**INSTITUTO FEDERAL DE RONDÔNIA**
***CAMPUS* PORTO VELHO CALAMA**
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**IARA RODRIGUES ABREU**

# PROTÓTIPO DE SISTEMA DE GERENCIAMENTO DE DEMANDAS URBANAS: ANÁLISE E APLICAÇÃO PARA A CIDADE DE PORTO VELHO

**Porto Velho/RO**
**2025**

**IARA RODRIGUES ABREU**

# PROTÓTIPO DE SISTEMA DE GERENCIAMENTO DE DEMANDAS URBANAS: ANÁLISE E APLICAÇÃO PARA A CIDADE DE PORTO VELHO

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus* Porto Velho Calama, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Leandro Ferrarezi Valiante

**Porto Velho/RO**
**2025**

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

A162p

Abreu, Iara Rodrigues.

Protótipo de sistema de gerenciamento de demandas urbanas: análise e aplicação para a cidade de Porto Velho / Iara Rodrigues Abreu. - Porto Velho, 2025.

19 f. : il.

Orientador(a): Me Leandro Ferrarezi Valiante.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho, 2025.

1. Análise de Sistemas. 2. Gerenciamento de dados. 3. Gestão pública. I. Valiante, Leandro Ferrarezi (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

CDD: 005.74

**Bibliotecário(a) Responsável:** Evandro Silva de Sousa, CRB-11-956

**IARA RODRIGUES ABREU**

**PROTÓTIPO DE SISTEMA DE GERENCIAMENTO DE DEMANDAS URBANAS: ANÁLISE E APLICAÇÃO PARA A CIDADE DE PORTO VELHO**

A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso “Protótipo de Sistema de Gerenciamento de Demandas Urbanas: Análise e Aplicação para a Cidade de Porto Velho” elaborado por “Iara Rodrigues Abreu” como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia.

Porto Velho/RO, 22/07/2025

**Comissão Examinadora**

gov.br
Documento assinado digitalmente
LEANDRO FERRAREZI VALIANTE
Data: 30/07/2025 19:19:14-0300
Verifique em https://validar.iti.gov.br

**Prof. Leandro Ferrarezi Valiante** - IFRO
(Orientador)

gov.br
Documento assinado digitalmente
CELSO GUEDES GOMES
Data: 30/07/2025 19:15:07-0300
Verifique em https://validar.iti.gov.br

**Prof. Celso Guedes Gomes** - IFRO

gov.br
Documento assinado digitalmente
PAULO SERGIO TOME
Data: 30/07/2025 18:50:54-0300
Verifique em https://validar.iti.gov.br

**Prof. Paulo Sérgio Tomé** - IFRO

# PROTÓTIPO DE SISTEMA DE GERENCIAMENTO DE DEMANDAS URBANAS: ANÁLISE E APLICAÇÃO PARA A CIDADE DE PORTO VELHO

**RESUMO:** O presente trabalho tem como objetivo analisar e propor um protótipo de sistema de gerenciamento de demandas urbanas voltado para a cidade de Porto Velho, Rondônia. A pesquisa é de natureza qualitativa, com caráter exploratório e descritivo, e concentra-se na avaliação de requisitos de usabilidade aplicados à interação entre cidadãos e órgãos públicos. Foram analisadas plataformas existentes, como COLAB, 156 (São José dos Campos) e NYC 311 (Nova Iorque), a fim de identificar boas práticas e limitações. Com base nesses estudos, foram levantados os requisitos do sistema, modelados por meio de diagramas UML (casos de uso e classes), além do desenvolvimento de um protótipo de interface utilizando a ferramenta Figma. O sistema proposto visa centralizar o registro, acompanhamento e resposta às demandas urbanas, promovendo maior eficiência, transparência e acessibilidade com o desenvolvimento de soluções digitais voltadas à gestão pública municipal.

**PALAVRAS-CHAVE**: Demandas urbanas. Análise de sistemas. Usabilidade. UML. Protótipo. Gestão pública.

**ABSTRACT:** This study aims to analyze and propose a prototype of an urban demand management system for the city of Porto Velho, Rondônia. The research follows a qualitative approach, with an exploratory and descriptive character, focusing on the evaluation of usability requirements applied to the interaction between citizens and public agencies. Existing platforms, such as COLAB, 156 (São José dos Campos), and NYC 311 (New York City), were analyzed to identify best practices and limitations. Based on these studies, system requirements were defined and modeled using UML diagrams (use case and class diagrams), in addition to the development of an interface prototype using the Figma tool. The proposed system aims to centralize the registration, monitoring, and response to urban demands, promoting greater efficiency, transparency, and accessibility through the development of digital solutions for municipal public management.

**KEYWORDS**: Urban demands. Systems analysis. Usability. UML. Prototype. Public management.

# 1 INTRODUÇÃO

De acordo com Fernandes (2011), a falta de planejamento e gerenciamento urbano é um dos principais fatores que contribuem para a precariedade da infraestrutura dos municípios brasileiros. Essa deficiência impacta diretamente a qualidade de vida dos cidadãos, que, privados de uma infraestrutura adequada, também acabam sendo privados do pleno usufruto dos bens públicos pelos quais pagam impostos.

Sistemas de gerenciamento de demandas urbanas têm sido adotados em cidades ao redor do mundo com o objetivo de facilitar a comunicação entre cidadãos e órgãos públicos. Essas plataformas permitem o registro e o acompanhamento de problemas de infraestrutura e manutenção urbana, oferecendo um canal mais eficiente para a resolução de demandas. Entre as soluções mais conhecidas está a plataforma COLAB, que se destaca no Brasil como uma ferramenta que centraliza o envio de demandas urbanas, permitindo que os cidadãos se comuniquem diretamente com as prefeituras para relatar problemas como buracos nas ruas, falhas na iluminação pública e deficiências na coleta de lixo. O COLAB também oferece funcionalidades que permitem aos usuários acompanhar o status de suas solicitações e interagir com outras demandas em sua região, promovendo maior transparência e engajamento cívico.

Além do COLAB, várias cidades, tanto no Brasil quanto no exterior, têm desenvolvido sistemas próprios para gerenciar demandas urbanas. Por exemplo, a cidade de São José dos Campos conta com o 156 um aplicativo onde permite aos cidadãos registrar pedidos de serviços públicos. Já a cidade de Nova Iorque possui o sistema NYC 311, que centraliza as demandas de manutenção e problemas públicos em uma única plataforma acessível via telefone, aplicativo móvel e website. Esses sistemas oferecem vantagens importantes, como a centralização das demandas, o que facilita tanto para o cidadão quanto para o governo, além da possibilidade de acompanhar o progresso das solicitações, promovendo uma relação de maior transparência entre a administração pública e a população. No entanto, um dos desafios comuns a essas plataformas é garantir a integração entre diferentes órgãos públicos, principalmente quando há várias entidades envolvidas em diferentes áreas de serviços. Essa falta de integração pode gerar atrasos ou duplicidade no processamento de demandas, comprometendo a eficiência do sistema.

Nos últimos anos, Porto Velho, capital do estado de Rondônia, testemunhou um rápido crescimento populacional. Segundo o Instituto Brasileiro de Geografia e Estatística (2022), o crescimento populacional da cidade aumentou 7,4% em comparação ao ano de $2010$, alcançando um total de aproximadamente $460.434$ habitantes em $2022$. Esse crescimento exponencial trouxe consigo novos desafios, incluindo a necessidade de modernizar a infraestrutura e a manutenção urbana. A cidade, outrora conhecida

por seu charme rural, agora se destaca como um polo econômico e cultural em crescimento. A eficiência na manutenção de ativos públicos, como postes de iluminação, ruas e parques, tornou-se fundamental para garantir a qualidade de vida da população, a segurança da infraestrutura urbana e a atratividade da cidade.

Atualmente, a gestão de demandas de manutenção urbana em Porto Velho é realizada de maneira descentralizada, sendo distribuída entre diferentes órgãos, cada um responsável por demandas específicas. Entre os principais órgãos envolvidos estão a Secretaria Municipal de Saneamento e Serviços Básicos (SEMUSB), a Empresa de Desenvolvimento Urbano (EMDUR), a Companhia de Água e Esgoto (CAERD) e a Energisa, cada um com a incumbência de atender demandas específicas, como iluminação pública, saneamento e fornecimento de água e energia. Cada um desses órgãos dispõe de canais próprios para receber as solicitações dos cidadãos. Por exemplo, a EMDUR oferece, em seu site oficial, uma funcionalidade dedicada ao registro de demandas relacionadas à manutenção da iluminação pública. Da mesma forma, a SEMUSB disponibiliza, em seu site, uma ferramenta que permite aos cidadãos reportar problemas como limpeza urbana e coleta de resíduos, além de oferecer contatos telefônicos e por e-mail. A CAERD, por sua vez, opera com um sistema de atendimento virtual via *chatbot*, que visa automatizar o processo de triagem das demandas e encaminhá-las ao setor responsável.

No entanto, essa abordagem fragmentada apresenta uma série de desafios. Primeiramente, a descentralização dos canais de atendimento dificulta o processo para os cidadãos, que frequentemente se deparam com dúvidas sobre a qual órgão devem direcionar suas demandas. A ausência de um canal centralizado compromete a transparência e a visibilidade do processo, uma vez que os cidadãos não têm uma visão clara sobre o status de suas solicitações ou das demandas gerais da cidade. Sem um sistema integrado e eficiente para gerenciar essas demandas, a qualidade dos serviços prestados e a satisfação dos cidadãos podem ser significativamente comprometidas.

O objetivo geral deste trabalho é analisar, do ponto de vista da usabilidade, como um sistema de gerenciamento de demandas de manutenção urbana pode atender aos requisitos de interação usuário-sistema de forma eficiente e eficaz, garantindo uma melhor experiência para seus usuários e facilitando a comunicação com os órgãos responsáveis. Como objetivos específicos, buscam-se: (i) estudar sistemas existentes de gerenciamento de demandas urbanas; (ii) levantar os requisitos essenciais para a usabilidade do sistema; (iii) descrever as funcionalidades e especificações técnicas do sistema; e (iv) identificar de que maneira os requisitos de usabilidade contribuem para a eficiência e eficácia na gestão das demandas urbanas, propondo um plano de implementação baseado nas necessidades levantadas.

# 2 FUNDAMENTAÇÃO TEÓRICA

Produtos de software estão presentes em todos os momentos de nossas vidas. Diariamente, mesmo sem nos darmos conta, entramos em contato com diferentes tipos de software (Reinehr, 2020). Segundo Pressman e Maxim (2021), o software é o elemento-chave na evolução de produtos e sistemas baseados em computador e uma das mais importantes tecnologias no cenário mundial. A Engenharia de Software, nesse sentido, utiliza processos, métodos e ferramentas que facilitam o desenvolvimento de sistemas complexos com qualidade e no prazo previsto. Essa disciplina abrange cinco atividades estruturais principais: comunicação, planejamento, modelagem, construção e entrega, aplicáveis a todos os projetos de software, o que permite uma abordagem organizada e sistemática no desenvolvimento de soluções eficazes (Pressman; Maxim, 2021).

O desenvolvimento de projetos de software apresenta complexidades desde as fases iniciais, especialmente no que diz respeito à definição de requisitos. Segundo Reinehr (2020), um requisito é caracterizado como uma condição ou capacidade necessária para que um usuário possa resolver um problema ou atingir um objetivo. Além disso, pode ser descrito como uma condição que um sistema deve atender para cumprir um contrato ou especificação. A mesma autora destaca ainda que o ciclo de vida de um produto de software abrange diversas etapas pelas quais o produto passa ao longo de sua existência.

O ciclo de vida de um produto de software é tradicionalmente dividido em quatro etapas principais: concepção, desenvolvimento, manutenção e descontinuação. Na fase de concepção, identifica-se a necessidade de um novo software e analisa-se a viabilidade de sua construção. Durante essa etapa, os requisitos iniciais são estabelecidos como objetivos de negócio de alto nível, que posteriormente são detalhados em requisitos funcionais. Além disso, é comum que surjam requisitos de projeto e de processo, os quais podem restringir o escopo e influenciar a forma como o projeto será desenvolvido (Pressman; Maxim, 2021; Sommerville, 2019).

A etapa de desenvolvimento engloba todas as atividades necessárias para a produção do software, incluindo a engenharia de requisitos, análise, design, implementação, testes e implantação. Após a definição inicial do escopo, as atividades de engenharia de requisitos são intensificadas, levando ao design da solução e à definição da arquitetura do sistema (Pfleeger; Atlee, 2010; Wiegers; Beatty, 2013). Essas atividades são fundamentais para garantir que o software atenda às necessidades dos *stakeholders* e seja construído de forma eficiente e sustentável.

A fase de manutenção representa a maior parte do tempo de vida de um software. Assim que a primeira funcionalidade é implementada, começam as solicitações

de mudanças, que podem incluir correções de bugs — falhas ou defeitos inesperados que afetam o funcionamento do software —, demandas legais ou melhorias. A rastreabilidade dos requisitos é essencial para analisar o impacto dessas solicitações. Por fim, a descontinuação ocorre quando um software não atende mais às suas finalidades. Essas fases são fundamentais para garantir que o software atenda às necessidades dos usuários ao longo de sua vida útil (Reinehr, 2020).

Um processo de software define a abordagem adotada conforme um software é elaborado pela Engenharia de Software. Entretanto, a Engenharia de Software também engloba tecnologias que fazem parte do processo — métodos técnicos e ferramentas automatizadas (Pressman; Maxim, 2021). Nesse contexto, a UML (*Unified Modeling Language*, ou Linguagem de Modelagem Unificada) desempenha um papel essencial, pois fornece uma notação robusta para a modelagem de sistemas orientados a objetos e estabelece um padrão tanto para a modelagem de requisitos quanto para a de projetos. Segundo Fowler (2005), a UML é uma família de notações gráficas, sustentada por um metamodelo único, que facilita a descrição e o design de sistemas de software, especialmente aqueles desenvolvidos com o estilo orientado a objetos.

Segundo Fowler (2005), a UML pode ser utilizada de três maneiras principais: como esboço, projeto e linguagem de programação. O uso mais comum da UML é como esboço, onde os desenvolvedores a utilizam para comunicar aspectos de um sistema, seja no desenvolvimento inicial ou na engenharia reversa. Os esboços são informais e dinâmicos, focando na comunicação de ideias e problemas específicos, muitas vezes facilitados por ferramentas simples, como quadros brancos. Por outro lado, a UML como projeto se concentra na completude, criando um modelo detalhado que pode ser seguido por programadores.

Segundo Pressman (2016), os diagramas de casos de uso têm como objetivo descrever as interações entre os usuários e o sistema, identificando as principais funcionalidades que este deve oferecer. O diagrama de classes define a estrutura do sistema, representando suas principais classes, atributos, métodos e os relacionamentos entre elas. Essa modelagem é essencial para a implementação do sistema, servindo como um guia para os desenvolvedores. Já o diagrama de sequência representa o fluxo de interações entre os usuários e o sistema ao longo do tempo, detalhando como as funcionalidades são executadas. Ele é composto por objetos, como os usuários, o sistema e o banco de dados, além das mensagens trocadas entre eles para garantir o funcionamento adequado do processo. Esse método requer ferramentas mais sofisticadas para gerenciar os detalhes e é comum em contextos de engenharia onde desenhos técnicos são necessários. A diferença entre esboços e projetos reside no fato de que os esboços são deliberadamente incompletos e exploratórios, enquanto os projetos são definitivos e visam simplificar a programação.

De acordo com Sommerville (2019), a UML pode ser utilizada não apenas como ferramenta de modelagem, mas também como uma linguagem de programação em contextos de engenharia dirigida por modelos (*Model-Driven Engineering*). Nessa abordagem, os diagramas são elaborados de forma tão completa e detalhada que podem ser compilados diretamente em código executável por ferramentas específicas, tornando o modelo e o código-fonte praticamente indistinguíveis e dispensando a necessidade de engenharia reversa. Essa aplicação, conforme destaca o autor, exige ferramentas sofisticadas para gerenciar os detalhes e assegurar a integridade do sistema. Assim, a UML demonstra sua versatilidade, sendo capaz de atuar desde a comunicação informal entre equipes até a construção automatizada de sistemas complexos.

Além dos aspectos técnicos e metodológicos da Engenharia de Software, a usabilidade é um fator crítico para o sucesso de sistemas interativos. Segundo Nielsen (1994), a usabilidade refere-se à facilidade com que os usuários podem aprender e utilizar um sistema para atingir seus objetivos de forma eficiente, eficaz e satisfatória. A usabilidade é especialmente relevante em sistemas de gerenciamento de demandas urbanas, onde a interação entre cidadãos e órgãos públicos deve ser intuitiva e transparente. Nielsen (1994) propôs um conjunto de 10 heurísticas de usabilidade, que são princípios gerais para avaliar e melhorar a qualidade da interação entre usuários e sistemas. Essas heurísticas incluem:

- Visibilidade do Status do Sistema: O sistema deve informar ao usuário o que está acontecendo, por meio de *feedbacks* claros e em tempo real.
- Correspondência entre o Sistema e o Mundo Real: O sistema deve utilizar linguagem e conceitos familiares ao usuário, seguindo convenções do mundo real.
- Controle e Liberdade para o Usuário: O sistema deve permitir que o usuário desfaça ações ou retorne a etapas anteriores.
- Consistência e Padrões: O sistema deve seguir padrões consistentes de design e funcionalidade.
- Prevenção de Erros: O sistema deve evitar que o usuário cometa erros, por meio de validações e mensagens de alerta.
- Reconhecimento em vez de Memorização: O sistema deve minimizar a necessidade de o usuário memorizar informações, exibindo opções e instruções de forma clara.
- Flexibilidade e Eficiência de Uso: O sistema deve atender tanto a usuários iniciantes quanto a experientes, oferecendo atalhos e personalizações.

- Estética e Design Minimalista: O sistema deve evitar informações desnecessárias, focando no que é relevante para o usuário.
- Auxílio aos Usuários para Reconhecer, Diagnosticar e Recuperar Erros: O sistema deve fornecer mensagens de erro claras e sugestões para correção.
- Ajuda e Documentação: O sistema deve oferecer ajuda e documentação de fácil acesso, mesmo que não seja necessária com frequência.

Tais heurísticas são amplamente utilizadas na avaliação de interfaces de usuário, permitindo identificar problemas de usabilidade e propor melhorias. Conforme destacado por Shneiderman e Plaisant (2016), a aplicação destes princípios resulta em sistemas mais intuitivos e eficientes, aumentando a satisfação dos usuários e reduzindo custos de treinamento e suporte.

## 3 METODOLOGIA

Este trabalho caracteriza-se como uma pesquisa de abordagem qualitativa, com caráter exploratório e descritivo. A pesquisa visa analisar soluções práticas para um sistema de gerenciamento de demandas urbanas para a cidade de Porto Velho. A natureza exploratória e descritiva justifica-se pela necessidade de compreender o cenário atual e descrever as características necessárias para um sistema eficiente e acessível.

Inicialmente, foi realizada uma pesquisa exploratória para identificar diversos sistemas de gerenciamento de demandas urbanas já existentes, tanto em âmbito nacional quanto internacional. Essa etapa envolveu conversas informais com um grupo de colegas de trabalho da área de tecnologia, a fim de reunir sugestões e percepções sobre sistemas similares.

Na sequência, procedeu-se à análise comparativa dos sistemas COLAB, aplicativo 156 (São José dos Campos) e NYC 311 (Nova Iorque). A análise foi realizada com base na usabilidade, na forma como os sistemas se relacionam com os usuários e na maneira como os órgãos responsáveis respondem às demandas registradas. Aspectos como interface, fluxo de comunicação e categorização de demandas também foram observados.

Com base nas análises realizadas, foram definidas as funcionalidades necessárias para o sistema proposto. Para representar essas funcionalidades e os requisitos do sistema, foram elaborados os diagramas de casos de uso e diagrama de classes, utilizando a Linguagem de Modelagem Unificada (UML). Esses diagramas possibilitaram visualizar de forma clara as interações entre usuários e sistema, as estruturas das classes e os fluxos de operação. Por fim, foi desenvolvido um protótipo de inter-

face utilizando a ferramenta Figma, com o objetivo de representar as telas principais do sistema proposto e demonstrar a navegação entre suas funcionalidades. O protótipo priorizou a experiência do usuário e a facilidade de uso, servindo como base para validação do conceito e para futuras implementações.

# 4 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados e discussões originados do presente estudo.

## 4.1 Estudo de Sistemas Existentes de Gerenciamento de Demandas Urbanas

A análise comparativa de plataformas de gestão de demandas urbanas foi realizada com base em três sistemas: COLAB, “156 ” (São José dos Campos) e “NYC 311” (Nova Iorque). O COLAB, uma plataforma brasileira, permite aos cidadãos registrar demandas urbanas e acompanhar o status das transações, utilizando “gamificação” para incentivo à participação. O “156”, aplicativo da prefeitura de São José dos Campos, centraliza o registro e o acompanhamento de demandas, proporcionando maior eficiência na gestão urbana. Já o “NYC 311”, plataforma da prefeitura de Nova York, integra vários órgãos públicos e oferece diversos canais de registro, como telefone, site e aplicativo, agilizando a resolução de problemas.

A avaliação desses sistemas foi realizada com base nas heurísticas de usabilidade de Nielsen (1994), permitindo identificar boas práticas e oportunidades de melhoria. O COLAB se destaca pela visibilidade do status do sistema, garantindo transparência e atualização contínua das demandas. O “156”, embora eficiente, poderia aprimorar a prevenção de erros, incluindo validações em tempo real. O “NYC 311”, por sua vez, segue padrões consistentes de design, facilitando a navegação e promovendo uma experiência mais intuitiva para os usuários. Além disso, estudos apontam que a falta de integração entre órgãos públicos, como observada no “156”, pode gerar atrasos no processamento de demandas. Em contrapartida, plataformas como o COLAB demonstram resultados positivos na participação cidadã e na transparência da gestão pública.

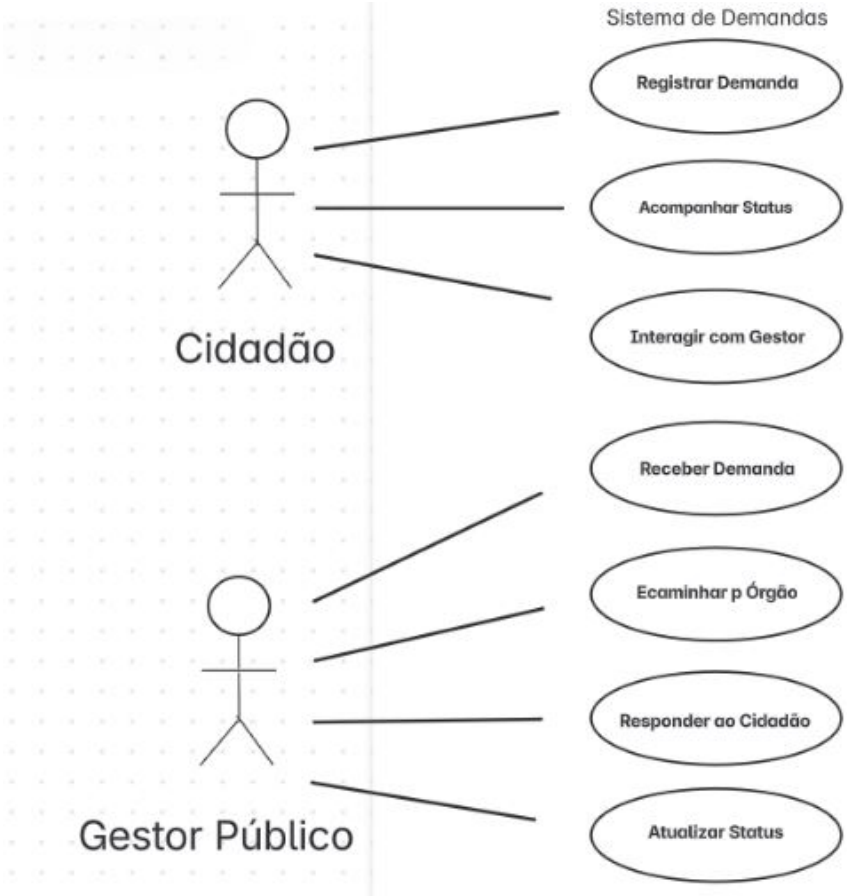
Com base na análise dos sistemas existentes, foi possível identificar práticas práticas e desafios que influenciam a eficiência das plataformas. A visibilidade do status no COLAB e a consistência do “NYC 311” são exemplos de abordagens bem-sucedidas, enquanto a falta de interoperabilidade entre órgãos públicos permanece como um obstáculo significativo. Esses *insights* poderão servir como referência para o desenvolvimento do sistema proposto, garantindo que ele atenda às necessidades

dos usuários e supere as limitações das soluções atuais.

## 4.2 Descrição das Funcionalidades e Especificações Técnicas do Sistema

Esta seção apresenta as funcionalidades e especificações técnicas do sistema proposto, com base no protótipo desenvolvido na ferramenta gratuita Figma. As telas principais foram estruturadas para atender às necessidades dos usuários, garantindo uma interface intuitiva e eficiente para o registro e acompanhamento de demandas urbanas.

Figura 1 – Diagrama de Casos de Uso

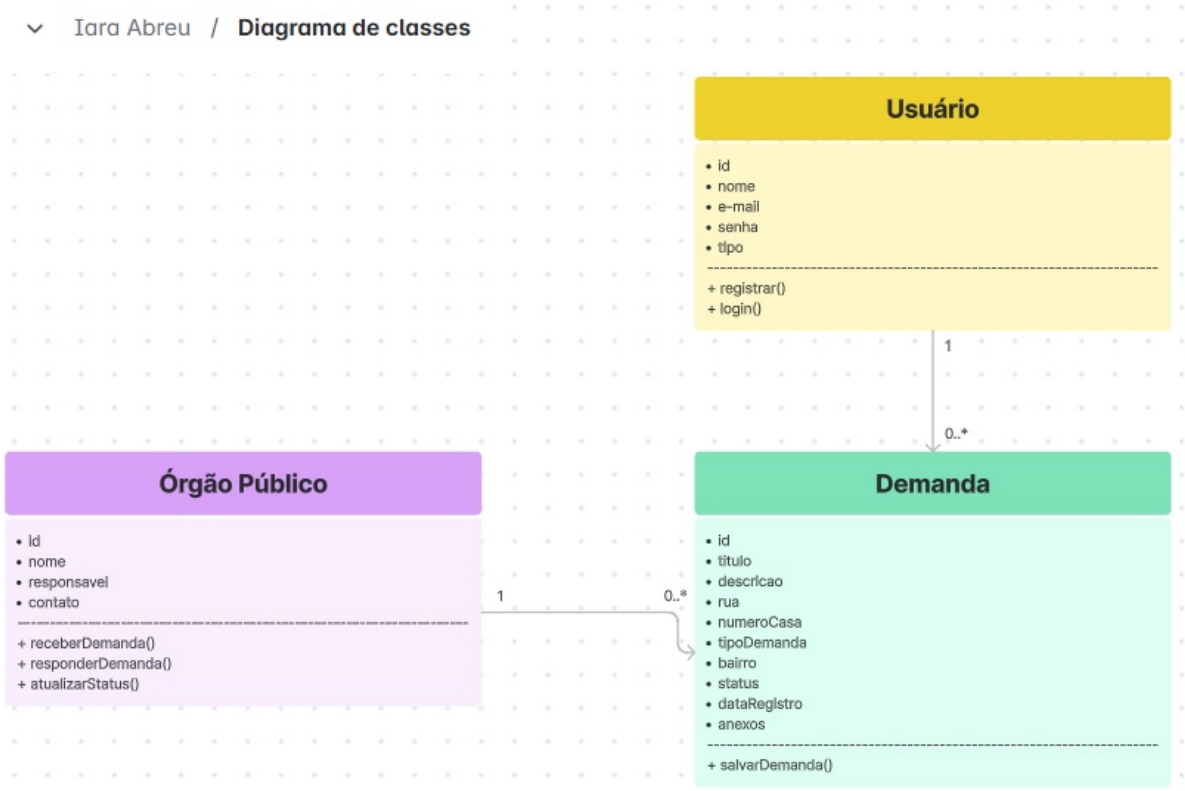



Fonte: A própria autora

A modelagem de requisitos foi realizada com a Linguagem de Modelagem Unificada (UML), com o objetivo de representar as funcionalidades do sistema de forma clara e organizada. No sistema, há dois tipos de usuários: o cidadão e o gestor público. O cidadão pode registrar exigências, acompanhar o status das obrigações e interagir com os órgãos públicos. O gestor público, por sua vez, é responsável por receber, analisar e encaminhar as demandas para os órgãos competentes. Entre as principais funcionalidades, o registro de demanda permite que o cidadão cadastre uma nova solicitação, informando o título, descrição, localização e tipo de demanda, além da opção

de anexar imagens. A funcionalidade de acompanhamento possibilita ao usuário visualizar o status atualizado de suas concessões, incluindo as respostas dos órgãos públicos intermediadas pelo gestor. Já a interação direta entre o cidadão e o gestor facilita a comunicação e o esclarecimento de dúvidas sobre as demandas registradas.

Figura 2 – Diagrama de Classes




Fonte: A própria autora

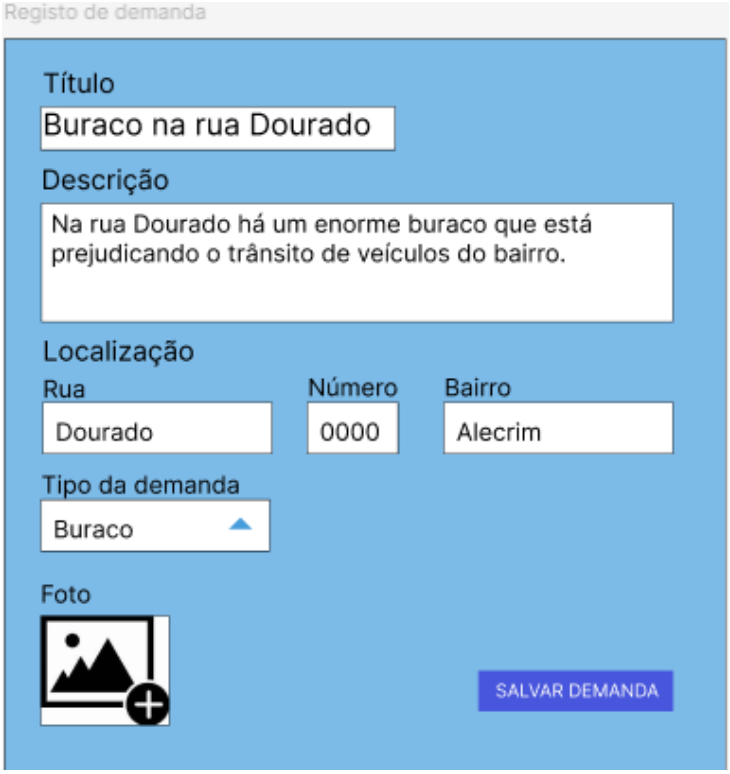
No que se refere às relações entre os usuários do sistema, o cidadão pode registrar e acompanhar suas propostas, além de receber *feedbacks* do órgão responsável. O gestor público, por sua vez, recebe as demandas, entra em contato com o órgão competente, recebe o retorno desse órgão e responde ao cidadão dentro do sistema. No fluxo de registro de demanda, o cidadão acessa a tela para inserir uma nova solicitação, momento em que o sistema exibe um formulário de registro. Após preencher os campos obrigatórios e enviar a demanda, o sistema valida as informações e os armazenamentos no banco de dados. Em seguida, uma notificação é enviada ao gestor público, que, por sua vez, encaminha a solicitação ao órgão responsável para o devido tratamento.

Além disso, foi desenvolvido um diagrama de classes, no qual foram definidas as principais classes e seus relacionamentos, considerando as entidades envolvidas no sistema, como Usuário, Demanda e Órgão Público, conforme apresentado na Figura 2.

- A classe Usuário possui os atributos `id, nome, e-mail, senha` e `tipo` (que definem se o usuário é um cidadão ou um gestor público), além dos métodos `registrar()` e `login()`.
- A classe `Demanda` contém os atributos `id`, `titulo`, `descricao`, `rua`, `numeroCasa`, `tipoDemanda`, `bairro`, `status`, `dataRegistro` e `anexos`, além do método `salvarDemanda()`.
- A classe `OrgaoPublico` possui os atributos `id`, `nome`, `responsável` e `contato`, e os métodos `receberDemanda()`, `responderDemanda()` e `atualizarStatus()`.

Os relacionamentos entre essas classes indicam que um usuário pode registrar diversas demandas, enquanto um órgão público pode receber e responder a diversas demandas. Esse modelo estrutural garante a organização e o correto funcionamento do sistema.

Figura 3 – Tela de Registro de Demandas



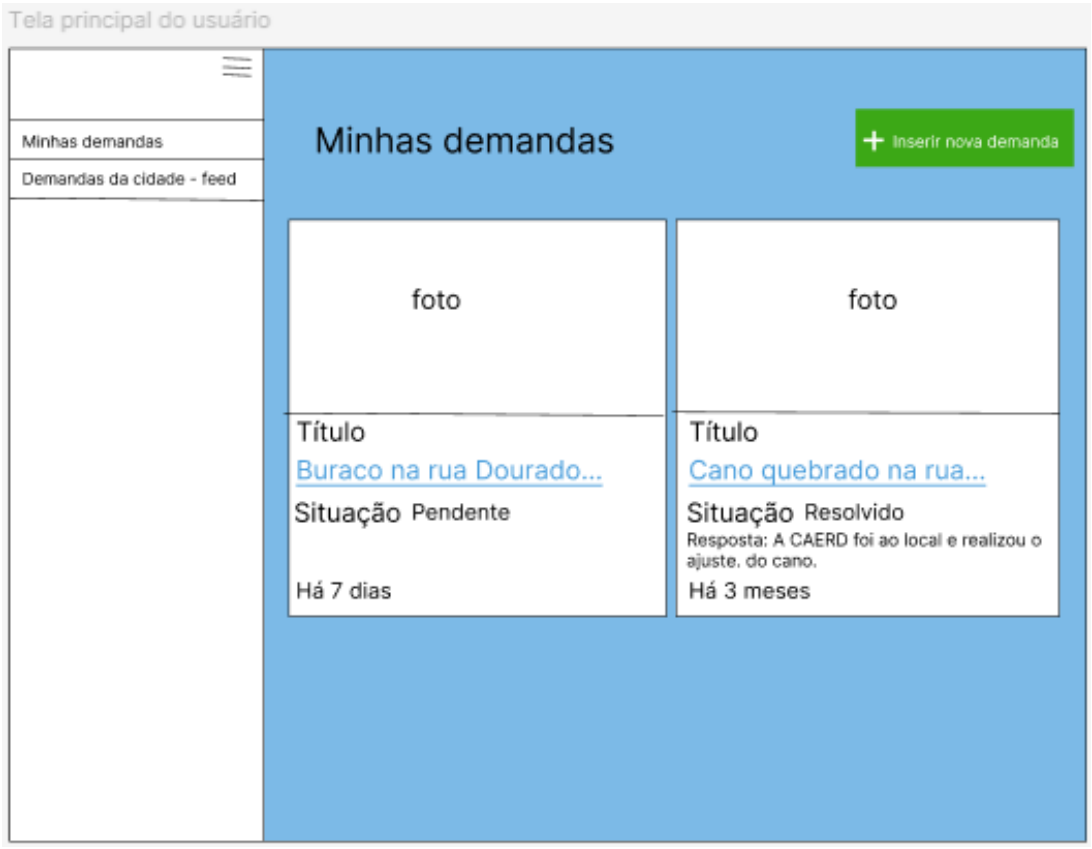

Fonte: A própria autora

## 4.3 Protótipo de Interface

O protótipo de interface foi desenvolvido utilizando a ferramenta gratuita Figma, com foco na usabilidade e na experiência do usuário. As telas principais são mostradas a seguir, com base nas imagens do protótipo.

A Figura 3 apresenta uma interface destinada ao registro de demandas pelos cidadãos, permitindo relatar problemas urbanos, como buracos nas ruas, falhas na iluminação pública, questões de coleta de lixo, problemas com encanação e asfalto. A tela contém campos específicos para inserção do título e descrição da demanda, além da localização específica, incluindo rua, número e bairro. Também há um campo de seleção para o tipo de demanda, com opções como iluminação, saneamento e pavimentação. Além disso, o sistema possibilita o anexo de fotos para complementar a solicitação e conta com um botão para salvar a demanda registrada.

Figura 4 – Tela de Acompanhamento de Solicitações




Fonte: A própria autora

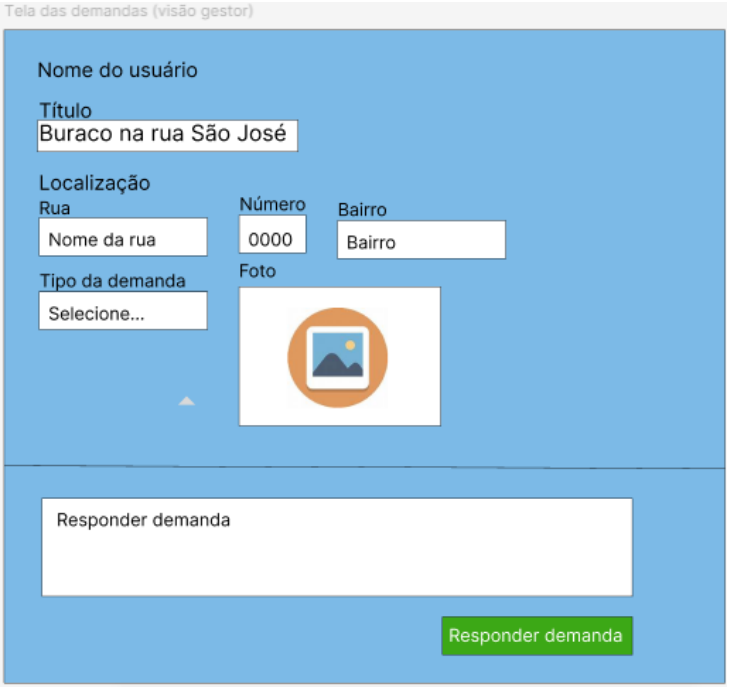
A Figura 4 apresenta a interface destinada ao acompanhamento das solicitações registradas pelos cidadãos. Nessa tela, os usuários podem visualizar a lista de demandas enviadas, acompanhadas de seu status atual, como “Pendente”, “Em execução” ou “Resolvido”. Além disso, é possível acessar detalhes individuais de cada demanda, incluindo a quantidade de dias desde o registro e as atualizações realizadas ao longo do processo. Essa funcionalidade visa proporcionar maior transparência e controle sobre o andamento das solicitações.

Figura 5 – Tela Geral de Registro de Demandas


Fonte: A própria autora

As Figuras 5 e 6 apresentam a interface destinada ao agente administrador dos órgãos públicos, responsável por gerenciar as demandas enviadas pelos cidadãos. Nessa tela, o gestor pode visualizar todas as solicitações registradas, filtrar demandas conforme critérios específicos e acompanhar o histórico de status de cada solicitação.

Figura 6 – Tela das Demandas (visão do Gestor)




Fonte: A própria autora

Ao selecionar uma demanda específica (Figura 6), o gestor tem acesso aos detalhes enviados pelo cidadão e pode fornecer uma resposta diretamente pelo sistema, garantindo um fluxo eficiente de comunicação e encaminhamento das demandas para os órgãos competentes.

## 5 CONCLUSÃO

Ao analisar a proposta deste trabalho foi possível compreender as etapas e processos envolvidos na análise e na concepção para um sistema de gerenciamento de demandas urbanas. A fundamentação teórica abordada permitiu apresentar conceitos relevantes de Engenharia de Software, com ênfase no ciclo de vida do software, na Engenharia de Requisitos, na modelagem UML e na usabilidade, os quais são essenciais para o planejamento e o desenvolvimento de soluções tecnológicas aplicadas à gestão pública.

A análise de sistemas existentes, como o COLAB, o “156” e o “NYC 311”, contribuiu significativamente para a identificação de boas práticas e aspectos relevantes relacionados à interação entre cidadãos e poder público. Essa etapa evidenciou a importância de se considerar a experiência do usuário, a acessibilidade e a eficiência dos serviços ofertados, aspectos que impactam diretamente na adesão e na efetividade de sistemas desse gênero. A metodologia adotada possibilitou a realização de um levantamento de requisitos bem estruturado, a elaboração de protótipos e a modelagem de diagramas UML, proporcionando uma visão clara e sistematizada do funcionamento do sistema proposto. As técnicas empregadas reforçam a necessidade de se investir na análise criteriosa de requisitos e na aplicação adequada de recursos de modelagem, uma vez que tais etapas influenciam diretamente na qualidade final, na usabilidade e na manutenção do software.

Dessa forma, conclui-se que a proposta deste trabalho cumpre seu objetivo de apresentar uma análise voltada à criação de um protótipo e análise de um sistema de gerenciamento de demandas urbanas, destacando-se como subsídio para futuras iniciativas na área de tecnologia para a administração pública.

Como sugestão para trabalhos futuros, recomenda-se a implementação prática do sistema proposto, permitindo validar os requisitos levantados e as funcionalidades modeladas neste estudo. Além disso, seria relevante realizar testes de usabilidade com usuários reais para avaliar a experiência de uso e identificar melhorias. Outras possibilidades incluem a integração do sistema com bases de dados geográficas e ferramentas de geolocalização para otimizar o atendimento das demandas.

# REFERÊNCIAS

FERNANDES, E. **Direito Urbanístico**. 5. ed.: Fórum, 2011.

FOWLER, Martin. **UML Essencial**. 3. ed.: Alta Books, 2005.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo 2022 Panorama**. 2022. `https://censo2022.ibge.gov.br/panorama/`. Acesso em 14/08/2024.

NIELSEN, Jakob. **Usability engineering**. Morgan Kaufmann, 1994.

PFLEEGER, Lawrence; ATLEE, Joanne M. **Software engineering: Theory and practice by shari**. Prentice Hall, 2010.

PRESSMAN, Roger S; MAXIM, Bruce R. **Engenharia de software-9**. McGraw Hill Brasil, 2021.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 8. ed.: AMGH, 2016.

REINEHR, Sheila. **Engenharia de Requisitos**. SAGAH, 2020.

SHNEIDERMAN, Ben; PLAISANT, Catherine. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. 6. ed.: Pearson, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed.: Pearson, 2019.

WIEGERS, Karl; BEATTY, Joy. **Software requirements**. Pearson Education, 2013.