*, por sua vez, é uma plataforma de hospedagem de repositórios *Git* baseada na nuvem, que facilita a colaboração entre desenvolvedores. Com funcionalidades como *pull requests*, *issues* e revisões de código, o *GitHub* melhora a comunicação entre equipes e possibilita um fluxo de trabalho estruturado. Além disso, sua integração com ferramentas de *CI/CD* (Integração Contínua/Entrega Contínua) permite a automação de testes e *deploys*, agilizando o ciclo de desenvolvimento (*GitHub*, 2024).

No contexto do Protege Meu Cerrado, o uso do *Git* e do *GitHub* foi fundamental para garantir a organização do código e facilitar o trabalho em equipe. Através de *branches* dedicadas para cada funcionalidade, foi possível desenvolver e testar novas implementações sem afetar a estabilidade do sistema. Além disso, a realização de *commits* frequentes e bem documentados garantiu um histórico claro das modificações, permitindo rastrear facilmente a evolução do projeto.

Outra prática adotada foi o uso de *pull requests*, onde cada nova funcionalidade ou correção de *bug* passava por uma revisão antes de ser integrada ao código principal. Isso garantiu um desenvolvimento mais seguro e colaborativo, permitindo que os integrantes da equipe analisassem e sugerissem melhorias antes da fusão do código (Chacon & Straub, 2021; GitHub, 2024). O conhecimento adquirido sobre *Git* e *GitHub* ao longo do desenvolvimento do *frontend* do Protege Meu Cerrado foi de grande valor para a formação profissional, pois essas ferramentas são amplamente utilizadas no mercado de tecnologia.

## 2.10 INSOMNIA E POSTMAN

O *Insomnia* e o *Postman* são ferramentas amplamente utilizadas para testar e consumir *APIs*, permitindo que desenvolvedores realizem requisições *HTTP* de forma simples e organizada. Essas ferramentas possibilitam o envio de requisições *GET*, *POST*, *PUT* e *DELETE*, bem como a configuração de headers, autenticações e corpo de requisição, sendo essenciais para validar a comunicação entre o *frontend* e o *backend* de uma aplicação. Além disso, ambas permitem o armazenamento e organização de requisições, facilitando a reutilização durante o desenvolvimento (Postman, 2024; Insomnia, 2024).

No desenvolvimento do Protege Meu Cerrado, o *Insomnia* e o *Postman* foram utilizados para testar endpoints da *API* do *backend*, garantindo que os dados fossem transmitidos corretamente entre as diferentes partes do sistema. Essas ferramentas permitiram simular requisições de cadastro de usuários, registro de denúncias e consultas a dados ambientais, ajudando na identificação de possíveis falhas antes da integração definitiva no *frontend*. Além disso, a análise das respostas da *API* possibilitou ajustes na interface para melhor visualização dos dados recebidos. Outro aspecto importante dessas ferramentas é a possibilidade de salvar coleções de requisições, o que facilitou a documentação da *API*.

O conhecimento adquirido sobre foi de grande importância para o desenvolvimento do projeto, pois permitiu um melhor entendimento sobre comunicação entre *frontend* e *backend*, bem como práticas de consumo de *APIs*. Essas ferramentas são amplamente utilizadas no mercado de tecnologia e seu domínio contribui significativamente para a atuação profissional em projetos que dependem de integração entre sistemas (Postman, 2024; Insomnia, 2024).

### 3 DESENVOLVIMENTO

O desenvolvimento da plataforma *web* Protege Meu Cerrado foi realizado com foco na construção do *frontend*, utilizando tecnologias modernas para garantir uma experiência fluida e responsiva ao usuário. A aplicação foi desenvolvida em *Next.js* e *React*, *frameworks* que possibilitam uma estrutura modular e otimizada para aplicações *web*. Além disso, foram utilizadas bibliotecas como *Bulma*, para estilização da interface, *Swiper*, para exibição dinâmica de conteúdos, *React-Input-Mask*, para manipulação de formulários, e *@react-google-maps/api*, para integração de mapas.

O projeto teve como premissa a construção de um *frontend* intuitivo, visando incentivar a participação dos usuários na denúncia de crimes ambientais. Dessa forma, a interface foi projetada com princípios de usabilidade, permitindo que qualquer pessoa consiga registrar uma ocorrência de forma rápida e simples. Durante o desenvolvimento, foram seguidas boas práticas de programação, como componentização, otimização de performance, controle de versão com *Git* e *GitHub*, além da realização de testes de integração utilizando *Insomnia* e *Postman*, garantindo a comunicação eficiente com o *backend*.

A seguir, serão apresentadas as principais etapas do desenvolvimento da aplicação, abrangendo desde a definição dos requisitos e configuração do ambiente, até a implementação das funcionalidades, testes e implantação final.

#### 3.1 DEFINIÇÃO DOS REQUISITOS E ARQUITETURA DO FRONTEND

O desenvolvimento do *frontend* da plataforma Protege Meu Cerrado exigiu um planejamento estruturado e alinhado com metodologias ágeis, a fim de garantir a entrega de um sistema funcional e intuitivo. Para isso, adotou-se a metodologia *Scrum*, amplamente utilizada no desenvolvimento de *software*, que permite a organização das tarefas em ciclos curtos de desenvolvimento chamados de *sprints*. Dessa forma, a entrega do conteúdo foi estruturada em pacotes semanais, cada um contendo funcionalidades específicas a serem planejadas, implementadas e revisadas.

Esse modelo de trabalho permitiu que, a cada semana, novas implementações fossem avaliadas e aprimoradas antes da iteração seguinte. Além

disso, proporcionou flexibilidade para realizar ajustes conforme surgissem desafios técnicos ou mudanças nas necessidades do projeto. Ao final de cada *sprint*, o progresso era revisado e, se necessário, novas estratégias eram aplicadas para otimizar o desenvolvimento.

A arquitetura do *frontend* foi projetada de forma modular e escalável, garantindo que os componentes pudessem ser reutilizados em diferentes partes da aplicação. Embora o projeto não tenha adotado um *Design System* formal, seguiu-se um conjunto de padrões visuais e técnicos para assegurar a coerência e a usabilidade da plataforma. O uso do *Next.js*, *React* e *TypeScript* possibilitou a criação de uma interface dinâmica e de alto desempenho, ao mesmo tempo em que melhorou a organização do código e a segurança na manipulação de dados.

A definição dos requisitos envolveu a identificação das necessidades dos usuários e das funcionalidades essenciais para o funcionamento da plataforma. Com base nisso, os requisitos foram categorizados em funcionais e não funcionais, garantindo que tanto a experiência do usuário quanto os aspectos técnicos fossem devidamente contemplados.

#### Requisitos Funcionais:

- Implementação de um sistema de denúncias ambientais, permitindo que os usuários relatem incidentes como desmatamento, queimadas ilegais e poluição;
- Cadastro e gerenciamento de denúncias, possibilitando o envio de imagens, descrições detalhadas e a geolocalização do evento relatado;
- Sistema de autenticação e controle de acesso baseado em papéis (*RBAC*), garantindo que diferentes tipos de usuários tenham acessos específicos:
  - Usuários comuns: podem realizar denúncias (incluindo denúncias anônimas) e acompanhar o status das denúncias enviadas;
  - Usuários pertencentes a uma autoridade ambiental: podem iniciar tratativas e acompanhar denúncias em andamento;
  - Administradores: podem cadastrar novas autoridades, vincular usuários a essas entidades e excluir denúncias quando necessário.
- Integração entre *frontend* e *backend* através de requisições *HTTP*, utilizando ferramentas como *Insomnia* e *Postman* para teste e validação das *APIs*.

#### Requisitos Não Funcionais:

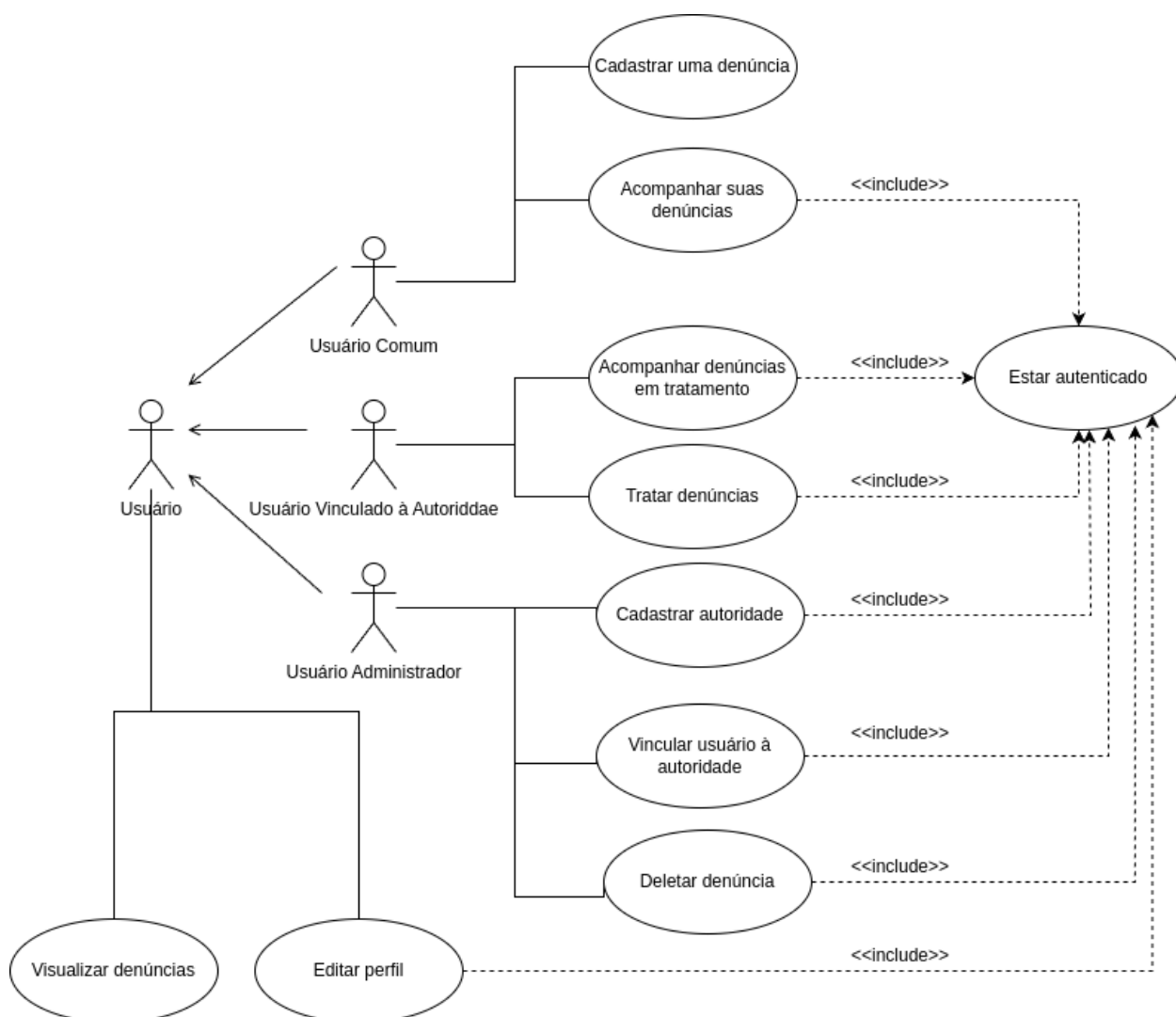
- Segurança: O sistema deve garantir a proteção dos dados pessoais dos

usuários, utilizando criptografia para senhas e informações sensíveis.

- Escalabilidade: A plataforma deve ser escalável, permitindo aumentar sua capacidade de processamento à medida que o número de usuários cresce.
- Usabilidade: A interface deve ser intuitiva e fácil de usar, garantindo que qualquer usuário, mesmo sem conhecimento técnico, possa registrar e acompanhar problemas ambientais.
- Compatibilidade: O sistema deve ser compatível com os principais navegadores de internet e acessível em dispositivos móveis (*smartphones* e *tablets*) e *desktops*.
- Utilização do *Next.js* para otimizar o carregamento das páginas.
- Uso de *TypeScript* para proporcionar maior segurança no código e evitar erros em tempo de execução.
- Adoção de componentes reutilizáveis com *React*, garantindo consistência visual e simplificação da manutenção do código.
- Uso do *Git* e *GitHub* para versionamento do código, permitindo controle e colaboração eficiente entre os desenvolvedores.

A definição desses requisitos foi essencial para direcionar o desenvolvimento e evitar a implementação de funcionalidades desnecessárias ou desalinhadas com os objetivos do projeto. Para melhor ilustrar as interações entre os usuários e o sistema, foi criado um diagrama de caso de uso, que detalha os principais atores e suas respectivas ações dentro da plataforma. Esse diagrama ajuda a visualizar as funcionalidades essenciais, facilitando a compreensão do comportamento esperado da aplicação. A figura a seguir apresenta o diagrama de caso de uso do projeto.

Figura 1 – Diagrama de caso de uso.



Fonte: Elaborado pelo autor, 2025

Dessa forma, a estruturação do desenvolvimento do *frontend* não apenas resultou em uma aplicação robusta, mas também proporcionou um aprendizado significativo sobre tecnologias modernas do mercado, boas práticas de desenvolvimento, trabalho em equipe e integração entre diferentes áreas do desenvolvimento de *software*. Essa abordagem permitiu consolidar conhecimentos valiosos e preparar para desafios reais no mercado de tecnologia.

### 3.2 CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

A configuração do ambiente de desenvolvimento é uma etapa crucial para garantir que todos os desenvolvedores trabalhem de forma consistente e eficiente.

No desenvolvimento do *frontend* da plataforma Protege Meu Cerrado, foi necessário estabelecer um ambiente adequado para o trabalho colaborativo, compatível com as tecnologias escolhidas e capaz de integrar todas as ferramentas necessárias para o desenvolvimento e testes da aplicação.

A escolha das ferramentas e tecnologias foi uma das primeiras decisões estratégicas a serem tomadas para o projeto. Com o objetivo de criar uma interface moderna, eficiente e de fácil manutenção, optou-se por tecnologias como *Next.js*, *React*, *TypeScript* e outras ferramentas como *Bulma* para estilização e *@react-google-maps/api* para a implementação do mapa dinâmico de denúncias. Essas ferramentas foram escolhidas por sua popularidade no mercado, pela robustez no desenvolvimento de aplicações *web*, e pela facilidade de integração entre elas.

A primeira etapa da configuração do ambiente envolveu a instalação do *Node.js* em sua versão mais recente, pois ele é fundamental para executar e gerenciar pacotes *JavaScript*, além de ser a base para o *Next.js* e outras dependências do projeto. Para gerenciar essas dependências, utilizou-se o *NPM* (*Node Package Manager*), que permitiu instalar as bibliotecas necessárias de forma eficiente. O *NPM* também possibilitou a atualização e o controle de versões das dependências ao longo do ciclo de vida do desenvolvimento.

Outro aspecto essencial na configuração foi a escolha das ferramentas de versionamento de código. O *Git*, aliado ao *GitHub*, foi adotado para o controle de versão do código-fonte, o que permitiu que toda a equipe trabalhasse de forma colaborativa, com a possibilidade de reverter alterações quando necessário, além de acompanhar o histórico de modificações. O *GitHub* também foi utilizado para centralizar o código-fonte e facilitar a integração contínua, além de possibilitar o gerenciamento das tarefas através de *issues* e *pull requests*, o que aumentou a produtividade da equipe.

Após a instalação das dependências essenciais, foi configurado o ambiente local de desenvolvimento. A principal ferramenta utilizada para o desenvolvimento foi o *VS Code* (*Visual Studio Code*), um editor de código leve, mas extremamente poderoso, com suporte a uma ampla gama de extensões, como a integração com *Git*, *Prettier* (para formatação automática do código) e *ESLint* (para garantir a qualidade do código e evitar erros de sintaxe). Essas extensões foram fundamentais para manter o código organizado, limpo e livre de erros.

Além disso, foram configurados ambientes de desenvolvimento para o *frontend* com ferramentas de *hot-reload* e *build* automatizado, garantindo que todas as mudanças feitas no código fossem refletidas em tempo real na aplicação, o que acelerou o processo de desenvolvimento e testes. Para isso, foi utilizado o próprio *Next.js*, que fornece recursos nativos para *hot-reloading* e otimização de *build*.

Embora não tenha sido realizado um processo formal de testes automatizados, foi adotada uma abordagem de testes manuais durante o desenvolvimento da plataforma. Esse processo envolveu a verificação contínua das funcionalidades implementadas, com testes realizados diretamente na interface da aplicação. A cada nova funcionalidade adicionada, a equipe realizava uma série de testes para validar se as interações e a experiência do usuário estavam funcionando conforme o esperado.

Esses testes manuais foram realizados no próprio ambiente de desenvolvimento local, com os desenvolvedores testando as funcionalidades. Além disso, a equipe também validava a integração entre o *frontend* e o *backend*, garantindo que as requisições de dados fossem realizadas corretamente e que a plataforma estivesse respondendo conforme as expectativas.

Para garantir que a comunicação entre o *frontend* e o *backend* estivesse funcionando corretamente, foram utilizadas ferramentas de teste de *APIs*, como o *Postman* e o *Insomnia*. Essas ferramentas permitiram simular as requisições *HTTP* e validar a resposta da *API* desenvolvida no *backend*, garantindo que a troca de dados entre as duas camadas estivesse funcionando de forma eficiente e sem erros.

A configuração do ambiente de desenvolvimento foi projetada para permitir um fluxo de trabalho eficiente, garantir a qualidade do código e facilitar a integração com o *backend* e outras ferramentas de apoio. Além disso, a constante realização de testes manuais assegurou que as funcionalidades da plataforma fossem verificadas e corrigidas, garantindo uma aplicação final funcional e alinhada aos requisitos propostos.

### 3.3 DESENVOLVIMENTO POR PACOTES

A metodologia utilizada para o desenvolvimento do *frontend* foi baseada em pacotes semanais de funcionalidades, seguindo o modelo ágil de desenvolvimento. Cada pacote contemplou uma parte específica da aplicação, permitindo que o



trabalho fosse progressivamente incrementado e ajustado conforme o *feedback* da equipe e a evolução do projeto. Abaixo, é apresentada uma explicação detalhada sobre o que foi feito em cada pacote, com foco no desenvolvimento da tela de denúncias, uma das funcionalidades centrais da plataforma.

### **3.3.1 Pacote 1: tela de denúncias**

O primeiro pacote de desenvolvimento foi voltado para a criação da tela de denúncias, uma das partes mais complexas e cruciais da plataforma Protege Meu Cerrado. A tela de denúncias foi dividida em três etapas, cada uma com sua própria funcionalidade, mas todas integradas em um fluxo contínuo para garantir uma experiência de usuário intuitiva e sem falhas.

A primeira etapa da tela de denúncias é dedicada ao preenchimento das informações pessoais do usuário. Nessa parte, o sistema realiza uma verificação para determinar se o usuário já está logado. Caso o usuário esteja autenticado, o sistema preenche automaticamente os campos com seus dados pessoais previamente cadastrados, como nome completo, *email*, CPF, telefone e data de nascimento. Caso o usuário não esteja logado, ele deve preencher esses dados manualmente para prosseguir.

Além disso, foi implementada uma funcionalidade que permite ao usuário, somente quando está logado, realizar uma denúncia anônima caso prefira não fornecer suas informações pessoais. Para isso, o sistema oferece a opção de ocultar os dados do usuário, mantendo a confidencialidade e proporcionando a possibilidade de denúncia sem identificar o denunciante.

A validação desses campos foi um ponto chave, visto que era necessário garantir que os dados inseridos fossem válidos antes de permitir o avanço para a próxima etapa. A principal preocupação foi fornecer uma experiência sem frustrações, por isso, foi implementada uma lógica de validação para cada campo. A figura abaixo exibe a primeira etapa da tela de denúncia.

Figura 2 – Etapa 1 da tela de denúncia.

Fonte: Elaborado pelo autor, 2025

A segunda etapa da tela de denúncias tem como objetivo coletar informações detalhadas sobre o ocorrido. Esse processo foi estruturado para garantir que todas as denúncias registradas contenham dados suficientes para uma análise precisa por parte das autoridades ambientais responsáveis. Nessa etapa, a interface do usuário foi projetada de forma intuitiva, guiando-o pelo preenchimento das informações de maneira fluida e objetiva.

O primeiro campo desta etapa é a data da denúncia, que é preenchida automaticamente pelo sistema com a data do dia atual e não pode ser alterada pelo usuário. Essa decisão de *design* foi tomada para garantir que todas as denúncias registradas reflitam a realidade do momento em que foram submetidas, evitando possíveis adulterações de data que poderiam comprometer a credibilidade das informações fornecidas.

Após isso, o usuário deve selecionar a categoria que melhor descreva a natureza da denúncia. Foram estabelecidas sete categorias principais para classificar os incidentes ambientais reportados:

- Queimadas e incêndios – Denúncias relacionadas a queimadas ilegais, incêndios criminosos ou incêndios naturais sem controle.
- Inundações e enchentes – Relatos de enchentes que impactam áreas

naturais ou urbanas, agravadas por ações humanas.

- Exploração e atividades ilegais – Denúncias de desmatamento ilegal, caça predatória, exploração irregular de recursos naturais, entre outros.
- Poluição hídrica – Problemas relacionados à contaminação de rios, lagos e outros corpos d'água devido ao despejo de resíduos industriais, esgoto não tratado ou outros poluentes.
- Poluição do solo – Relatos de descarte inadequado de resíduos sólidos, contaminação de solos por produtos químicos e outros problemas ambientais relacionados ao solo.
- Poluição do ar – Denúncias de emissão irregular de poluentes por indústrias, veículos ou outras fontes que afetam a qualidade do ar.
- Ameaça à fauna – Situações em que animais silvestres estejam sendo ameaçados por ações humanas, como tráfico de animais, caça ilegal ou destruição de habitats naturais.

A escolha da categoria é obrigatória e influencia diretamente a forma como as denúncias serão tratadas pelas autoridades competentes. Para garantir uma experiência fluida, foi implementado um sistema de seleção intuitivo, onde o usuário pode clicar em uma opção pré-definida sem necessidade de digitar informações adicionais.

Em seguida, o usuário deve fornecer uma descrição detalhada da denúncia, onde poderá explicar o ocorrido com mais profundidade. Esse campo tem um papel essencial, pois permite que o denunciante forneça informações adicionais que não foram contempladas na categorização anterior.

Por fim, o usuário pode anexar até três imagens que auxiliem na comprovação da denúncia. A adição de imagens é uma funcionalidade importante, pois permite que as autoridades tenham uma visão mais clara da situação reportada.

Essa segunda etapa foi projetada para equilibrar praticidade e eficiência, garantindo que os usuários pudessem fornecer as informações necessárias sem enfrentar dificuldades no preenchimento dos campos. A validação de cada entrada foi essencial para evitar erros e garantir que todas as denúncias cadastradas estivessem devidamente estruturadas, permitindo um processamento mais ágil por parte do sistema e das autoridades responsáveis pelo tratamento dos casos

ambientais. A Figura 3 apresenta a etapa 2 da tela de denúncia.

Figura 3 – Etapa 2 da tela de denúncia.

Fonte: Elaborado pelo autor, 2025

A terceira e última etapa da tela de denúncias tem o objetivo de registrar a localização exata do incidente relatado. Essa informação é fundamental para que as autoridades ambientais possam analisar a ocorrência e tomar as medidas cabíveis de forma eficaz. Para tornar esse processo intuitivo, a interface permite que o usuário escolha entre duas formas de inserir a localização: utilizar sua localização atual ou selecionar manualmente um ponto no mapa.

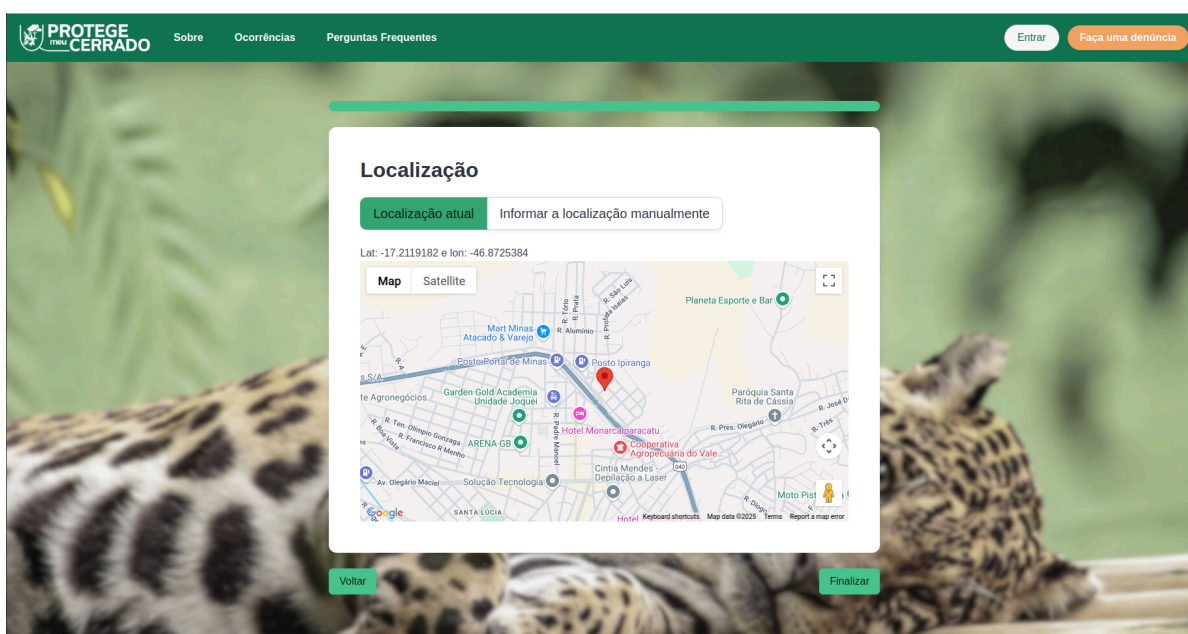
Se o usuário optar por usar sua localização atual, o sistema acessa as coordenadas do dispositivo por meio da *API* de geolocalização do navegador. Isso permite que a latitude e longitude sejam preenchidas automaticamente, facilitando o preenchimento para o denunciante. Para garantir a privacidade dos usuários, o sistema solicita permissão antes de acessar essa informação, conforme as diretrizes de segurança da *web*.

Caso prefira, o usuário pode selecionar a localização diretamente no mapa, arrastando e soltando um marcador no ponto exato do incidente. Para isso, foi utilizada a biblioteca *@react-google-maps/api*, que oferece uma interface interativa e responsiva. Além disso, ao clicar em um ponto do mapa, o sistema captura automaticamente as coordenadas geográficas, preenchendo os campos de latitude

e longitude. Esse método é especialmente útil quando o usuário deseja reportar um problema ambiental que ocorreu em um local diferente de onde ele está no momento da denúncia.

Além das coordenadas, o sistema exibe um campo de endereço, que é preenchido automaticamente com base na localização escolhida. Essa funcionalidade utiliza o serviço de geocodificação da *Google Maps API*, convertendo as coordenadas selecionadas em um endereço textual. Caso necessário, o usuário pode editar esse campo para fornecer um endereço mais detalhado ou corrigir eventuais imprecisões. A figura a seguir exibe a etapa 3 da tela de denúncia.

Figura 4 – Etapa 3 da tela de denúncia.



Fonte: Elaborado pelo autor, 2025

Além de preencher as informações obrigatórias em cada etapa, o usuário pode transitar livremente entre elas, permitindo revisar ou corrigir dados antes do envio final. A navegação entre as etapas é facilitada por botões intuitivos que permitem avançar e retroceder, garantindo que o denunciante tenha total controle sobre as informações fornecidas. Essa abordagem melhora a experiência do usuário, tornando o processo de denúncia mais confiável.

Após revisar todas as informações, o usuário pode enviar a denúncia, momento em que os dados são transmitidos ao *backend* para serem armazenados

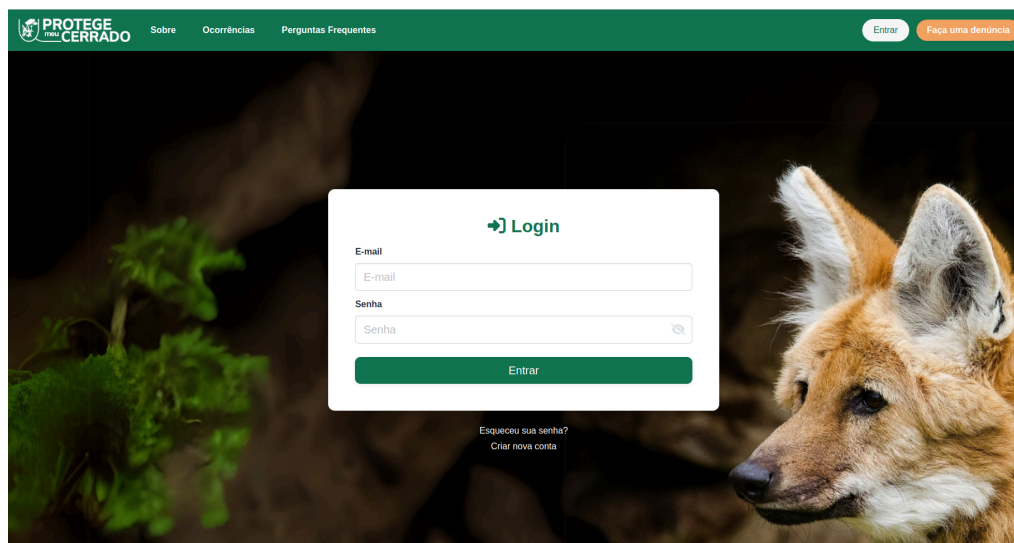
e disponibilizados para as autoridades responsáveis. Para garantir que nenhuma informação essencial fique de fora, o sistema realiza validações antes do envio, alertando o usuário caso algum campo obrigatório não tenha sido preenchido corretamente.

O cadastro de denúncias foi a primeira funcionalidade desenvolvida no *frontend* da plataforma Protege Meu Cerrado, pois representa o cerne da aplicação e sua principal utilidade. A implementação dessa funcionalidade exigiu um planejamento detalhado para garantir usabilidade e integração eficiente com o *backend* e *APIs* externas, como o *Google Maps*.

### **3.3.2 Pacote 2: Cadastro de usuário e listagem de ocorrências**

O segundo pacote de desenvolvimento da plataforma Protege Meu Cerrado teve como foco a implementação do cadastro e autenticação de usuários, além da criação da tela de listagem de ocorrências. Essas funcionalidades são essenciais para estruturar o controle de acesso ao sistema, garantindo que apenas usuários autenticados possam interagir com a plataforma e que as denúncias registradas sejam expostas para todos, independentemente de estarem ou não logados.

A tela de *login* permite que os usuários acessem a plataforma informando suas credenciais, como *email* e senha. Para garantir uma experiência segura e intuitiva, diversas validações foram implementadas, incluindo mensagens de erro em tempo real caso os dados inseridos estejam incorretos. Além disso, a persistência da sessão foi aplicada, permitindo que o usuário permaneça autenticado ao retornar à plataforma. A implementação dessa funcionalidade é essencial para garantir um ambiente seguro e confiável, onde somente usuários previamente cadastrados podem acessar as funcionalidades exclusivas da plataforma. E essa tela pode ser vista na Figura 5.

Figura 5 – Tela de *login*.

Fonte: Elaborado pelo autor, 2025

Para possibilitar o registro de novos usuários, a tela de cadastro foi criada, exigindo o preenchimento de informações obrigatórias, como nome completo, *email*, CPF, telefone, data de nascimento e senha. Um sistema de validação rigoroso foi desenvolvido para assegurar a integridade dos dados e evitar erros. Um validador de CPF foi implementado, garantindo que apenas CPFs legítimos sejam aceitos, reforçando a segurança da plataforma. Além disso, a senha precisa atender a critérios de segurança específicos, como tamanho mínimo e presença de caracteres variados, para evitar vulnerabilidades. A tela de cadastro é um dos elementos mais importantes deste pacote, pois estabelece a base para a organização dos usuários dentro da plataforma, permitindo que cada um tenha um perfil único e autenticado. A Figura 6 exhibe a tela de cadastro de usuário.



Figura 6 – Tela de cadastro de usuário.

**PROTEGE MATA CERRADO** Sobre Ocorrências Perguntas Frequentes Entrar Faça uma denúncia

### Nova Conta

E-mail\*  
administrador@administrador.com.br

Nome\* Digite seu nome CPF\* Digite seu CPF

Telefone\* Digite seu telefone Data de nascimento\* Digite sua data de nascimento

Senha\* Senha Confirmar senha\* Digite sua senha

**Criar Conta**

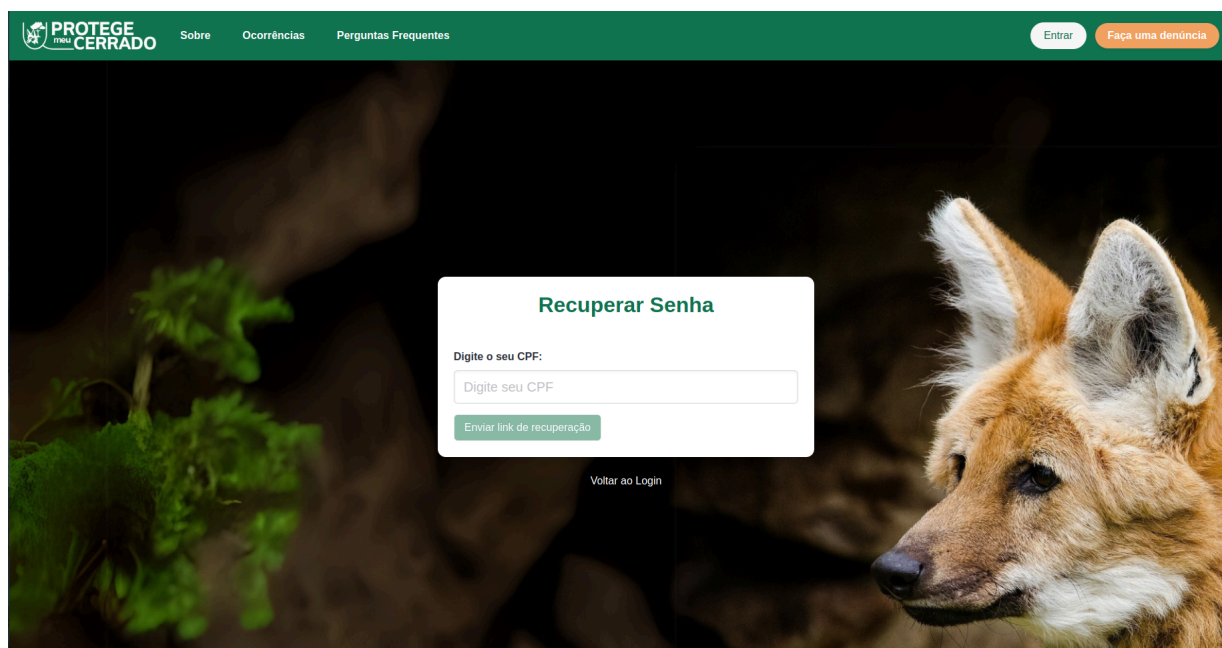
Já tem uma conta? Fazer Login

Fonte: Elaborado pelo autor, 2025

Além do *login* e do cadastro, a funcionalidade de recuperação de senha foi desenvolvida para permitir que os usuários redefinam sua senha caso a esqueça e a tela desta funcionalidade está sendo apresentada pela Figura 6. O processo solicita que o usuário informe o seu CPF e a partir dele um link é enviado para o *email* cadastrado deste usuário para que ele possa recuperar a sua conta.



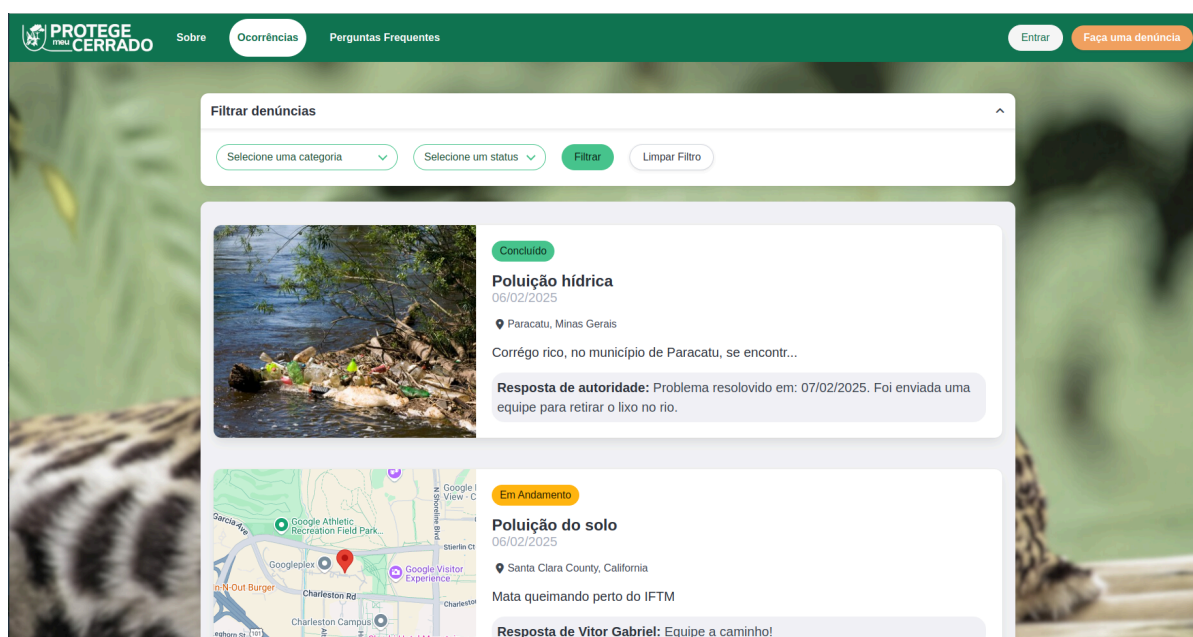
Figura 7 – Tela de recuperação de conta.



Fonte: Elaborado pelo autor, 2025

Outra funcionalidade essencial desse pacote é a tela de listagem de ocorrências, que exibe todas as denúncias cadastradas na plataforma. A interface apresenta informações como categoria, data de registro, localização, descrição e status, que pode indicar se a denúncia está pendente, em andamento ou concluída. Além disso, caso a denúncia já apresente uma resposta de uma autoridade, essa resposta também é exibida. Para melhorar a experiência do usuário, um sistema de filtragem foi implementado, permitindo que as ocorrências sejam organizadas por categoria ou *status*, facilitando a busca por informações específicas. Ao clicar em uma denúncia, o usuário pode visualizar um card expandido contendo todos os detalhes da ocorrência, incluindo uma descrição completa e as imagens anexadas no momento do registro. Essa funcionalidade é essencial para garantir que as denúncias sejam acessíveis de forma organizada, tornando a plataforma mais eficiente na apresentação das informações. A figura a seguir mostra a tela descrita.

Figura 8 – Tela de listagem de denúncias.



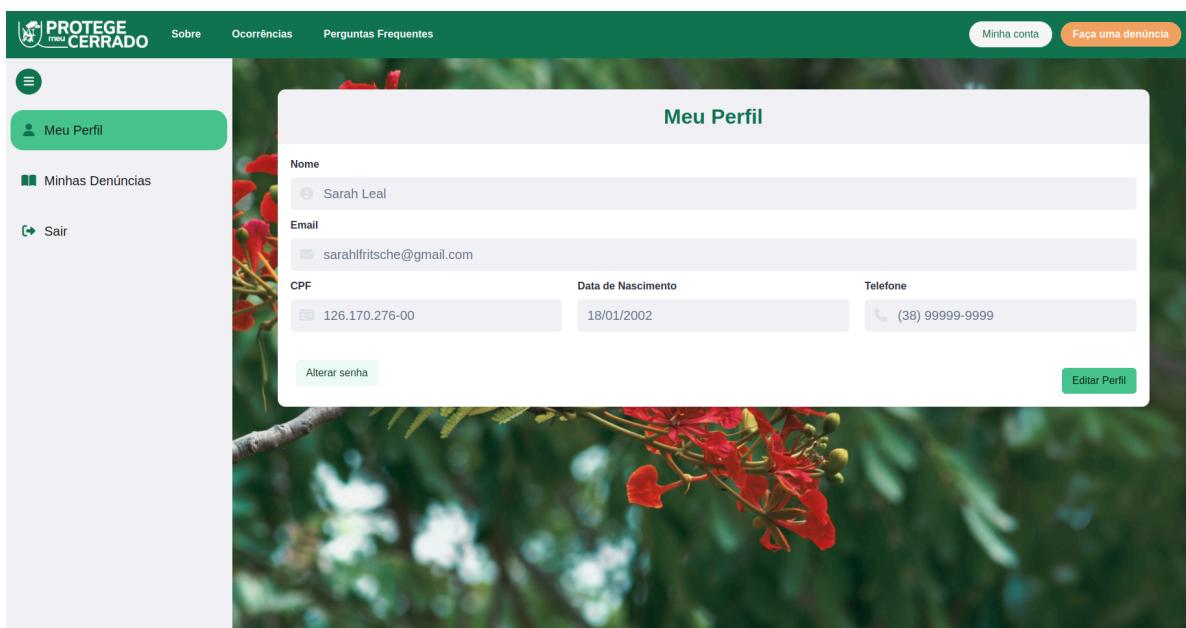
Fonte: Elaborado pelo autor, 2025

A implementação dessas funcionalidades representa um grande avanço no desenvolvimento da plataforma, pois não apenas estabelece um sistema seguro de autenticação e gerenciamento de usuários, mas também possibilita a exibição estruturada das denúncias registradas. Com a conclusão desse pacote, a plataforma passa a contar com uma base sólida para controle de acesso e organização das ocorrências, preparando o sistema para implementações que permitem maior interação entre os usuários e as denúncias cadastradas.

### 3.3.3 Pacote 3: *Dashboard* para usuário comum

O terceiro pacote de desenvolvimento da plataforma Protege Meu Cerrado foca na implementação do *dashboard* do usuário comum, permitindo que ele gerencie suas informações pessoais e acompanhe as denúncias registradas. Para isso, foi necessário desenvolver um sistema de autenticação robusto, garantindo que apenas usuários logados possam acessar essa área e interagir com suas próprias denúncias. O *dashboard* apresenta um menu lateral, onde o usuário pode editar seus dados pessoais, como nome, CPF, data de nascimento, telefone, *email* e senha, além de visualizar as denúncias feitas e realizar *logout*. A tela de edição de perfil está representada na Figura 9.

Figura 9 – Tela de edição de perfil.



Fonte: Elaborado pelo autor, 2025

A criação dessa área demandou um planejamento estruturado, principalmente para garantir a persistência da autenticação do usuário. Ao acessar o sistema, as informações do usuário precisam ser carregadas automaticamente, permitindo que ele veja seus dados sem precisar digitá-los novamente a cada sessão. Para isso, foi necessário implementar um contexto de autenticação no *Next.js*, garantindo que os dados do usuário estivessem acessíveis em qualquer parte da aplicação sem a necessidade de repetições desnecessárias de chamadas à *API*.

O *AutenticacaoContext* foi criado justamente para resolver esse problema. Em aplicações *React* e *Next.js*, o uso de um contexto global permite centralizar estados e compartilhá-los entre diversos componentes sem precisar passar propriedades manualmente. No caso do sistema, o contexto gerencia o estado do usuário autenticado, permitindo verificar se ele está logado, recuperar seus dados e definir suas permissões dentro da plataforma. Sem o uso desse contexto, cada componente que necessitasse de informações do usuário teria que fazer requisições separadas ao *backend*, o que aumentaria a complexidade do código e poderia impactar negativamente o desempenho da aplicação.

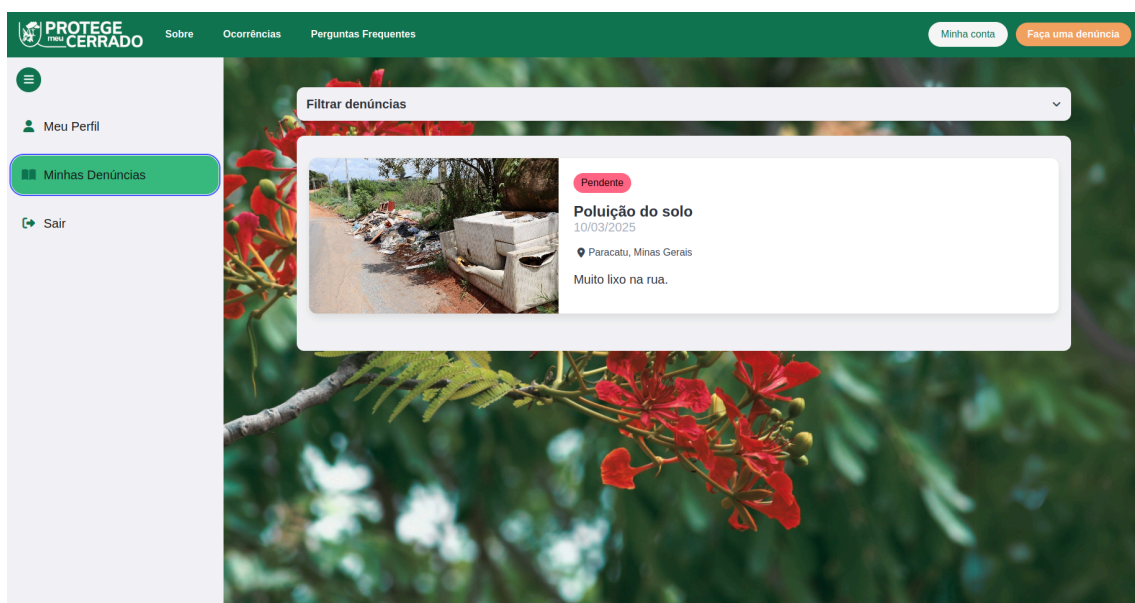
Dentro do *AutenticacaoContext*, existem algumas funções fundamentais para o funcionamento do sistema. A função *login* é responsável por armazenar os dados de autenticação do usuário no *localStorage*, garantindo que, ao recarregar a

página, ele continue logado sem precisar inserir suas credenciais novamente. O *logout*, por sua vez, remove essas informações do armazenamento local, finalizando a sessão do usuário e garantindo que ele não tenha mais acesso às funcionalidades restritas. Além disso, há uma lógica específica para determinar o tipo de usuário, visto que dentro da plataforma existem diferentes perfis com permissões distintas (usuário comum, usuário vinculado à autoridade ambiental e administrador).

A aplicação do `AutenticacaoContext` é feita envolvendo a aplicação dentro do `AutenticacaoProvider`, que encapsula toda a lógica de autenticação e fornece os dados necessários para os componentes que precisam dessa informação. Qualquer componente da aplicação pode acessar os dados do usuário logado utilizando o *hook* personalizado `useAutenticacaoContext`, o que torna a estrutura do código mais organizada e eficiente. Essa abordagem evita a necessidade de passar *props* manualmente por diversos níveis de componentes e garante que os dados do usuário estejam sempre atualizados e acessíveis onde for necessário.

Além da autenticação, outro ponto essencial desse pacote foi a implementação da listagem de denúncias feitas pelo usuário. Dentro do *dashboard*, o usuário pode visualizar todas as denúncias que ele mesmo registrou, acompanhando seu status e verificando se alguma autoridade ambiental já iniciou uma tratativa. Cada denúncia é apresentada em formato de *card*, onde são exibidas informações como a categoria, a data do registro, o *status* e a descrição. Isso proporciona uma experiência mais intuitiva e facilita o acompanhamento das denúncias. A Figura 10 apresenta a tela de acompanhamento de denúncias localizada no *dashboard* de um usuário comum.

Figura 10 – Tela de acompanhamento de denúncias.



Fonte: Elaborado pelo autor, 2025

Essa implementação representou um avanço significativo no desenvolvimento da plataforma, pois consolidou a estrutura de autenticação e gerenciamento de usuários, preparando o sistema para a introdução de novas funcionalidades. Com um fluxo de autenticação bem definido e um *dashboard* funcional, a plataforma se torna mais completa, oferecendo aos usuários um ambiente seguro e eficiente para registrar e acompanhar suas denúncias ambientais. Além disso, a modularização do código com o uso do contexto de autenticação permite escalabilidade, facilitando a futura adição de novos tipos de usuários e funcionalidades sem comprometer a organização e o desempenho da aplicação.

### 3.3.4 Pacote 4: *Dashboard* para usuário vinculados à autoridades

O quarto pacote de desenvolvimento da plataforma Protege Meu Cerrado focou na implementação do *dashboard* para usuários vinculados a uma autoridade ambiental. Essa funcionalidade foi essencial para garantir que as denúncias feitas na plataforma pudessem ser tratadas de maneira organizada e eficiente, permitindo que as autoridades responsáveis tivessem um espaço dedicado para acompanhar e gerenciar as denúncias sob sua responsabilidade.

Esse *dashboard* compartilha algumas funcionalidades com o usuário comum, como a possibilidade de editar suas informações pessoais (nome, CPF, data de

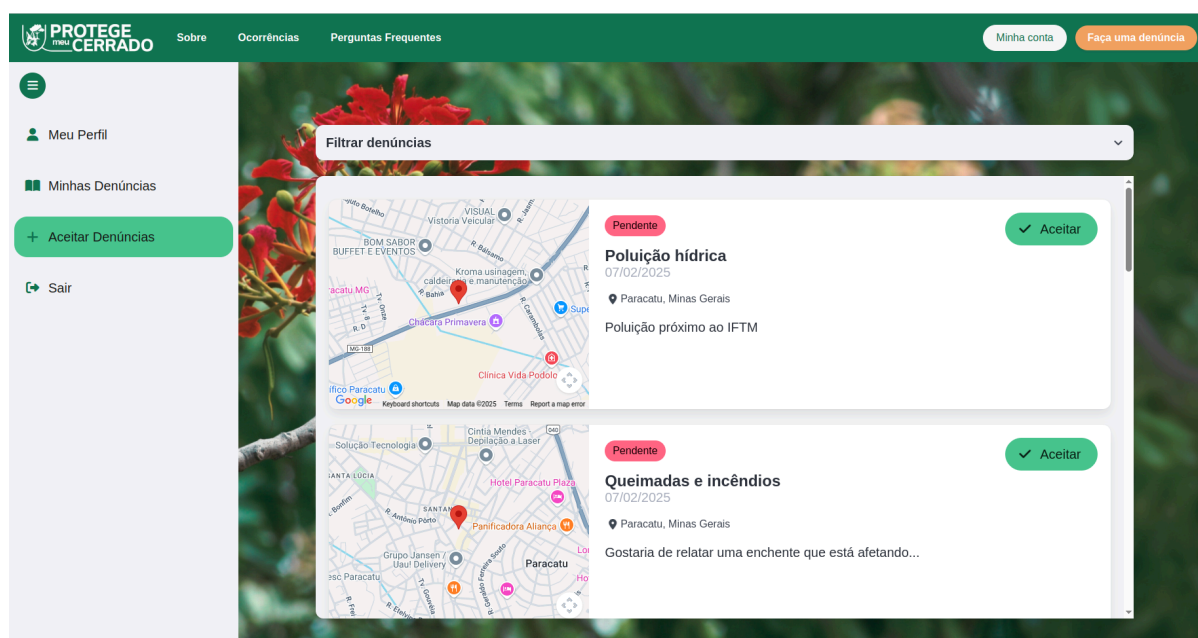


nascimento, telefone, *email* e senha) e realizar *logout*. No entanto, além dessas funções básicas, o painel de uma autoridade ambiental introduz novas capacidades relacionadas ao gerenciamento e acompanhamento das denúncias ambientais.

Ao acessar a plataforma, um usuário vinculado a uma autoridade ambiental pode visualizar duas categorias principais de gerenciamento de denúncias:

- Aceitar denúncias: exibe todas as denúncias recém-registradas na plataforma, ainda sem nenhuma tratativa associada, como demonstrado na Figura 11.

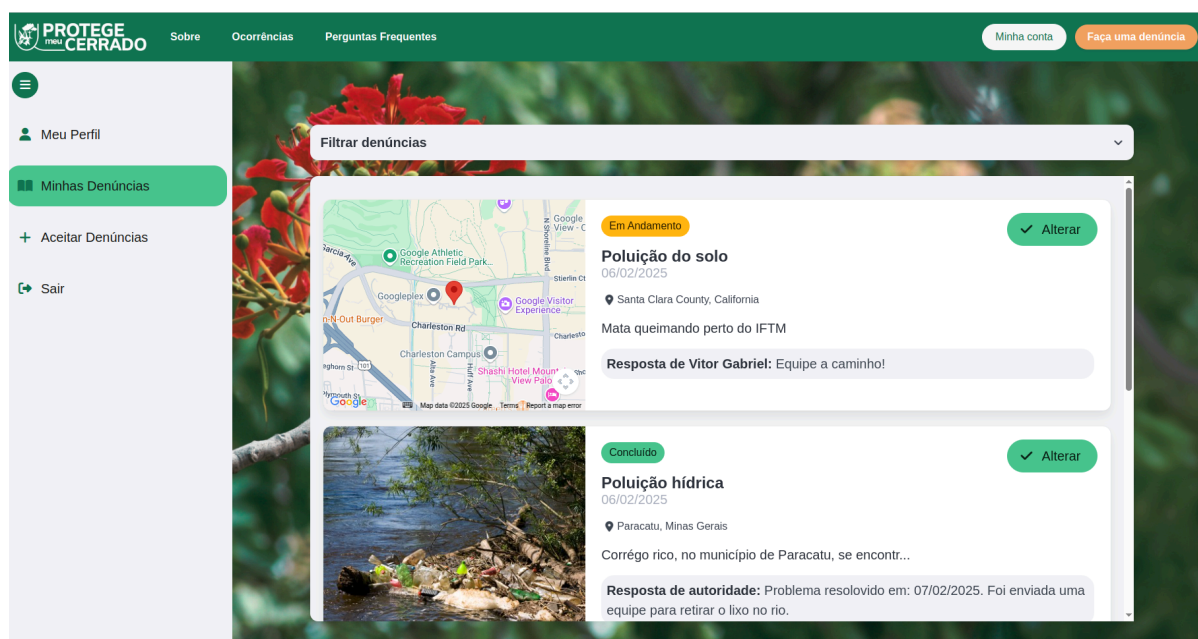
Figura 11 – Tela de aceitar denúncias.



Fonte: Elaborado pelo autor, 2025

- Minhas denúncias: exibe as denúncias que já estão sendo tratadas por alguma autoridade e as denúncias que já foram concluídas, como demonstrado na Figura 12.

Figura 12 – Tela de acompanhar denúncias em tratamento.



Fonte: Elaborado pelo autor, 2025

A principal funcionalidade deste *dashboard* é permitir que a autoridade ambiental selecione uma denúncia pendente para iniciar sua tratativa. Ao fazer isso, o sistema redireciona o usuário para uma página específica, onde ele pode visualizar todos os detalhes da denúncia, incluindo a categoria, a data do registro, a descrição, as imagens anexadas e a localização do problema. Nesse espaço, o usuário da autoridade pode adicionar uma descrição detalhada sobre a tratativa, explicando as ações tomadas para resolver o problema. Além disso, ele pode e deve alterar o *status* desta denúncia para uma denúncia “em andamento” ou “concluída” (caso o problema já tenha sido resolvido). A tela responsável por adicionar a tratativa da autoridade está sendo exibida na Figura 13.

Figura 13 – Tela de adicionar uma tratativa à denúncia.

The screenshot displays the 'Protege Meu Cerrado' interface. On the left is a sidebar with navigation links: 'Meu Perfil', 'Minhas Denúncias', '+ Aceitar Denúncias', and 'Sair'. The main content area is titled 'Ocorrência #4' with the category 'Poluição hídrica'. It shows the date '07/02/2025', description 'Poluição próximo ao IFMT', location 'Paracatu, Minas Gerais', and coordinates '-17.2386812, -46.8625957'. A map view is included with 'Map' and 'Satellite' toggles. Below the map, the 'Adicionar Tratativa' section contains a text input field for a message, an 'Alterar Status' dropdown menu currently set to 'Selecione um status', and 'Salvar' and 'Cancelar' buttons.

Fonte: Elaborado pelo autor, 2025

Além de iniciar novas tratativas, a autoridade ambiental pode alterar o *status* de uma denúncia que já está em andamento, na aba de “Minhas Denúncias”. Caso a denúncia tenha sido resolvida, é possível marcá-la como concluída. Entretanto, se for necessário reavaliar o caso ou tomar novas medidas, a plataforma permite que uma denúncia que já foi concluída volte para o estado de em andamento. Essa flexibilidade garante que a gestão das denúncias seja dinâmica e adaptável às necessidades do processo de fiscalização e resolução dos problemas ambientais.

A implementação desse *dashboard* exigiu um planejamento detalhado, principalmente para garantir que apenas usuários autorizados pudessem acessar e modificar as denúncias. Para isso, o sistema verifica o tipo de usuário autenticado e bloqueia o acesso a funcionalidades restritas para aqueles que não possuem permissão. Além disso, foi fundamental garantir uma experiência fluida para o usuário, permitindo que as denúncias fossem filtradas por *status* e/ou categoria e organizadas de forma intuitiva.

Esse pacote representou um grande avanço na plataforma, pois estabeleceu um fluxo claro para o tratamento das denúncias ambientais, tornando o sistema mais eficiente e funcional. Com autoridades ambientais podendo assumir e acompanhar as denúncias, a plataforma Protege Meu Cerrado se aproxima ainda mais do seu objetivo de facilitar o monitoramento e a resolução de problemas



ambientais no cerrado brasileiro.

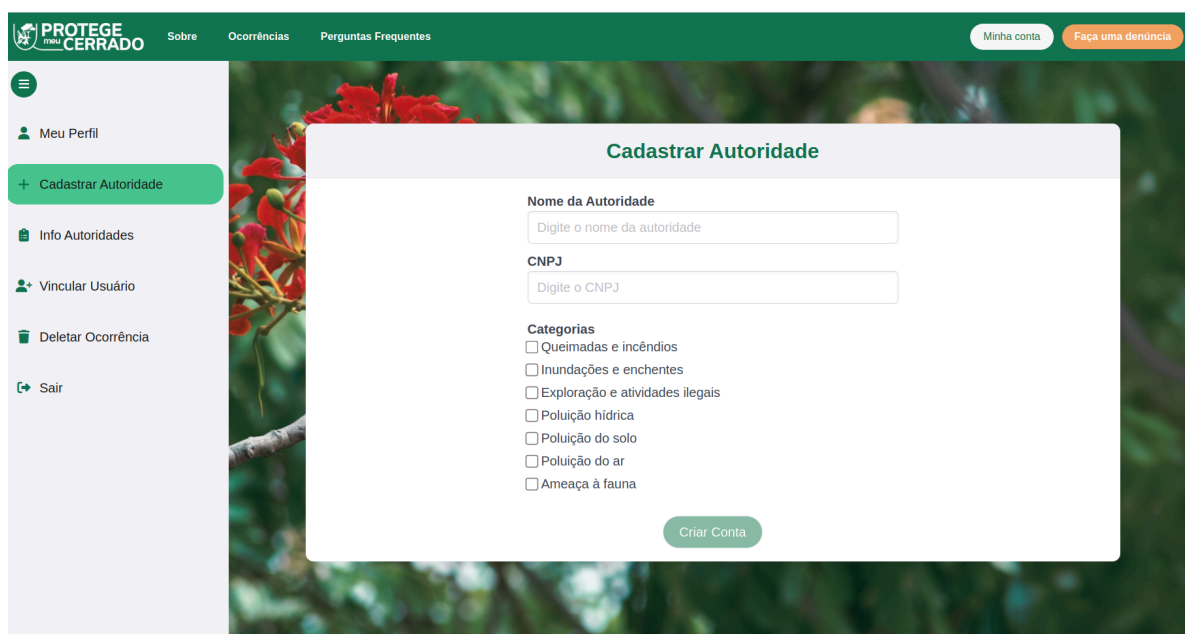
### 3.3.5 Pacote 5: *Dashboard* para usuário administrador

O quinto pacote de desenvolvimento da plataforma Protege Meu Cerrado foi dedicado à criação do *dashboard* para usuários administradores. Esse painel trouxe funcionalidades exclusivas para a gestão do sistema, permitindo que o administrador tivesse um controle mais amplo sobre autoridades ambientais, usuários e denúncias.

Assim como nos outros *dashboards*, o administrador pode editar suas informações pessoais (nome, CPF, data de nascimento, telefone, *email* e senha) e realizar *logout* a qualquer momento. No entanto, o grande diferencial desse painel está nas ferramentas voltadas para a gestão do sistema.

Uma das principais funcionalidades implementadas foi a criação de uma autoridade ambiental. Nessa tela, o administrador pode cadastrar novas autoridades responsáveis por tratar denúncias dentro da plataforma. O cadastro exige informações essenciais sobre a autoridade, garantindo que os usuários vinculados a ela possam acessar o sistema e desempenhar suas funções corretamente. A Figura 14 demonstra esta tela.

Figura 14 – Tela de cadastrar autoridade.



A imagem mostra a interface de usuário para o cadastro de uma autoridade ambiental. No topo, há uma barra de navegação verde com o logo "PROTEGE MEU CERRADO" e links para "Sobre", "Ocorrências" e "Perguntas Frequentes". À direita da barra, há botões para "Minha conta" e "Faça uma denúncia". No lado esquerdo, um menu lateral contém opções como "Meu Perfil", "Cadastrar Autoridade" (destacado em verde), "Info Autoridades", "Vincular Usuário", "Deletar Ocorrência" e "Sair". O formulário principal, intitulado "Cadastrar Autoridade", está sobreposto a uma imagem de fundo de flores vermelhas. Ele contém campos de texto para "Nome da Autoridade" e "CNPJ", e uma seção "Categorias" com seis opções de seleção por meio de caixas de seleção: "Queimadas e incêndios", "Inundações e enchentes", "Exploração e atividades ilegais", "Poluição hídrica", "Poluição do solo" e "Poluição do ar". A opção "Ameaça à fauna" não possui uma caixa de seleção visível. Um botão verde "Criar Conta" está localizado na base do formulário.

Fonte: Elaborado pelo autor, 2025

Além de cadastrar novas autoridades, o administrador também pode vincular usuários a uma autoridade específica. Para isso, foi desenvolvida uma interface intuitiva que exibe todas as autoridades cadastradas no sistema e todos os usuários ativos. O administrador pode selecionar uma autoridade e um usuário, e com um simples clique no botão “Vincular”, estabelecer a relação entre eles. Essa funcionalidade é crucial para garantir que as denúncias sejam tratadas por profissionais devidamente associados a uma entidade responsável. A tela de vincular usuário está representada na figura abaixo.

Figura 15 – Tela de vincular usuário.

**PROTEGE meu CERRADO** Sobre Ocorrências Perguntas Frequentes [Minha conta](#) [Faça uma denúncia](#)

**Vincular Usuário**

Selecione a autoridade:

Instituto Estadual de Florestas de Minas Gerais (IEF)

Nome Empresarial	CNPJ	Categorias
Instituto Estadual de Florestas de Minas Gerais (IEF)	3389126000198	Queimadas e incêndios Inundações e enchentes Exploração e atividades ilegais

Selecione usuário:

Admin

Nome	CPF	Email
Admin	701.809.046-85	administrado@administrador.com.br

**Vincular**

Fonte: Elaborado pelo autor, 2025

Outra funcionalidade relevante dentro desse *dashboard* é a possibilidade de visualizar informações detalhadas sobre uma autoridade ambiental. Ao acessar essa tela, o administrador pode ver quais usuários estão vinculados a uma determinada autoridade. A tela de informações detalhadas está representada na Figura 16.

Figura 16 – Tela de informações de autoridades.

**Informações de Autoridades**

Selecione a autoridade:

Instituto Estadual de Florestas de Minas Gerais (IEF)

Dados:

Nome Empresarial	CNPJ	Categorias
Instituto Estadual de Florestas de Minas Gerais (IEF)	3389126000198	Queimadas e incêndios Inundações e enchentes Exploração e atividades ilegais

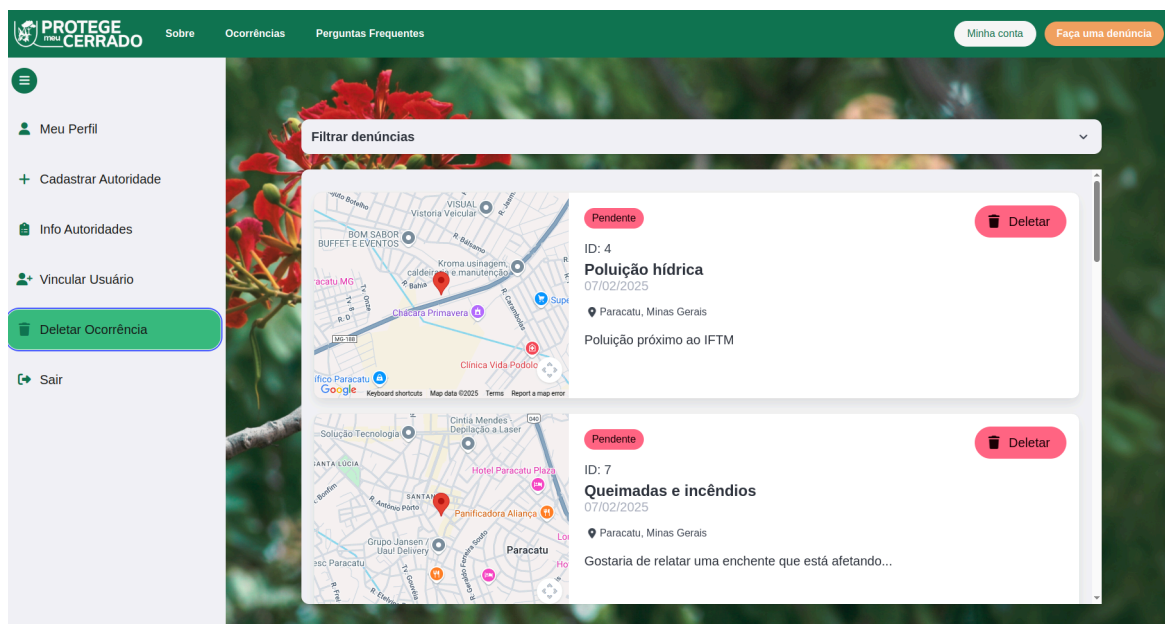
Usuários vinculados à esta autoridade:

Usuário	CPF	Telefone	Email
Vitor Gabriel	55882948070	38999375403	Vitor@gmail.com
administrador	153.122.111-11	(38)99999-9999	administrador@administrador.com.br
Vitor Gabriel Siega	123.456.789-99	(38)99988-7766	vitorgsiega@gmail.com

Fonte: Elaborado pelo autor, 2025

Por fim, um recurso exclusivo desse painel é a exclusão de denúncias pendentes. Essa funcionalidade foi pensada para situações em que usuários possam registrar denúncias falsas ou inapropriadas com o objetivo de prejudicar a plataforma ou gerar ruído nas operações das autoridades. A exclusão de denúncias é um privilégio exclusivo do administrador e foi implementada com um fluxo de confirmação, garantindo que apenas denúncias realmente problemáticas sejam removidas do sistema. A interface da tela que contém essa funcionalidade pode ser vista na Figura 17.

Figura 17 – Tela de deletar denúncias.



Fonte: Elaborado pelo autor, 2025

A construção desse painel foi essencial para garantir a organização da plataforma e a segurança do fluxo de denúncias. A implementação dessas funcionalidades exigiu um cuidado especial com controle de permissões, garantindo que apenas administradores pudessem acessar e modificar informações críticas do sistema. Com esse painel, a Protege Meu Cerrado ganhou mais robustez, assegurando que o sistema funcione de maneira eficiente e organizada, com a supervisão necessária para manter a integridade dos dados e a credibilidade da plataforma.

### 3.3.6 Pacote 6: Criação da tela inicial e perguntas frequentes

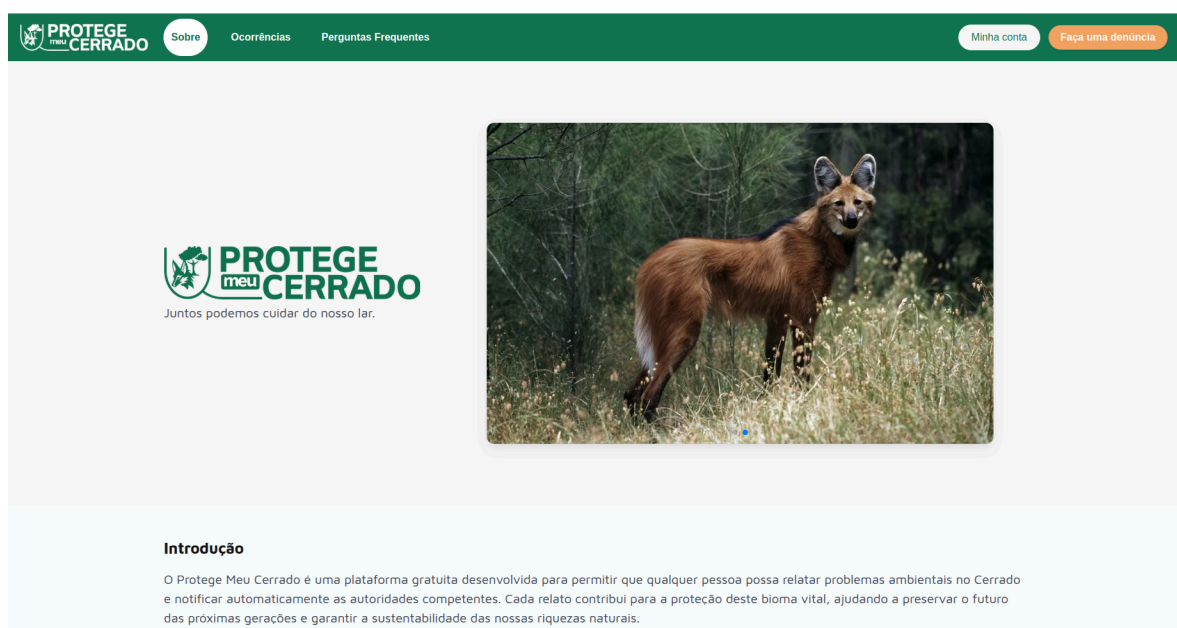
O pacote seis foi um marco importante no desenvolvimento da plataforma Protege Meu Cerrado, pois trouxe a criação da tela inicial, que funciona como o cartão de visitas do sistema, além da implementação da página de perguntas frequentes, termos de uso e política de privacidade. Com essas adições, a plataforma ganhou uma estrutura mais completa e profissional, proporcionando aos usuários um ambiente mais informativo logo no primeiro contato.

A tela inicial foi um dos elementos centrais deste pacote. Ela foi projetada para ser atrativa, moderna e funcional, servindo como uma introdução ao projeto Protege Meu Cerrado. O objetivo era garantir que qualquer visitante que

acesse o site entendesse rapidamente do que se trata a plataforma, qual sua finalidade e como utilizá-la. Para isso, foram adicionadas seções explicativas sobre a missão do projeto, a importância da denúncia de crimes ambientais e os benefícios de participar ativamente dessa iniciativa. A página foi estruturada de forma intuitiva, utilizando imagens impactantes relacionadas ao meio ambiente, elementos visuais modernos e chamadas para ação bem posicionadas, incentivando o usuário a se cadastrar e começar a interagir com o sistema.

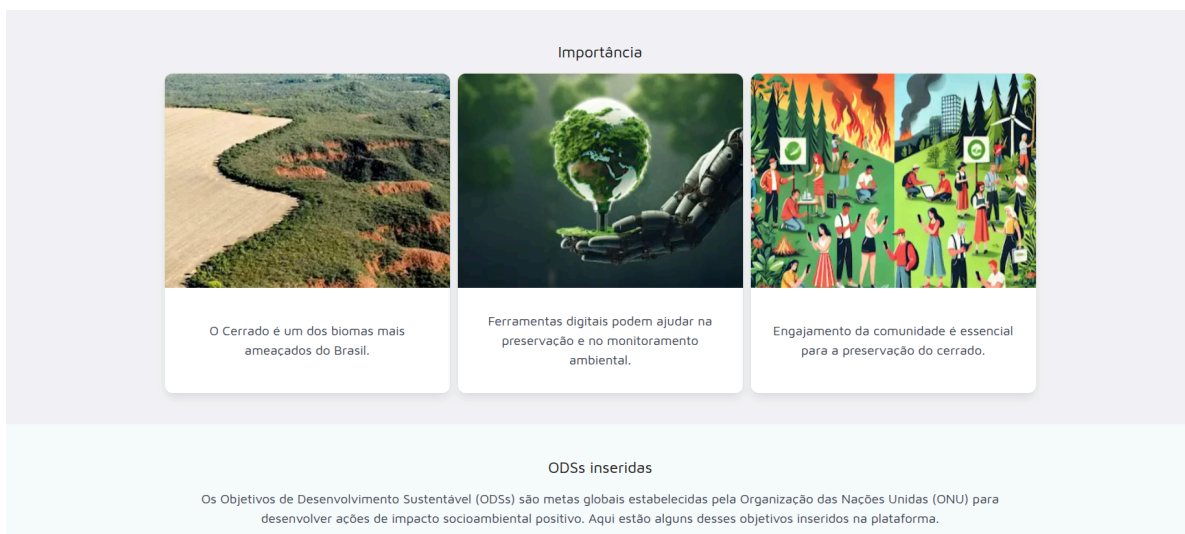
Além do design, a *UX* (experiência do usuário) foi um fator essencial no desenvolvimento da tela inicial. O *design* responsivo permitiu que a página se ajustasse automaticamente a *smartphones*, *tablets* e *desktops*, proporcionando uma experiência consistente independentemente do dispositivo utilizado. As figuras a seguir exibem a tela inicial.

Figura 18 – Tela inicial - parte 1.



Fonte: Elaborado pelo autor, 2025

Figura 19 – Tela inicial - parte 2.



Fonte: Elaborado pelo autor, 2025

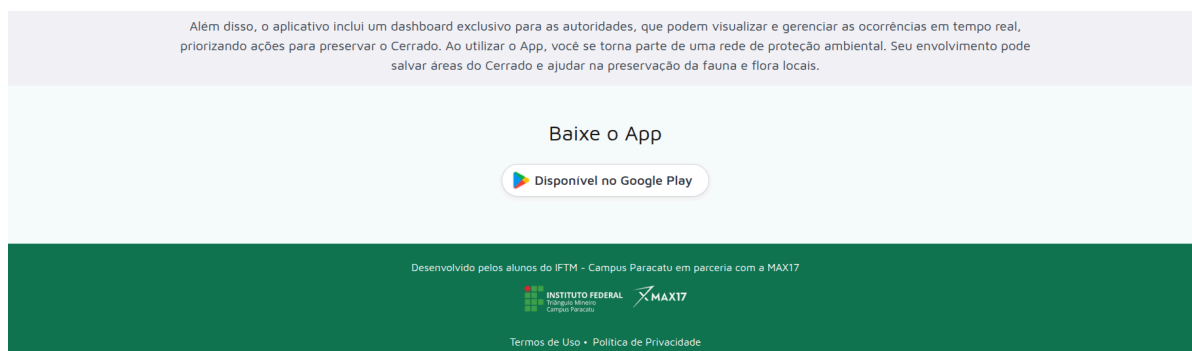
Figura 20 – Tela inicial - parte 3.



Fonte: Elaborado pelo autor, 2025



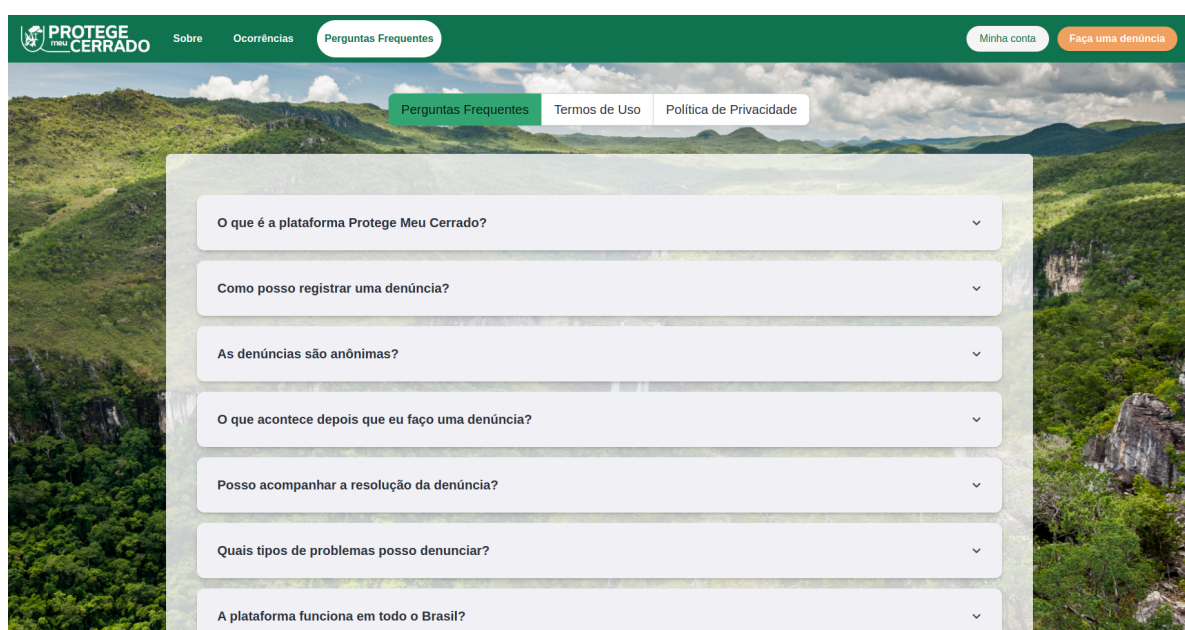
Figura 21 – Tela inicial - parte 4.



Fonte: Elaborado pelo autor, 2025

Outro elemento fundamental deste pacote foi a página de perguntas frequentes. Essa seção foi criada para fornecer respostas rápidas e objetivas para os questionamentos mais comuns. Foram incluídas perguntas sobre como fazer uma denúncia, como funcionam as denúncias anônimas, como as autoridades tratam as denúncias registradas, entre outras informações essenciais. Isso não apenas ajudou a reduzir a necessidade de suporte, como também tornou a navegação mais intuitiva, permitindo que os usuários encontrassem respostas por conta própria sem precisar recorrer a canais externos de atendimento. A Figura 22 apresenta a seção de perguntas frequentes.

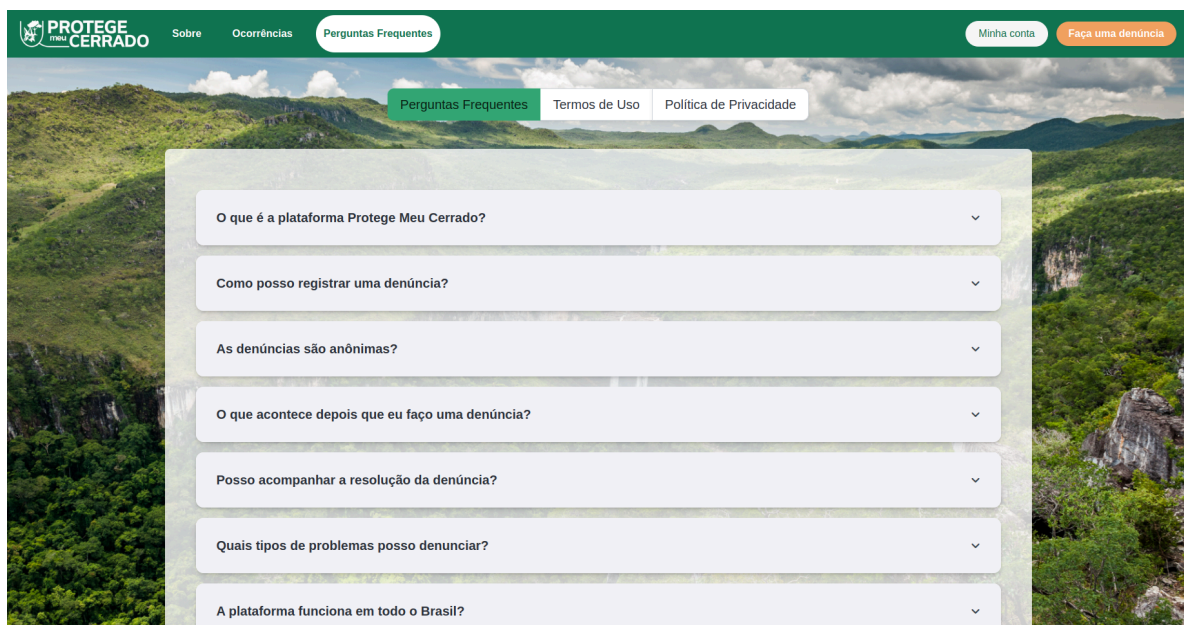
Figura 22 – Tela de perguntas frequentes.



Fonte: Elaborado pelo autor, 2025

Além das perguntas frequentes, também foram desenvolvidas as seções de termos de uso e política de privacidade na mesma página. Esses documentos são essenciais para garantir transparência e conformidade legal, estabelecendo as regras e diretrizes do uso da plataforma. Os termos de uso detalham as responsabilidades dos usuários, as diretrizes para o uso correto da plataforma e as penalidades para casos de uso indevido. Já a política de privacidade explica como os dados dos usuários são coletados, armazenados e protegidos, assegurando que todas as práticas estejam alinhadas com normas de proteção de dados, como a LGPD (Lei Geral de Proteção de Dados). Ambas as seções foram escritas de forma clara, evitando linguagem excessivamente técnica para que qualquer usuário pudesse compreender as informações sem dificuldades. A seção de termos de uso é demonstrada na Figura 23, enquanto a seção de política de privacidade é demonstrada na Figura 24.

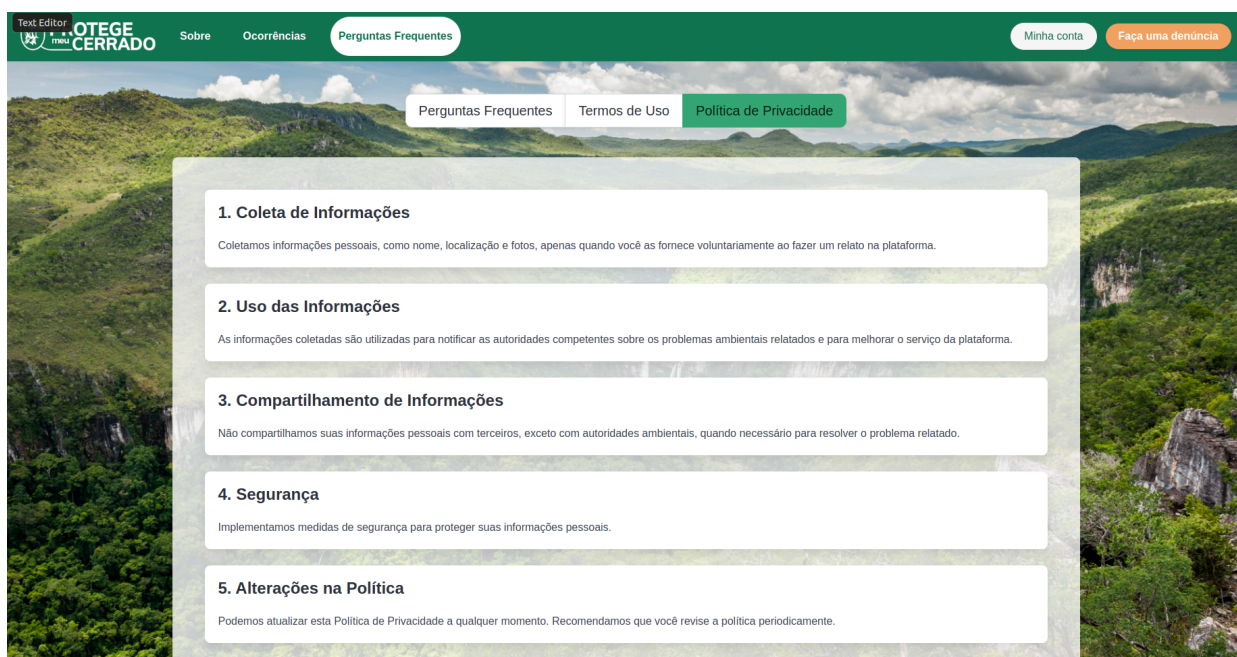
Figura 23 – Seção de termos de uso.



Fonte: Elaborado pelo autor, 2025



Figura 24 – Seção de política de privacidade.



Fonte: Elaborado pelo autor, 2025

A implementação dessas páginas foi fundamental para tornar a plataforma mais completa e confiável. A tela inicial serviu como uma vitrine para atrair novos usuários, destacando o propósito e os benefícios do Protege Meu Cerrado de forma envolvente. A inclusão da página de perguntas frequentes, termos de uso e política de privacidade tornou o suporte mais eficiente, fornecendo informações diretas e acessíveis.

Esse pacote representou um passo importante na consolidação do projeto, pois, além das funcionalidades técnicas já desenvolvidas, trouxe um refinamento essencial para a identidade e credibilidade da plataforma. Com essas adições, o Protege Meu Cerrado não apenas se tornou mais funcional, mas também mais transparente e profissional, oferecendo uma experiência mais completa para todos os usuários.

### 3.3.7 Pacote 7: Melhoria na estilização e identidade visual

O pacote 7 teve um papel essencial na experiência do usuário dentro da plataforma Protege Meu Cerrado, pois trouxe diversas melhorias voltadas para a usabilidade, navegação e refinamento visual. Até este momento do desenvolvimento, a plataforma já possuía diversas funcionalidades implementadas

e funcionando corretamente, mas ainda havia pontos que poderiam ser aprimorados para oferecer uma interação mais fluida e profissional. O objetivo principal deste pacote foi garantir que o usuário tivesse uma experiência mais intuitiva, sem frustrações com carregamentos demorados ou transições bruscas, além de tornar a interface mais agradável e coerente.

Uma das melhorias mais significativas foi a implementação de telas de carregamento. Durante o uso da plataforma, o usuário frequentemente precisa aguardar o processamento de informações, como navegar entre telas, fazer *login* ou cadastrar uma denúncia. Sem um *feedback* visual adequado, ele poderia ficar sem saber se a ação estava realmente sendo executada ou se o sistema estava travado. Para evitar essa sensação de incerteza e tornar a experiência mais previsível, foram adicionadas telas de carregamento em pontos estratégicos. Quando um usuário faz *login*, por exemplo, ao clicar no botão de entrada, uma tela de carregamento é exibida até que a requisição ao *backend* seja concluída e a autenticação confirmada. Isso evita que o usuário clique várias vezes no botão ou pense que o sistema não está respondendo. O mesmo conceito foi aplicado ao cadastro de denúncias, pois, como envolve múltiplas etapas e a necessidade de comunicação com o *backend*, foi importante garantir que o usuário soubesse que sua denúncia estava sendo processada corretamente. Além disso, ao navegar entre páginas, as telas de carregamento foram incluídas para garantir que os dados fossem completamente carregados antes de serem exibidos, evitando que o usuário visse telas vazias ou incompletas.

A implementação dessas telas de carregamento seguiu um *design* minimalista e eficiente, utilizando animações suaves, ícones de carregamento modernos e mensagens de *feedback* informativas, garantindo que a transição entre estados fosse agradável e sem impacto negativo na experiência do usuário. Esse cuidado foi essencial para consolidar a percepção de que a plataforma é confiável.

Outro ponto importante abordado neste pacote foi a melhoria na estilização e identidade visual da plataforma. Até então, a interface estava funcional, mas ainda precisava de ajustes para oferecer uma experiência mais agradável e coerente. Foram feitas diversas mudanças na paleta de cores, garantindo um visual mais equilibrado e harmonioso. A escolha das fontes foi revisada para melhorar a legibilidade e tornar a leitura dos textos mais fluida, além de ajustes nos tamanhos

de botões e espaçamentos para garantir uma organização visual mais agradável. Um dos aprimoramentos mais notáveis foi a adição de imagens de fundo, que trouxeram mais vida para a plataforma e ajudaram a criar uma identidade visual mais marcante, alinhada com o propósito ambiental do projeto.

Além das melhorias visuais e funcionais, essa etapa também foi importante para a correção de pequenos *bugs* e refinamentos no sistema. Durante os testes e uso contínuo da plataforma, foram identificadas algumas falhas que poderiam impactar a usabilidade. Alguns redirecionamentos entre páginas não estavam funcionando corretamente, fazendo com que certos fluxos de navegação se tornassem confusos, e isso foi ajustado para garantir uma experiência mais intuitiva. A validação de formulários também recebeu refinamentos, corrigindo pequenos erros que poderiam ocorrer ao inserir informações incorretas ou inválidas.

Esse pacote representou uma etapa essencial no refinamento final da plataforma, consolidando todo o trabalho realizado anteriormente e garantindo que os usuários tivessem uma experiência mais agradável, intuitiva e profissional. Com as telas de carregamento, os tempos de espera tornaram-se mais previsíveis e menos frustrantes. A estilização aprimorada trouxe um visual mais moderno e alinhado com o propósito da plataforma, tornando a interação mais agradável. E as correções de *bugs* garantiram que todas as funcionalidades estivessem funcionando da melhor forma possível. No fim, essas melhorias fizeram com que a Protege Meu Cerrado não fosse apenas uma plataforma funcional, mas também um sistema confiável e bem estruturado, pronto para ser utilizado de forma contínua.

### 3.4 DESAFIOS ENCONTRADOS

Durante o desenvolvimento do Protege Meu Cerrado, diversos desafios técnicos foram enfrentados, exigindo adaptações e melhorias para garantir a estabilidade e funcionalidade da plataforma. Um dos principais problemas ocorreu na comunicação entre componentes, pois inicialmente a passagem de variáveis era feita via *props*, o que gerava confusão no carregamento e tornava o código difícil de manter. Conforme o projeto crescia, essa abordagem se mostrou ineficiente, especialmente ao lidar com estados globais, como informações do

usuário, *status* de autenticação e dados compartilhados entre diferentes partes da aplicação. Para solucionar isso, foi implementado o *Context API*, permitindo o gerenciamento centralizado desses estados e garantindo que os dados estivessem disponíveis para qualquer parte da aplicação sem a necessidade de repassá-los manualmente. Essa solução não apenas melhorou a organização do código, mas também eliminou inconsistências que estavam surgindo devido à forma como os componentes eram atualizados.

A *navbar* também apresentou desafios, pois precisava ser atualizada dinamicamente para exibir diferentes opções dependendo do status de autenticação do usuário. No início, as mudanças nesse estado nem sempre eram refletidas corretamente na interface, gerando atrasos ou até falhas na atualização do menu. Com a implementação do *Context API*, a autenticação passou a ser gerenciada de maneira centralizada, garantindo que qualquer alteração no *status* do usuário fosse imediatamente refletida na *navbar*, sem necessidade de re-renderizações manuais. Esse mesmo problema de estado ocorreu na implementação do side-menu, que deveria aparecer e desaparecer ao clicar em um ícone. Inicialmente, a lógica para controlar sua visibilidade estava descentralizada e causando comportamentos inesperados. A solução veio com a criação do *MenuContext*, que permitiu um controle mais eficiente do estado do menu lateral, garantindo que sua exibição funcionasse corretamente em toda a aplicação.

Outro grande desafio foi a integração da *API do Google Maps* para permitir a geolocalização das denúncias. Durante a implementação, surgiram dificuldades na configuração da chave de *API*, permissões de uso e carregamento dos mapas dentro do *React*. A exibição dos marcadores e do próprio mapa nem sempre funcionava corretamente devido a problemas de reatividade dos componentes, o que gerava carregamentos desnecessários e impactava o desempenho da aplicação. Para resolver isso, além dos ajustes nas permissões do *Google Cloud* e da adoção da biblioteca *@react-google-maps/api*, foi implementado um *MapContext*.

Os desafios enfrentados trouxeram aprendizados importantes sobre gerenciamento de estado, integração de *APIs* externas e atualização dinâmica de componentes em *React*. A adoção do *Context API* simplificou a lógica do sistema e garantiu maior previsibilidade no comportamento da aplicação, enquanto as

melhorias na integração do *Google Maps* e no controle de componentes interativos, como a *navbar* e o menu lateral, proporcionaram uma experiência de usuário mais intuitiva e confiável. Essas soluções resultaram em um sistema mais estável, modular e escalável, pronto para futuras evoluções.

## 4 CONCLUSÃO

O desenvolvimento do Protege Meu Cerrado foi uma jornada enriquecedora que uniu inovação tecnológica e impacto ambiental. Ao longo do processo, foi possível aplicar práticas modernas de desenvolvimento e metodologias ágeis, como o uso de *Next.js*, *React*, *TypeScript* e *Scrum*, que contribuíram para a criação de uma plataforma eficiente, escalável e de alto desempenho. O projeto não só proporcionou um aprendizado técnico significativo, mas também destacou a importância de soluções tecnológicas na busca por um futuro mais sustentável.

A aplicação de boas práticas de modularização de componentes, uso da *Context API* e versionamento de código com *Git* e *GitHub* resultaram em um sistema mais organizado e de fácil manutenção, permitindo que a plataforma fosse desenvolvida de forma contínua e sem grandes obstáculos. A metodologia *Scrum*, por sua vez, foi essencial para garantir o cumprimento dos prazos e ajustes rápidos às necessidades emergentes do projeto.

A entrega oficial do Protege Meu Cerrado marcou um momento significativo no projeto, consolidando o esforço dedicado ao seu desenvolvimento. No dia 07 de fevereiro de 2025, a plataforma foi oficialmente inaugurada na Casa Paracatu, localizada em Paracatu-MG, em um evento especial que reuniu importantes autoridades ambientais, membros da comunidade e figuras influentes no cenário da preservação ecológica. Durante a cerimônia, foram apresentados os principais recursos e funcionalidades do sistema, ressaltando seu papel fundamental na denúncia e combate a crimes ambientais. A recepção positiva por parte dos presentes demonstrou a relevância da iniciativa e seu potencial de impacto na sociedade. A Figura 25 apresenta o evento de inauguração da plataforma na Casa Paracatu.

Figura 25 – Inauguração do Protege Meu Cerrado.



Fonte: Elaborado pelo autor, 2025

A experiência adquirida ao longo do desenvolvimento foi valiosa, consolidando conhecimentos em tecnologias modernas, metodologias ágeis e boas práticas de programação. Mais do que um projeto acadêmico, a plataforma representa uma solução real para um problema ambiental urgente, destacando a importância da inovação tecnológica na construção de um futuro mais sustentável.

Com a constante evolução do código e a implementação de novas funcionalidades, o Protege Meu Cerrado tem potencial para expandir seu impacto e alcançar um público cada vez maior.

## REFERÊNCIAS

ANDRADE, J. **Por que escolher Bulma como seu framework CSS?**. 2023.

Disponível em: <https://medium.com/@jandrade/bulma-framework-css>. Acesso em: fev. 2025.

APPMASTER. **As vantagens de utilizar TypeScript no desenvolvimento Web**.

2024. Disponível em:

<https://appmaster.io/pt/blog/vantagens-do-typescript-desenvolvimento-web>. Acesso em: fev. 2025.

BRASIL CODE. **10 Bibliotecas do React Que Você Precisa Conhecer**. 2024.

Disponível em:

<https://www.brasilcode.com.br/10-bibliotecas-do-react-que-voce-precisa-conhecer/>. Acesso em: fev. 2025.

BULMA. **Official Documentation**. 2024. Disponível em:

<https://bulma.io/documentation/>. Acesso em: fev. 2025.

CAPUTO, M. **Bulma: Um framework CSS moderno e responsivo**. 2023.

Disponível em: <https://www.devto.com/bulma-framework>. Acesso em: fev. 2025.

CHACON, S.; STRAUB, B. **Pro Git**. 2021. Disponível em:

<https://git-scm.com/book/en/v2>. Acesso em: fev. 2025.

DATE-FNS. **Documentação do date-fns**. 2024. Disponível em: <https://date-fns.org>.

Acesso em: fev. 2025.

DEVELOPER, D. **Introdução ao @react-google-maps/api**. 2023. Disponível em:

<https://developer.google.com/maps/documentation/react>. Acesso em: fev. 2025.

DIO. **Motivos para Utilizar o TypeScript no Desenvolvimento Backend**. 2024.

Disponível em:

<https://www.dio.me/articles/motivos-para-utilizar-o-typescript-no-desenvolvimento-backend>. Acesso em: fev. 2025.

DIO. **Vantagens de Utilizar TypeScript no Desenvolvimento de Software Front-End**. 2024. Disponível em:

<https://www.dio.me/articles/vantagens-de-utilizar-typescript-no-desenvolvimento-de-software-front-end>. Acesso em: fev. 2025.

GEEKSFORGEEKS. **ReactJS Virtual DOM**. 2024. Disponível em:

<https://www.geeksforgeeks.org/reactjs-virtual-dom/>. Acesso em: fev. 2025.



GITHUB. **GitHub Documentation**. 2024. Disponível em: <https://docs.github.com>. Acesso em: fev. 2025.

GOOGLE MAPS. **Google Maps API Documentation**. 2023. Disponível em: <https://maps.google.com/maps/api>. Acesso em: fev. 2025.

HYBRID HEROES. **Next.js Rendering Strategies: SSR, SSG, and ISR Compared**. 2023. Disponível em: <https://hybridheroes.de/blog/2023-05-31-next-js-rendering-strategies/>. Acesso em: fev. 2025.

KARIMOV, F. **Understanding the Differences Between SSR, SSG, and ISR in Next.js**. 2023. Disponível em: <https://javascript.plainenglish.io/understanding-differences-between-ssr-ssg-and-isr-in-next-js-8b8dd240264d>. Acesso em: fev. 2025.

LOGROCKET. **Using Next.js with TypeScript**. 2024. Disponível em: <https://blog.logrocket.com/using-next-js-with-typescript/>. Acesso em: fev. 2025.

NEXT.JS DOCUMENTATION. **Incremental Static Regeneration (ISR)**. [s.d.]. Disponível em: <https://nextjs.org/docs/pages/building-your-application/data-fetching/incremental-static-regeneration>. Acesso em: fev. 2025.

NEXT.JS DOCUMENTATION. **Server-side Rendering (SSR)**. [s.d.]. Disponível em: <https://nextjs.org/docs/pages/building-your-application/rendering/server-side-rendering>. Acesso em: fev. 2025.

PRADANA, L. E. **Optimizing Next.js Performance: Best Practices for SSG, SSR, ISR, and CSR**. 2023. Disponível em: <https://medium.com/@lelianto.eko/optimizing-next-js-performance-best-practices-for-ssg-ssr-isr-and-csr-02aba860429b>. Acesso em: fev. 2025.

REACT DOCUMENTATION. **React – Uma biblioteca JavaScript para criar interfaces de usuário**. 2024. Disponível em: <https://react.dev/>. Acesso em: fev. 2025.

REACT GOOGLE MAPS. **React Google Maps: A new generation of mapping**. 2024. Disponível em: <https://react-google-maps.com>. Acesso em: fev. 2025.

REACT INPUT MASK. **Documentação do React Input Mask**. 2023. Disponível em: <https://react-input-mask.github.io/react-input-mask/>. Acesso em: fev. 2025.

REVELO COMMUNITY. **ReactJS: Uma Jornada pelo Mundo das Interfaces Interativas**. 2023. Disponível em: <https://community.revelo.com.br/reactjs-uma-jornada-pelo-mundo-das-interfaces-interativas/>. Acesso em: fev. 2025.

SACHS, J. D. et al. **Technological innovations for sustainable development and environmental protection**. Springer Nature, 2020.

SWIPER. **Swiper Documentation**. 2023. Disponível em: <https://swiperjs.com>. Acesso em: fev. 2025.