

Laboratório de Engenharia de Software

Projeto: Plataforma de Acessibilidade Urbana com
IA

São Paulo
2025

Conteúdo

1 Introdução

O Projeto em questão é uma aplicação Web que tem como ponto central a sistematização do processo de relato, monitoramento e divulgação das falhas de problemas de acessibilidade, na rede de transporte público de grandes centros urbanos. A proposta é inspirada pelo Objetivo de Desenvolvimento Sustentável (ODS) de número 11, sobre Cidades e Comunidades Sustentáveis, da Organização das Nações Unidas (ONU).

A plataforma proposta utiliza tecnologias acessíveis para criar um mapa colaborativo de barreiras de acessibilidade urbana na rede pública de transporte, transformando dados enviados por cidadãos em informações estruturadas e acionáveis para gestão pública.

2 Definição da Demanda

2.1 Problema/Oportunidade Percebida

As cidades brasileiras enfrentam graves deficiências em acessibilidade urbana, impactando diretamente a qualidade de vida de pessoas com mobilidade reduzida, idosos, usuários de cadeiras de rodas, pessoas com deficiência visual e famílias com carrinhos de bebê. Os principais problemas identificados incluem:

- **Falta de dados estruturados:** inexistência de base centralizada e atualizada sobre barreiras de acessibilidade
- **Informações fragmentadas:** relatos dispersos em redes sociais e canais isolados sem sistematização
- **Ausência de priorização técnica:** gestores públicos tomam decisões sem dados concretos sobre impacto e urgência
- **Invisibilidade do problema:** barreiras não documentadas perpetuam a exclusão social
- **Alto custo de mapeamento manual:** levantamentos tradicionais demandam recursos humanos e tempo excessivos
- **Baixa transparência social:** ausência de acesso público e acompanhamento das ações reduz a confiança e a participação cidadã

2.2 Razão/Justificativa da Demanda

A demanda por uma solução tecnológica de mapeamento de acessibilidade urbana se justifica pelos seguintes fatores:

- **Impacto social:** 17,2 milhões de brasileiros com dificuldade de locomoção (IBGE 2019) enfrentam perda de autonomia e dignidade
- **Conformidade legal:** Necessidade de cumprimento da Lei Brasileira de Inclusão (Lei 13.146/2015)
- **Eficiência na gestão pública:** Má alocação de recursos públicos em intervenções não prioritárias devido à falta de dados

- **Inclusão social:** Barreiras arquitetônicas perpetuam a exclusão de uma parcela significativa da população
- **Sustentabilidade urbana:** Alinhamento com os Objetivos de Desenvolvimento Sustentável da ONU

2.3 Descrição do Produto de Software

A solução proposta é uma plataforma open-source que utiliza tecnologias acessíveis para criar um mapa colaborativo e inteligente de barreiras de acessibilidade urbana. Os componentes principais incluem:

Principais componentes:

- **Sistema de Coleta Colaborativa:** cidadãos reportam problemas com fotos e geolocalização.
- **Plataforma de Dados Abertos:** disponibilização pública dos dados.
- **Mapa Interativo:** visualização em tempo real das barreiras.
- **Painel de Gestão:** dashboard para gestores públicos e sociedade civil acompanharem indicadores.

2.4 Clientes, Usuários e Demais Grupos de Interesse

Usuários Primários:

- Pessoas com deficiência física, visual ou mobilidade reduzida
- Idosos
- Famílias com crianças pequenas (carrinhos de bebê)
- Cidadãos engajados em causas de acessibilidade

Clientes/Beneficiários:

- Prefeituras e órgãos de gestão urbana
- Secretarias de mobilidade e acessibilidade
- Organizações da sociedade civil
- Pesquisadores e acadêmicos

Grupos de Interesse:

- Conselhos municipais de pessoas com deficiência
- Empresas de tecnologia assistiva
- Mídia e formadores de opinião
- Organismos internacionais (ONU, Banco Mundial)

2.5 Etapas de Desenvolvimento

Visão geral: o semestre (16 semanas) será dividido em fases iterativas com entregáveis claros para garantir validação contínua e integração entre front, back e dados.

- **Fase 0 — Preparação (Semana 1)**

- Atividades: alinhamento de requisitos, definição do escopo do MVP e critérios de sucesso.
- Entregáveis: backlog priorizado, cronograma detalhado e ambiente de desenvolvimento inicial.

- **Fase 1 — Protótipo Web Mockado (Semanas 2–4)**

- Atividades: criar protótipo PWA simples (mock de dados) para fluxos principais (cadastro de ocorrência, mapa, painel gestor).
- Entregáveis: protótipo navegável (Figma/React básico) e roteiro de testes de usabilidade.

- **Fase 2 — Análise de Casos de Uso e Entidades (Semanas 4–6)**

- Atividades: mapear atores e casos de uso prioritários; identificar entidades, atributos e relacionamentos.
- Entregáveis: lista de casos de uso, dicionário de dados e modelo conceitual (ER alto nível).

- **Fase 3 — Diagrama de Classes de Domínio e Modelagem (Semanas 6–8)**

- Atividades: construir diagrama de classes do domínio com responsabilidades e agregados; definir serviços de domínio.
- Entregáveis: diagrama de classes e especificação de métodos principais por classe/domain service.

- **Fase 4 — Diagramas de Sequência e Identificação de Métodos (Semanas 8–9)**

- Atividades: criar diagramas de sequência para fluxos críticos (reportar barreira, validar, atualizar status); derivar APIs e métodos necessários.
- Entregáveis: diagramas de sequência e lista de endpoints/métodos com contratos iniciais.

- **Fase 5 — Definição da Estrutura do Banco de Dados (Semanas 9–10)**

- Atividades: converter modelo conceitual em esquema físico (PostGIS), definir índices geoespaciais, políticas de privacidade/anonimização.
- Entregáveis: script DDL inicial, diagrama ER físico e plano de migração.

- **Fase 6 — Implementação da API CRUD e Integração (Semanas 10–13)**

- Atividades: desenvolver API REST/GraphQL com endpoints CRUD para entidades principais; conectar ao banco; autenticação básica JWT.

- Entregáveis: API funcional com testes automatizados, documentação OpenAPI e pipeline CI básico.
- **Fase 7 — Integração Frontend-Backend e Testes (Semanas 13–14)**
 - Atividades: integrar protótipo web com APIs reais; validar fluxos fim-a-fim; testes de performance e acessibilidade (WCAG AA).
 - Entregáveis: versão integrada do MVP, relatórios de testes e correções prioritárias.
- **Fase 8 — Finalização, Documentação e Entrega (Semanas 15–16)**
 - Atividades: documentação do projeto (arquitetura, API, como rodar), preparar demonstração para stakeholders, planejar próximos ciclos.
 - Entregáveis: release do MVP em ambiente de staging, guia de contribuição e lista de trabalhos futuros priorizados.

Observações operacionais:

- Iterar com validação de usuários durante cada fase curta (sprints de 1 semana).
- Deixar o projeto aberto para evolução: APIs versionadas, dados exportáveis e roadmap para IA e integrações externas.

2.6 Critérios de Qualidade

Confiabilidade:

- Detecção de barreiras com acurácia superior a 85% e menos de 5% de falsos positivos
- Disponibilidade do sistema > 99%

Usabilidade:

- Interface simples, inclusiva e responsiva
- Interface acessível seguindo WCAG 2.1 nível AA
- Compatibilidade com leitores de tela
- Design responsivo para diversos dispositivos

Performance:

- Tempo de resposta da API < 2 segundos
- Sistema deve processar e validar, via IA, até 1000 relatórios por hora

Transparência:

- Algoritmos auditáveis e documentação pública

Segurança e privacidade:

- Conformidade com a LGPD

Escalabilidade:

- Suporte a múltiplas cidades e grandes volumes de dados
- Sistema deve ser escalável para suportar um aumento de 50% no número de usuários

Engajamento:

- Incentivo à participação cidadã com feedback sobre impacto
- Mínimo de 1000 usuários ativos/mês por cidade

3 Requisitos do Produto

3.1 Requisitos Funcionais

| ID | Tipo | Prioridade | Descrição do Requisito |
|------|------|------------|---|
| RF01 | RF | Alta | O sistema deve permitir visualizar mapa com geolocalização de problemas de acessibilidade |
| RF02 | RF | Alta | O usuário deve gerar relatório com foto e descrição de novo problema |
| RF03 | RF | Alta | O problema deve aparecer no mapa após criação do relatório |
| RF04 | RF | Alta | O sistema deve notificar entidade responsável quando marcada |
| RF05 | RF | Alta | O sistema não deve permitir relatórios duplicados |
| RF06 | RF | Alta | O sistema deve usar IA para validar relatórios |
| RF07 | RF | Média | O usuário pode filtrar região específica no mapa |
| RF08 | RF | Média | O usuário deve marcar entidade responsável |
| RF09 | RF | Alta | Permitir cadastro com diferentes perfis (cidadão, gestor, validador) |
| RF10 | RF | Média | Gerar relatórios analíticos para gestores |
| RF11 | RF | Média | Permitir validação manual das classificações automáticas |
| RF12 | RF | Baixa | Exportar dados em formatos padrão (CSV, GeoJSON) |

3.2 Requisitos Não-Funcionais

| ID | Tipo | Prioridade | Descrição do Requisito |
|-------|------|------------|---|
| RNF01 | RNF | Alta | Tempo de resposta <2s para carregar mapa |
| RNF02 | RNF | Alta | Processar e validar até 1000 relatórios/hora via IA |

| ID | Tipo | Prioridade | Descrição do Requisito |
|-------|------|------------|--|
| RNF03 | RNF | Média | Compatível com iOS e Android recentes |
| RNF04 | RNF | Média | Disponibilidade 99% do tempo |
| RNF05 | RNF | Média | Escalável para 50% aumento de usuários |
| RNF06 | RNF | Média | Notificações em tempo real |
| RNF07 | RNF | Baixa | Segurança e proteção de dados |
| RNF08 | RNF | Baixa | Interface intuitiva e acessível |
| RNF09 | RNF | Alta | Conformidade WCAG 2.1 AA |
| RNF10 | RNF | Alta | Conformidade LGPD |

3.3 Categorias de Barreiras Detectáveis

1. Infraestrutura Física

- Escadas sem alternativa acessível
- Rampas fora de norma ($>8,33\%$ inclinação)
- Calçadas danificadas ou estreitas ($<1,20\text{m}$)
- Desníveis e buracos

2. Sinalização e Orientação

- Ausência de piso tátil
- Falta de sinalização visual/sonora
- Semáforos sem recurso sonoro

3. Obstáculos

- Mobiliário urbano mal posicionado
- Veículos estacionados irregularmente
- Obras sem passagem alternativa
- Comércio informal obstruindo passagem

4 Wireframes

4.1 Página Principal

Resumo inicial do sistema com estatísticas de locais, reports ativos e categorias de problemas mais comuns.

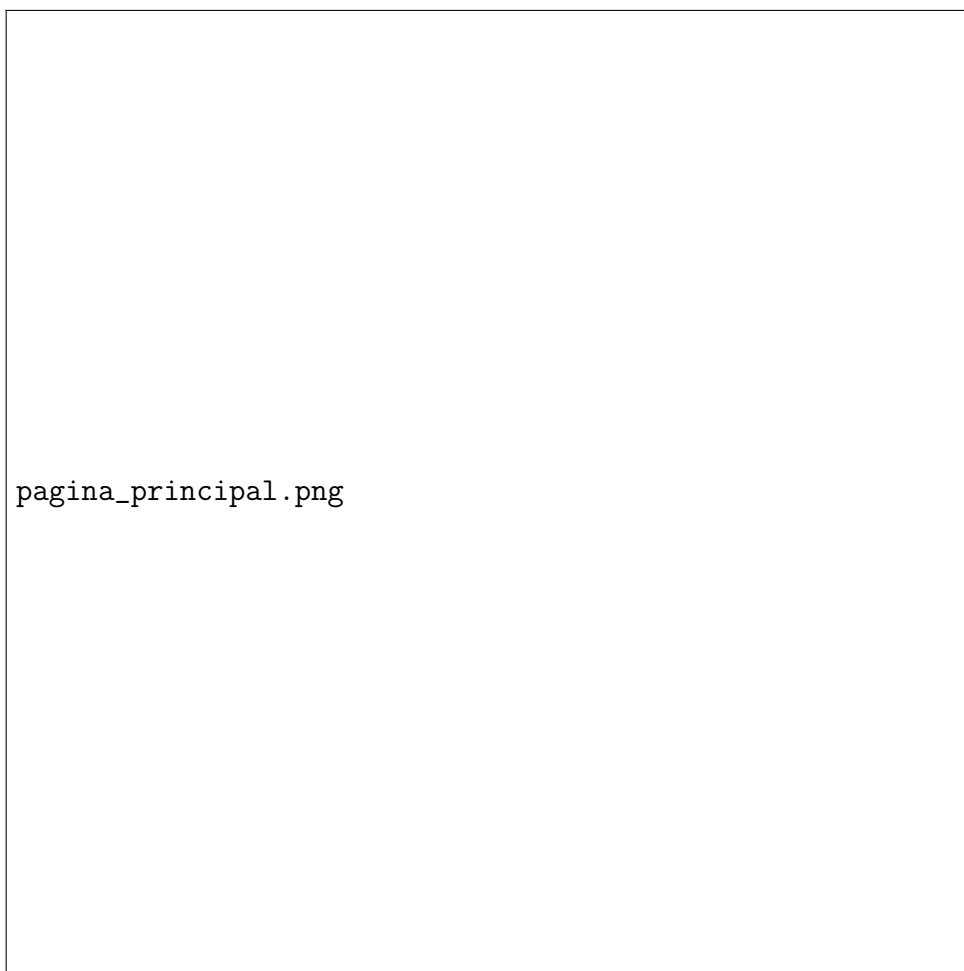


Figura 1: Página Principal

4.2 Área do Cliente (Login/Cadastro)

Tela de acesso ao sistema, onde o usuário entra com nome e e-mail para começar a reportar.

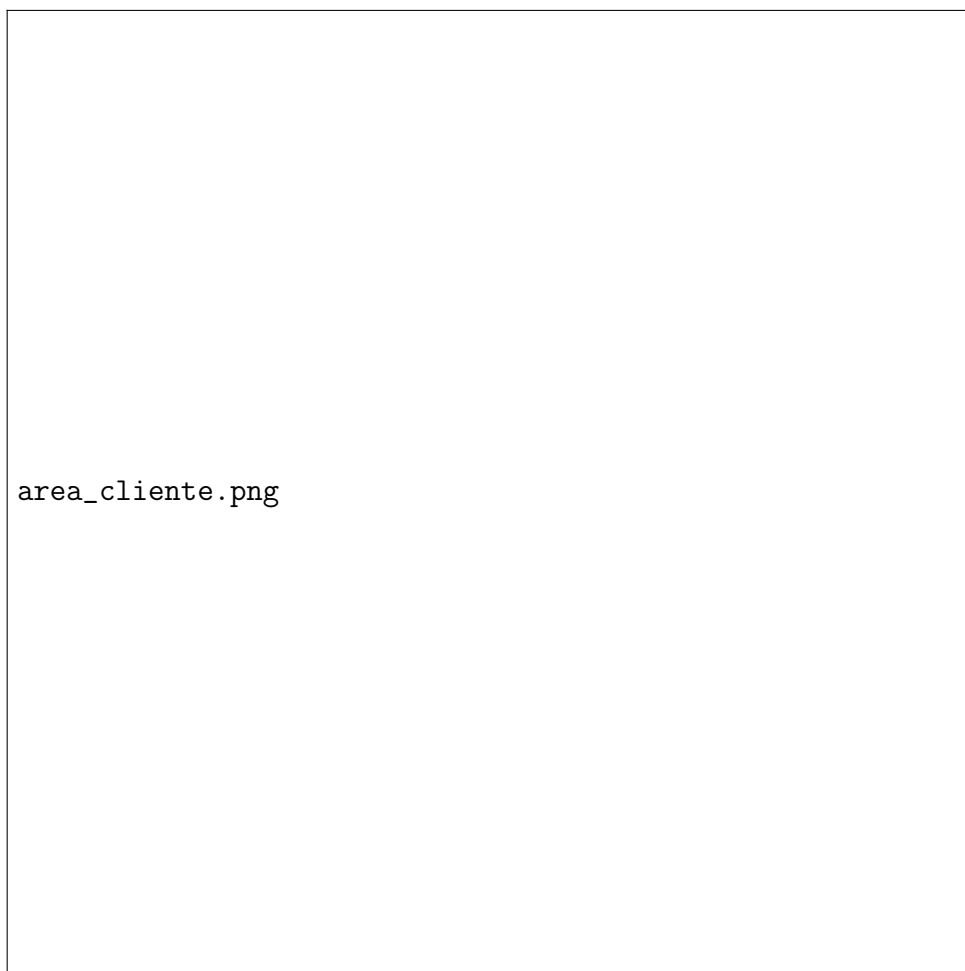


Figura 2: Área do cliente

4.3 Selecionar Local

Etapa onde o usuário escolhe o ponto de ônibus ou estação onde identificou o problema.



selecionar_local.png

Figura 3: Adicionar um reports

4.4 Reportar Problema – Tipo

Seleção da categoria do problema (rampa, piso tátil, elevador, etc.).



Figura 4: Reportar problema

4.5 Reportar Problema – Descrição

Formulário para detalhar o problema, com título e descrição completa.



Figura 5: Descrição do report

4.6 Dashboard – Visão Geral

Painel com estatísticas globais: número de reports, status, categorias e taxa de resolução.



Figura 6: Dashboards

4.7 Dashboard – Meus Reports

Lista dos problemas reportados pelo próprio usuário, com seus status.



dashboard_meus.png

Figura 7: Dashboards - Meus reports

4.8 Dashboard – Todos os Reports

Exibe todos os reports do sistema, permitindo visualizar detalhes e acompanhar pendências.



Figura 8: Dashboards - All reports

5 Modelagem ”Leve do Sistema”

—
—
— —

UML - Casos de Uso

[H] *wiki/uml/casos_uso.texDiagramadeCasosdeUso*
#I

5.0.1 Especificações dos Casos de Uso

5.1.1.1 Reportar Problema

| | |
|----------------------|--|
| Identificador | UC001 – Reportar Problema |
| Nome | Reportar Problema |
| Atores | Primário: Pedestre utilizando o Sistema de Reportes de Acessibilidade. Secundário: Administrador de um ponto registrado no Sistema. |

| | |
|---------------------------|--|
| Sumário | Um pedestre cria na plataforma um reporte para um problema de acessibilidade, visto em ponto de transporte público |
| Complexidade | Média |
| Regras de Negócio | <ul style="list-style-type: none"> - RN001: Apenas usuários autenticados podem registrar reportes. - RN002: Todo reporte deve estar vinculado a um local existente no sistema. - RN003: O reporte deve conter obrigatoriamente um título e uma descrição mínima. - RN004: O sistema deve registrar data, hora e usuário responsável pelo reporte. - RN005: O reporte pode conter evidências (foto ou vídeo), mas estas não são obrigatórias. - RN006: O reporte não pode exceder o limite máximo de caracteres definidos pelo sistema (para título e descrição). |
| Pré-condições | <p>O pedestre deve possuir informações do problema (foto, vídeo).</p> <p>O pedestre deverá disponibiliza sua localização.</p> <p>O pedestre deve ter um cadastro para registrar um reporte.</p> |
| Pós-condição | <p>O Sistema terá um reporte com todas as informações do problema registrado em seu banco de dados</p> <p>O Sistema registrará o reporte e atualizará o mapa para que o reporte conste para a localização</p> |
| Pontos de Inclusão | UC004 – Autenticar Usuário (quando necessário). |
| Pontos de Extensão | UC005 – Notificar Administrador (para pontos monitorados). |

Fluxo Principal

| Ações do Ator | Ações do Sistema |
|---|---|
| 1. O Pedestre seleciona que deseja reportar um problema na página principal | 2. O Sistema exibe uma listagem dos pontos e estações e o mapa |
| 3. Escolhe a localização através da busca pelo nome ou endereço, ou ainda através do mapa | 4. O Sistema retorna e solicita a confirmação do pedestre para que ele prosiga |
| 5. O pedestre verifica as informações do local selecionado e confirma | 6. O Sistema exibe uma lista de categorias problemas e oferece outro passo de confirmação para o pedestre |
| 7. O pedestre confirma a seleção da categoria e local | 8. O Sistema solicita informações sobre o problema e oferece dicas para que o reporte tenha boa qualidade |

| | |
|--|---|
| 9. O Pedestre registra um título para o problema e uma descrição que dê mais detalhes sobre a natureza dele e ao chegar as informações envia o reporte | 10. O Sistema informa o envio do reporte e solicita se o pedestre deseja criar um novo reporte ou visualizar o reporte publicado. |
|--|---|

Fluxo Alternativo – Pedestre não está logado no sistema

| Ações do Ator | Ações do Sistema |
|---|---|
| 1a. O pedestre seleciona a opção de reportar sem estar logado | 2a. O Sistema redireciona para a tela de login/cadastro (UC004 – Autenticar Usuário). |
| 3a. Após login/cadastro concluído | 4a. O Sistema retorna ao ponto em que o fluxo principal estava, permitindo prosseguir do passo 3. |

Fluxos de Exceção

| Ações do Ator | Ações do Sistema |
|--|--|
| 1e. O pedestre seleciona uma localização inexistente ou inválida | 2e. O sistema exibe mensagem de erro e solicita nova seleção de localização. |
| 2e. O pedestre não insere título ou descrição | 3e. O sistema informa que os campos são obrigatórios e impede o envio até que sejam preenchidos. |
| 3e. O pedestre tenta anexar arquivo em formato não suportado | 4e. O sistema rejeita o upload e informa quais formatos são permitidos. |
| 4e. O sistema encontra falha de conexão no envio do reporte | 5e. O sistema notifica falha técnica e oferece a opção de salvar localmente para tentar enviar depois. |

5.1.1.2 Consultar o Mapa de Problemas

| | |
|----------------------|--|
| Identificador | UC002 – Consultar Mapa de Problemas |
| Nome | Consultar Mapa de Problemas |
| Atores | Primário: Pedestre (usuário do sistema). Secundário: Administrador (para análise de ocorrências). |
| Sumário | O pedestre acessa o sistema para visualizar, no mapa, os problemas de acessibilidade reportados em pontos de transporte público. |
| Complexidade | Baixa |

| | |
|---------------------------|--|
| Regras de Negócio | <ul style="list-style-type: none"> - RN007: O mapa deve exibir apenas problemas confirmados e devidamente registrados. - RN008: O usuário pode aplicar filtros (categoria, data, status). - RN009: Cada problema deve estar associado a uma localização válida. - RN010: O sistema deve mostrar a data de registro e a situação do problema (pendente, em análise, resolvido). - RN011: Problemas com mais de 1 ano podem ser arquivados, mas ainda disponíveis mediante filtro avançado. |
| Pré-condições | O sistema deve conter reportes cadastrados. O pedestre precisa ter acesso à plataforma (não é obrigatório login apenas para consulta). |
| Pós-condição | O usuário terá acesso visual aos problemas existentes no mapa e poderá selecionar pontos para ver detalhes. |
| Pontos de Inclusão | UC006 – Visualizar Detalhes de Problema (quando o usuário seleciona um ponto específico). |
| Pontos de Extensão | UC001 – Reportar Problema (usuário pode criar um novo reporte a partir do mapa). |

Fluxo Principal

| Ações do Ator | Ações do Sistema |
|--|--|
| 1. O pedestre acessa a opção "Consultar mapa de problemas". | 2. O sistema carrega o mapa com todos os pontos e marcadores de problemas registrados. |
| 3. O pedestre pode aplicar filtros (categoria, data, status). | 4. O sistema atualiza o mapa exibindo apenas os problemas que atendem ao filtro. |
| 5. O pedestre navega pelo mapa e seleciona um marcador específico. | 6. O sistema exibe informações resumidas do problema e opção de ver detalhes (UC006). |

Fluxo Alternativo – Sem problemas cadastrados

| Ações do Ator | Ações do Sistema |
|---|--|
| 1a. O pedestre acessa o mapa sem que haja reportes cadastrados. | 2a. O sistema exibe mensagem "Nenhum problema cadastrado até o momento" e mostra apenas os pontos sem ocorrências. |

Fluxos de Exceção

| Ações do Ator | Ações do Sistema |
|--|--|
| 1e. O sistema não consegue carregar o mapa (falha na API de mapas ou conexão). | 2e. O sistema exibe mensagem de erro "Não foi possível carregar o mapa. Tente novamente mais tarde". |

| | |
|---|--|
| 2e. O usuário aplica um filtro inválido ou inexistente. | 3e. O sistema exibe mensagem "Filtro inválido" e mantém a última visualização correta. |
| 3e. O marcador selecionado não possui dados válidos (erro no registro). | 4e. O sistema exibe mensagem de inconsistência e oculta o marcador defeituoso. |

5.1.1.3 Atualizar Status de Problema

| | |
|----------------------------|---|
| Identificador | UC003 – Atualizar Status de Problema |
| Nome | Atualizar Status de Problema |
| Atores | Primário: Administrador (responsável por um ponto de transporte público). Secundário: Pedestre autor do reporte. |
| Sumário | O administrador acessa o sistema para alterar o status de um problema reportado, e o pedestre autor do reporte deve confirmar a resolução antes do status ser consolidado como "Resolvido". |
| Complexidade | Alta |
| Regras de Negócio | <ul style="list-style-type: none"> - RN012: Apenas administradores autenticados podem atualizar status. - RN013: O status pode assumir: Pendente, Em Análise, Resolvido Provisório, Resolvido Confirmado, Arquivado. - RN014: Toda atualização deve registrar data, hora e autor da alteração. - RN015: Ao alterar para "Resolvido Provisório", o sistema notifica o pedestre autor para validação. - RN016: O pedestre tem um prazo (ex.: 7 dias) para confirmar a resolução; se não houver resposta, o sistema automaticamente consolida como "Resolvido Confirmado". - RN017: Caso o pedestre rejeite a resolução, o status volta para Em Análise. |
| Pré-condições | O administrador deve estar autenticado. Deve existir pelo menos um problema reportado no ponto administrado. O pedestre autor deve estar vinculado ao reporte. |
| Pós-condição | O status do problema será atualizado no sistema, refletindo tanto a ação do administrador quanto a validação do pedestre. |
| Pontos de Inclusão | UC002 – Consultar Mapa de Problemas (para localizar o problema). |
| Pontos de Ex-tensão | UC006 – Visualizar Detalhes de Problema (para acompanhar histórico e status detalhado). |

Fluxo Principal

| Ações do Ator | Ações do Sistema |
|---|--|
| 1. O administrador acessa a lista/mapa de problemas reportados. | 2. O sistema exibe os problemas associados ao ponto. |
| 3. O administrador seleciona um problema específico. | 4. O sistema exibe detalhes do problema e status atual. |
| 5. O administrador escolhe a opção de atualizar status para "Resolvido Provisório". | 6. O sistema registra a alteração e envia notificação ao pedestre autor. |
| 7. O pedestre recebe a notificação e abre o problema. | 8. O sistema exibe detalhes da resolução e opções de confirmar ou rejeitar. |
| 9. O pedestre confirma a resolução. | 10. O sistema altera o status para "Resolvido Confirmado", registra histórico e atualiza o mapa. |

Fluxo Alternativo - Pedestre rejeita a resolução

| Ações do Ator | Ações do Sistema |
|-------------------------------------|--|
| 9a. O pedestre rejeita a resolução. | 10a. O sistema altera o status de volta para Em Análise e notifica o administrador. |

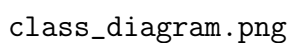
Fluxo Alternativo – Pedestre não responde

| Ações do Ator | Ações do Sistema |
|--|---|
| 9b. O pedestre não responde dentro do prazo (ex.: 7 dias). | 10b. O sistema consolida automaticamente o status como Resolvido Confirmado e registra no histórico. |

Fluxo de Exceção

| Ações do Ator | Ações do Sistema |
|--|--|
| 1e. O administrador não está autenticado. | 2e. O sistema bloqueia e redireciona para login. |
| 2e. O pedestre não consegue abrir o reporte (erro técnico). | 3e. O sistema exibe mensagem de erro e orienta tentar novamente. |
| 3e. O administrador tenta alterar status de problema fora de sua jurisdição. | 4e. O sistema impede a ação e informa falta de permissão. |

5.1 UML - Diagrama de Classe de Domínio



class_diagram.png

Figura 9: Diagrama de Classes

5.2 UML - Diagrama de Sequência

6 Descrição da Arquitetura e Ferramentas Utilizadas

6.1 Stack Tecnológico

Backend:

- Node.js com framework NestJS
- PostgreSQL com extensão PostGIS para dados geoespaciais

- Docker para containerização

Frontend:

- React + Next.js para aplicação web
- ShadcnUI para componentes de interface

Infraestrutura:

- Kubernetes para orquestração
- AWS S3 ou MinIO para armazenamento de objetos
- GitHub Actions para CI/CD

6.2 Padrões e Princípios

- Arquitetura de microsserviços
- API RESTful seguindo OpenAPI 3.0
- Autenticação via JWT
- Princípios SOLID e Clean Architecture
- Testes automatizados (unitários, integração, E2E)

7 Plano de Desenvolvimento

7.1 Cronograma Detalhado (16 semanas)

Fase 0 — Preparação (Semana 1)

- Alinhamento de requisitos e definição do escopo MVP
- Backlog priorizado e ambiente de desenvolvimento

Fase 1 — Protótipo Web (Semanas 2–4)

- Protótipo PWA com mock de dados
- Fluxos principais: cadastro, mapa, painel

Fase 2 — Análise e Modelagem (Semanas 4–6)

- Casos de uso e entidades
- Modelo conceitual e dicionário de dados

Fase 3 — Design de Classes (Semanas 6–8)

- Diagrama de classes do domínio
- Especificação de serviços

Fase 4 — Diagramas de Sequência (Semanas 8–9)

- Fluxos críticos modelados
- Definição de APIs e contratos

Fase 5 — Banco de Dados (Semanas 9–10)

- Esquema físico PostGIS
- Scripts DDL e migração

Fase 6 — Implementação API (Semanas 10–13)

- API REST/GraphQL com CRUD
- Testes automatizados e CI

Fase 7 — Integração e Testes (Semanas 13–14)

- Integração frontend-backend
- Testes E2E e acessibilidade

Fase 8 — Finalização (Semanas 15–16)

- Documentação completa
- Release MVP e roadmap futuro

7.2 Métricas de Sucesso

- **Técnicas:** Acurácia >85% na detecção
- **Impacto:** Taxa de resolução de barreiras

8 Resultados

[Esta seção será preenchida após a implementação, incluindo:

- *Métricas de desempenho do sistema*
- *Estatísticas de uso e engajamento*
- *Casos de sucesso e impactos mensuráveis*
- *Feedback dos usuários e gestores*
- *Lições aprendidas]*

9 Conclusão e Trabalhos Futuros

9.1 Conclusão Preliminar

O projeto visa criar uma solução tecnológica escalável e sustentável para o mapeamento colaborativo de barreiras de acessibilidade urbana, contribuindo diretamente para o cumprimento do ODS 11 e promovendo cidades mais inclusivas e acessíveis.

9.2 Próximos Passos

1. Formar equipe técnica multidisciplinar
2. Desenvolver prova de conceito
3. Estabelecer parceria com cidade piloto
4. Lançar MVP e coletar feedback
5. Iterar com base em dados reais

9.3 Trabalhos Futuros

- Integração com sistemas de transporte público
- Expansão para detecção de barreiras em ambientes internos
- Desenvolvimento de rotas acessíveis personalizadas
- Implementação de realidade aumentada para navegação
- Criação de índice de acessibilidade por cidade/bairro