**ESCOLA E FACULDADE DE TECNOLOGIA SENAI ROBERTO MANGE**

**DESENVOLVIMENTO DE SISTEMAS**

**GABRIEL BOSCO DEOLINDO**

**SMART CITY**

CAMPINAS

2025

**GABRIEL BOSCO DEOLINDO**

**SMARTCITY**

Projeto Integrador apresentado a Escola e Faculdade de Tecnologia Senai Roberto Mange - Campinas, como parte dos requisitos necessários à obtenção do título de Desenvolvedor de Sistemas.

CAMPINAS

2025

**SUMÁRIO**

[**1. INTRODUÇÃO** 4](#_Toc201169425)

[**1.1 OBJETIVO** 4](#_Toc201169426)

[**1.2 ESCOPO** 4](#_Toc201169427)

[**2. VISÃO GERAL DO SOFTWARE** 5](#_Toc201169428)

[**2.1 PERSPECTIVA DO PRODUTO** 5](#_Toc201169429)

[**2.2 FUNÇÕES DO PRODUTO** 5](#_Toc201169430)

[**2.3 RESTRIÇÕES** 5](#_Toc201169431)

[**3. DESCRIÇÃO DOS REQUISITOS** 6](#_Toc201169432)

[**3.1 REQUISITOS FUNCIONAIS** 6](#_Toc201169433)

[**3.2 REQUISITOS NÃO FUNCIONAIS** 7](#_Toc201169434)

[**4. PROTÓTIPO DAS TELAS** 8](#_Toc201169435)

[**5. IDENTIDADE VISUAL** 11](#_Toc201169436)

[**5.1 PALETA DE CORES** 11](#_Toc201169437)

[**5.2 ÍCONES USADOS** 11](#_Toc201169438)

[**6. DESENVOLVIMENTO DO SISTEMA** 12](#_Toc201169439)

[**6.1 DIAGRAMA DE CASOS DE USO** 12](#_Toc201169440)

[**6.2 LINKS** 12](#_Toc201169441)

[**7. CONCLUSÃO** 13](#_Toc201169442)

# **1. INTRODUÇÃO**

## **1.1 OBJETIVO**

Este Documento de Especificação de Requisitos de Software (SRS) descreve detalhadamente o funcionamento do software de cidade inteligente, servindo como guia para o desenvolvimento, validação e manutenção do sistema. Ele será utilizado por desenvolvedores, gestores, stakeholders e testadores, sendo o ponto central de referência para garantir que a solução proposta atenda às necessidades identificadas. Este documento traduz a visão do projeto em especificações técnicas claras e verificáveis.

## **1.2 ESCOPO**

O software de cidade inteligente é uma aplicação web que permite o monitoramento de sensores ambientais distribuídos em uma cidade. A plataforma proporciona aos usuários a capacidade de:

* Realizar cadastro e login;
* Visualizar sensores e os ambientes nos quais estão inseridos;
* Consultar o histórico de dados registrados;
* Acompanhar informações através de gráficos e mapas.

A aplicação busca facilitar a tomada de decisões baseadas em dados, contribuindo para uma gestão mais eficiente, transparente e sustentável. O sistema conta com backend desenvolvido em Python (Django REST Framework com autenticação JWT) e frontend construído com React e TailwindCSS. Ferramentas como Pandas são utilizadas para importação dos dados de Excel.

# **2. VISÃO GERAL DO SOFTWARE**

## **2.1 PERSPECTIVA DO PRODUTO**

A solução proposta é uma aplicação web modular e escalável voltada para a coleta, visualização e análise de dados de sensores distribuídos pela escola. A aplicação oferece uma interface amigável com foco na usabilidade e performance.

## **2.2 FUNÇÕES DO PRODUTO**

* Cadastro e autenticação de usuários com segurança baseada em JWT;
* Visualização de sensores;
* Acompanhamento de ambientes, onde os sensores estão instalados;
* Histórico de leituras com filtros por id;
* Análise visual por meio de gráficos interativos e mapas geográficos;
* Interface administrativa para gerenciar sensores e ambientes.

## **2.3 RESTRIÇÕES**

* A aplicação deve ser compatível com os navegadores mais modernos (Chrome, Firefox, Edge);
* O sistema será acessível apenas via web;
* O backend será desenvolvido com Python utilizando o Django REST Framework e JWT;
* O frontend será desenvolvido com React e TailwindCSS;
* Bibliotecas como Pandas serão utilizadas para análise de dados;

# **3. DESCRIÇÃO DOS REQUISITOS**

## **3.1 REQUISITOS FUNCIONAIS**

**RF01 – Página de Login:**

* Permitir que o usuário insira seu nome e número de matricula para acessar a plataforma.
* Validar autenticação por JWT.

**RF02 – Página de Cadastro:**

* Permitir que novos usuários se registrem com nome e senha (esses usuários não tem acesso a visualização nem edição dos dados, apenas o super usuário).

**RF03 – Visualização de Sensores:**

* Listar todos os sensores registrados na base de dados, exibindo nome, tipo e localização.

**RF04 – Visualização de Ambientes:**

* Apresentar os ambientes cadastrados.

**RF05 – Histórico de Dados:**

* Permitir ao usuário visualizar o histórico de medições dos sensores.

**RF06 – Gráficos e Mapas:**

* Exibir gráficos interativos com leitura dos dados dos sensores.
* Mostrar mapas geográficos com a localização dos sensores ativos.

**RF07 – Dashboard:**

* Exibir resumos e estatísticas gerais sobre os dados coletados.

## **3.2 REQUISITOS NÃO FUNCIONAIS**

**RNF01 – Tempo de Carregamento:**

* Todas as páginas devem carregar em até 2 segundos em rede padrão.

**RNF02 – Segurança:**

* Dados do usuário e dos sensores devem ser armazenados de forma segura, com criptografia em repouso e em trânsito.

**RNF03 – Disponibilidade:**

* A aplicação deve estar disponível 99,9% do tempo.

**RNF04 – Compatibilidade:**

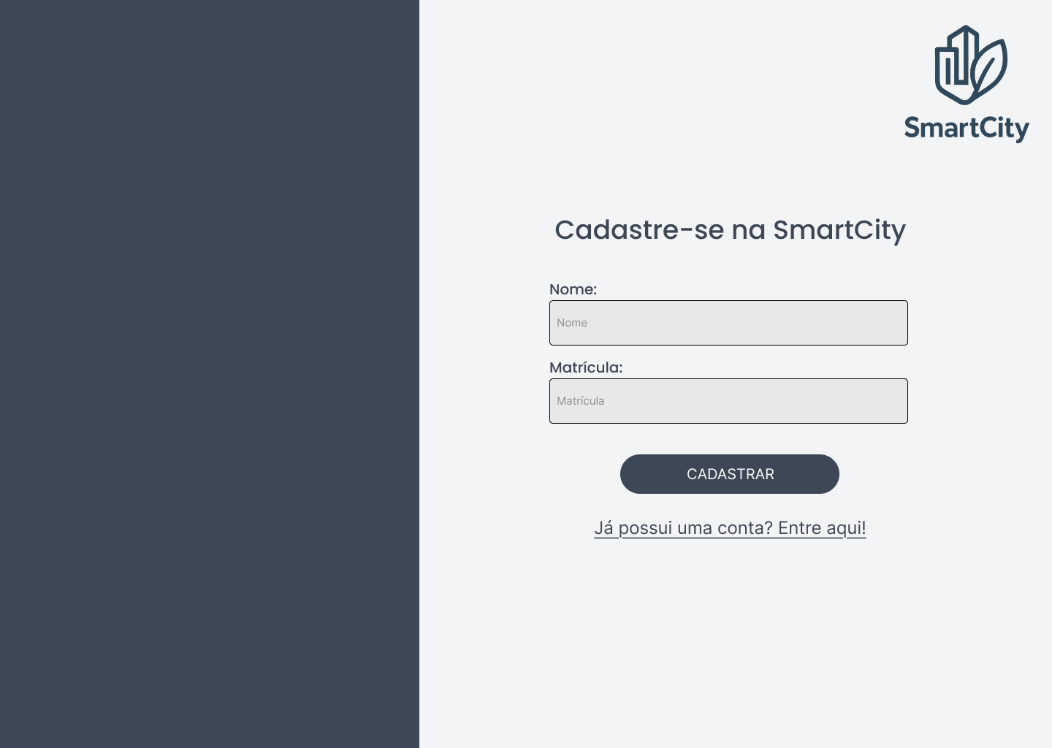
* Compatível com as duas últimas versões dos navegadores mais populares.

**RNF05 – Usabilidade:**

* Interface intuitiva e acessível, com curva de aprendizado inferior a 15 minutos.

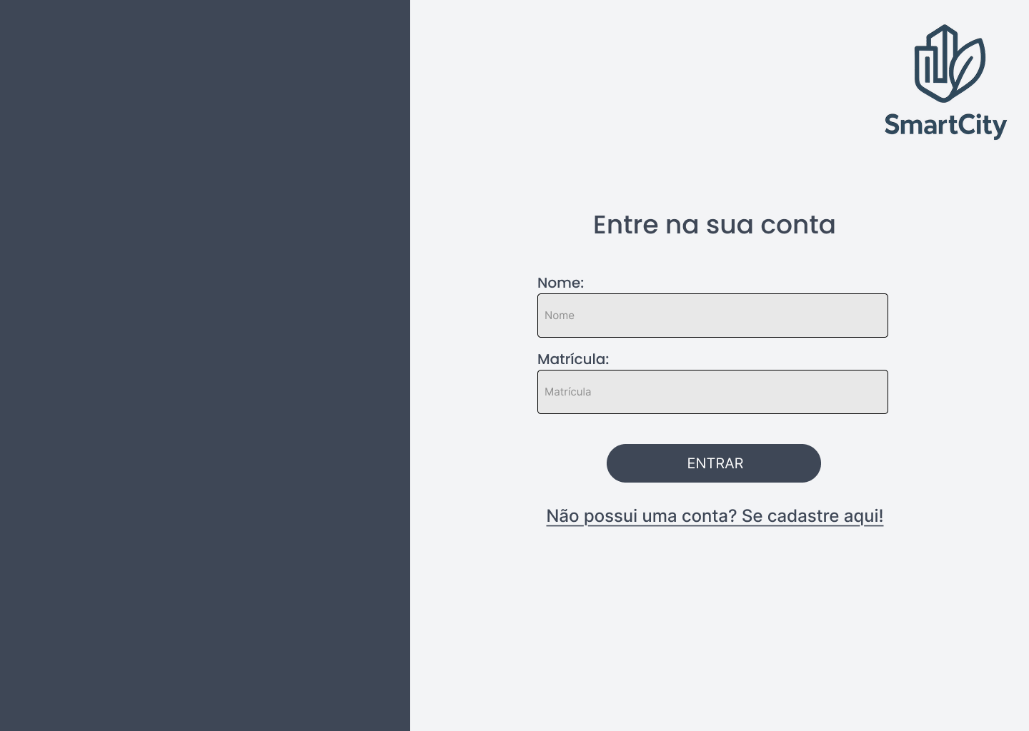
# **4. PROTÓTIPO DAS TELAS**

Figura 1 – Tela de Cadastro



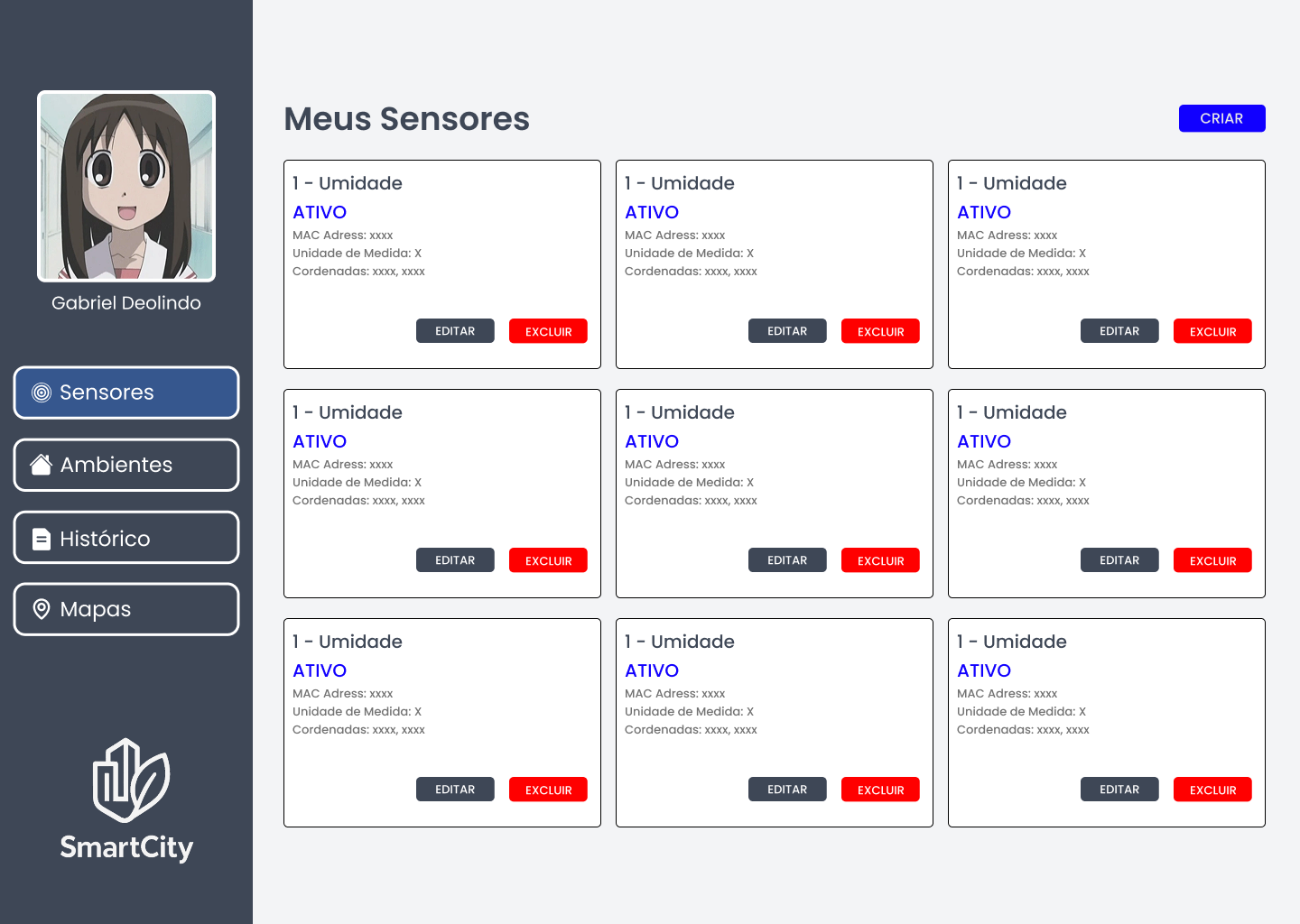
Fonte: Elaborado pelo autor

Figura 2 – Tela De Login



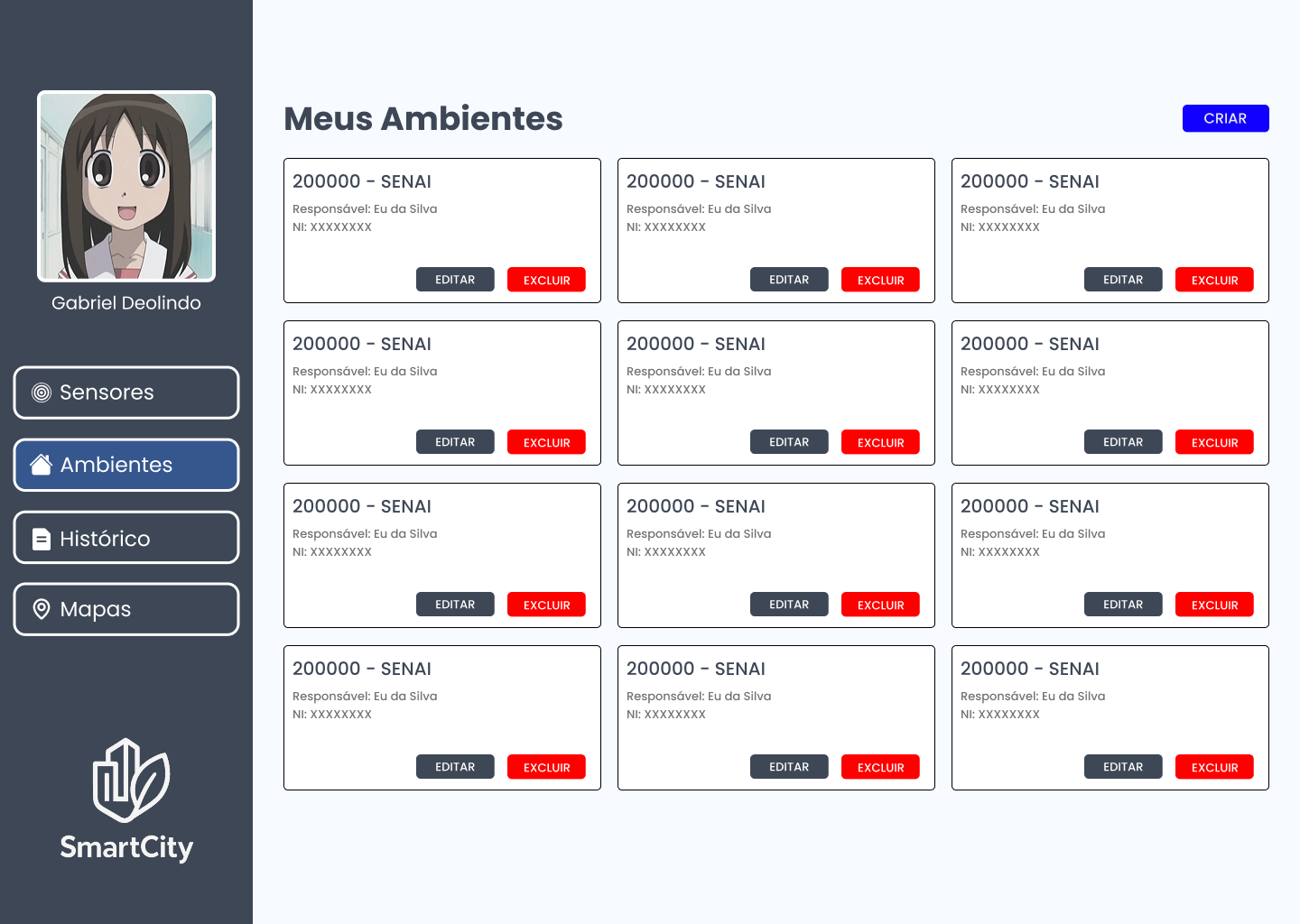
Fonte: Elaborado pelo autor

Figura 3 – Tela de Sensores



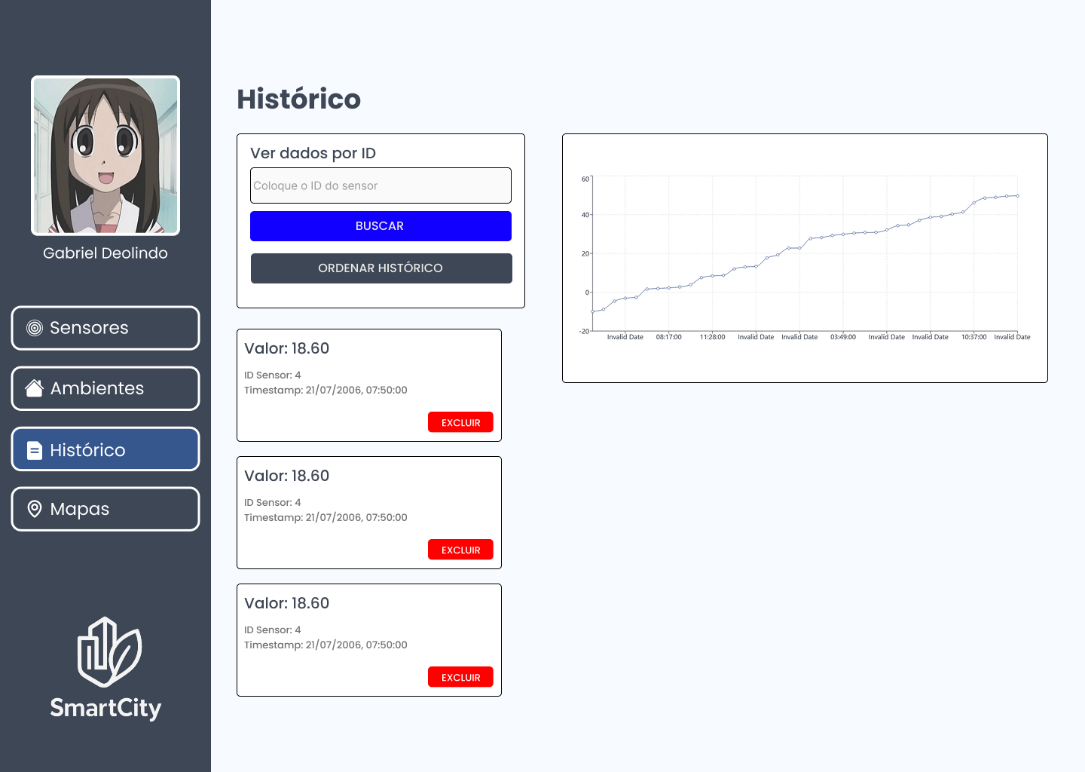
Fonte: Elaborado pelo autor

Figura 4 – Tela de Ambientes



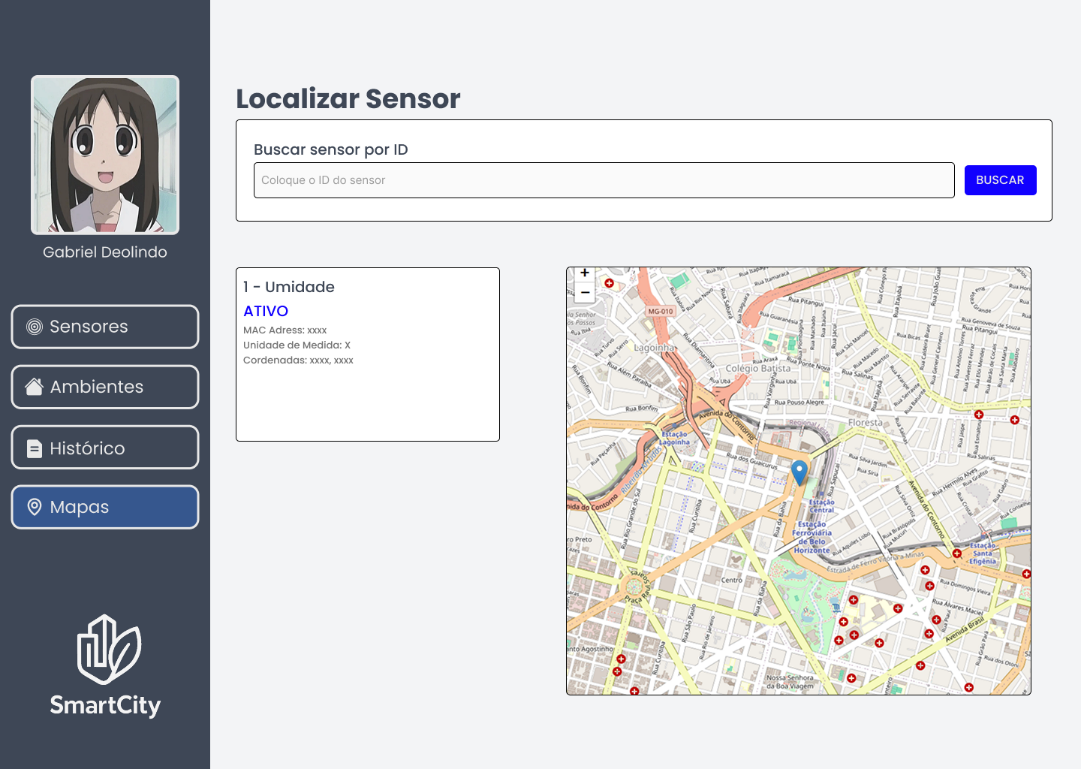
Fonte: Elaborado pelo autor

Figura 5 – Tela de Histórico



Fonte: Elaborado pelo autor

Figura 6 – Tela de Mapa



Fonte: Elaborado pelo autor

# **5. IDENTIDADE VISUAL**

## **5.1 PALETA DE CORES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **#3E4756** | **#F3F4F6** | **#FFFFFF** | **#000000** | #ee2525 | #155dfc |
| **Cor principal do sistema, usado em backgrounds e alguns botões** | **Cor secundária do sistema, usado em backgrounds** | **Cor usada no background dos sensores para melhor visibilidade** | **Textos/Bordas** | **Botões de exclusão, Sensores Inativos** | **Botões de confirmação, Sensores Ativos** |

## **5.2 ÍCONES USADOS**

Figura 7 – Logo SmartCity

Logotipo

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelo autor

Ícone

O conteúdo gerado por IA pode estar incorreto.Ícone

O conteúdo gerado por IA pode estar incorreto.Ícone

O conteúdo gerado por IA pode estar incorreto.Ícone

O conteúdo gerado por IA pode estar incorreto.Ícone

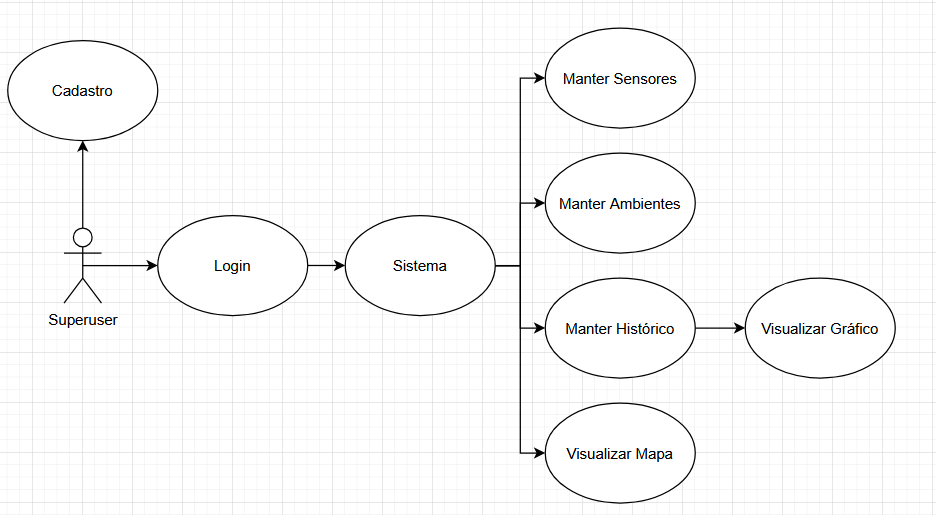
O conteúdo gerado por IA pode estar incorreto.

Fonte: React Icons

# **6. DESENVOLVIMENTO DO SISTEMA**

## **6.1 DIAGRAMA DE CASOS DE USO**

Figura 8 - Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

## **6.2 LINKS**

**Figma:** <https://www.figma.com/design/WCJHlqZHViqJPIkrGO23t5/Integras?t=BvNxPUAKsJgensYn-1>

**Github:**Professor Dorival pediu para o repositório ficar privado, então adicionei a professora como colaboradora

# **7. CONCLUSÃO**

O desenvolvimento deste sistema de cidade inteligente representa um avanço significativo na integração entre tecnologia e gestão de sensores. Por meio de uma arquitetura moderna composta por Django REST Framework no backend e React com Tailwind CSS no frontend, o projeto oferece uma solução robusta, segura e acessível para o monitoramento e análise de ambientes urbanos.

Durante o desenvolvimento, foram aplicados conceitos importantes de segurança, responsividade e usabilidade. O projeto também foi pensado para ser escalável, com possibilidade de integração com novos sensores e módulos no futuro.