Caminho Fácil

Projeto - Laboratório de Engenharia de Software

Alan Meniuk Gleizer Caio Vinicius Corsini Filho

Felipe Konishi Brum Gilberto De Melo Junior

Setembro de 2025

Sumário

1 Introdução

1.1 Sobre: Caminho Fácil

O produto consiste em um aplicativo web colaborativo que permite que cidadãos registrem e consultem informações sobre calçadas em um mapa interativo. Os usuários podem avaliar calçadas (nota de 1 a 5), enviar fotos, adicionar comentários e marcar itens de acessibilidade (largura, conservação, presença de piso tátil, condições para cadeirantes, carrinhos de bebê, etc).

A partir desses dados coletados, o sistema pode gerar relatórios sobre regiões específicas, úteis para cidadãos, associações e órgãos públicos.

Com isso o aplicativo pretende facilitar e auxiliar a locomoção inclusiva e digna de cidadãos que fazem parte de grupos vulneráveis, marginalizados e/ou pessoas que simplesmente necessitam de mais auxílio de locomoção em calçadas como pessoas portadoras de deficiência (PCD), mães com carrinhos de bebês, entre outros indivíduos.

Dessa forma, o projeto contribuirá com o Objetivo de Desenvolvimento Sustentável (ODS) 11: Cidades e Comunidades Sustentáveis, pois serve como uma forma de melhorar a infraestrutura urbana e incentivar o seu desenvolvimento para gerações futuras.

Figura 1: Exemplo ilustrativo do projeto *Caminho Fácil* (Fonte: Freepik).

2 Definição da Demanda

2.1 Problema

Grande parte das cidades brasileiras apresenta calçadas em mau estado de conservação, sem padronização de largura, com obstáculos e sem recursos de acessibilidade. Isso gera dificuldades para mobilidade urbana e acessibilidade universal.

2.2 Justificativa

A ausência de informações sistematizadas sobre o estado das calçadas dificulta tanto a vida dos cidadãos quanto a ação das prefeituras. Um sistema colaborativo pode centralizar dados, gerar relatórios e até apoiar decisões de políticas públicas.

2.3 Descrição do Produto

Aplicativo web que permite que usuários consultem e cadastrem informações sobre calçadas, incluindo avaliações, fotos e comentários.

O app é colaborativo e permite que permite que cidadãos registrem e consultem informações sobre calçadas em um mapa interativo, por meio de avaliações (nota 1 a 5), fotos, comentários e dados de acessibilidade. O sistema também consolida essas informações em relatórios úteis para cidadãos, associações e órgãos públicos.

2.4 Clientes

- Usuários finais: cidadãos, em especial pedestres com requisitos de acessibilidade, como pessoas usuárias de cadeira de rodas, responsáveis com carrinhos de bebê e pessoas com deficiência visual.
- Órgãos públicos: secretarias municipais, prefeituras.
- Sociedade civil: associações de bairro, ONGs ligadas à acessibilidade.

2.5 Casos de Uso Principais

- Avaliar calçada: cidadãos registram avaliação com nota (1–5 estrelas) e, opcionalmente, largura, estado de conservação e itens de acessibilidade (cadeira de rodas, carrinho de bebê, piso tátil).

- Consultar calçadas: cidadãos visualizam no mapa o estado de calçadas próximas, com notas, fotos e comentários.
- Gerar relatório: prefeituras, ONGs e associações selecionam uma área geográfica (ponto + raio) e recebem relatório com calçadas/ruas problemáticas, nota média da região e problemas mais citados.

2.6 Etapas de Desenvolvimento

1. Levantamento de requisitos e design da solução.
2. Modelagem de dados e definição da arquitetura.
3. Implementação (front-end, back-end, API e banco de dados).
4. Integração contínua e testes.
5. Implantação em nuvem (AWS).

2.7 Critérios de Qualidade

- Usabilidade simples e intuitiva.
- Dados georreferenciados consistentes.
- Desempenho adequado no carregamento do mapa.
- Segurança em autenticação e armazenamento de dados.
- Escalabilidade para múltiplas cidades.

3 Requisitos de Produto

3.1 Requisitos Funcionais

ID	Requisito
RF1	Apresentação de mapa para que o usuário possa localizar as calçadas
RF2	Localização de uma rua via mapa (touch) ou via pesquisa (digitação)
RF3	Visualização dos detalhes de cada calçada (avaliação média, fotos, comentários)
RF4	Funcionalidade de avaliação das calçadas (upload de foto opcional)

Tabela 1: Requisitos Funcionais do sistema

3.2 Requisitos Não Funcionais

ID	Requisito
NF1	Disponibilidade da aplicação por 24h 7 dias por semana
NF2	Proteção de dados pessoais dos usuários
NF3	Criptografia de dados sensíveis em trânsito usando protocolos HTTPS e TLS
NF4	Resposta a cada clique não pode demorar mais do que dois segundos
NF5	Compatibilidade com browsers: Chrome, Edge e Firefox
NF6	Compatibilidade com celulares Android e iOS
NF7	O sistema deve estar disponível em nuvem (AWS)
NF8	O banco de dados deve suportar pelo menos 10 mil registros iniciais
NF9	Testes automatizados de pelo menos 70% das funcionalidades

Tabela 2: Requisitos Não Funcionais do sistema

4 Protótipos

4.1 Protótipo Estático (Telas)

4.1.1 Caso de uso 1: Consultar calçadas

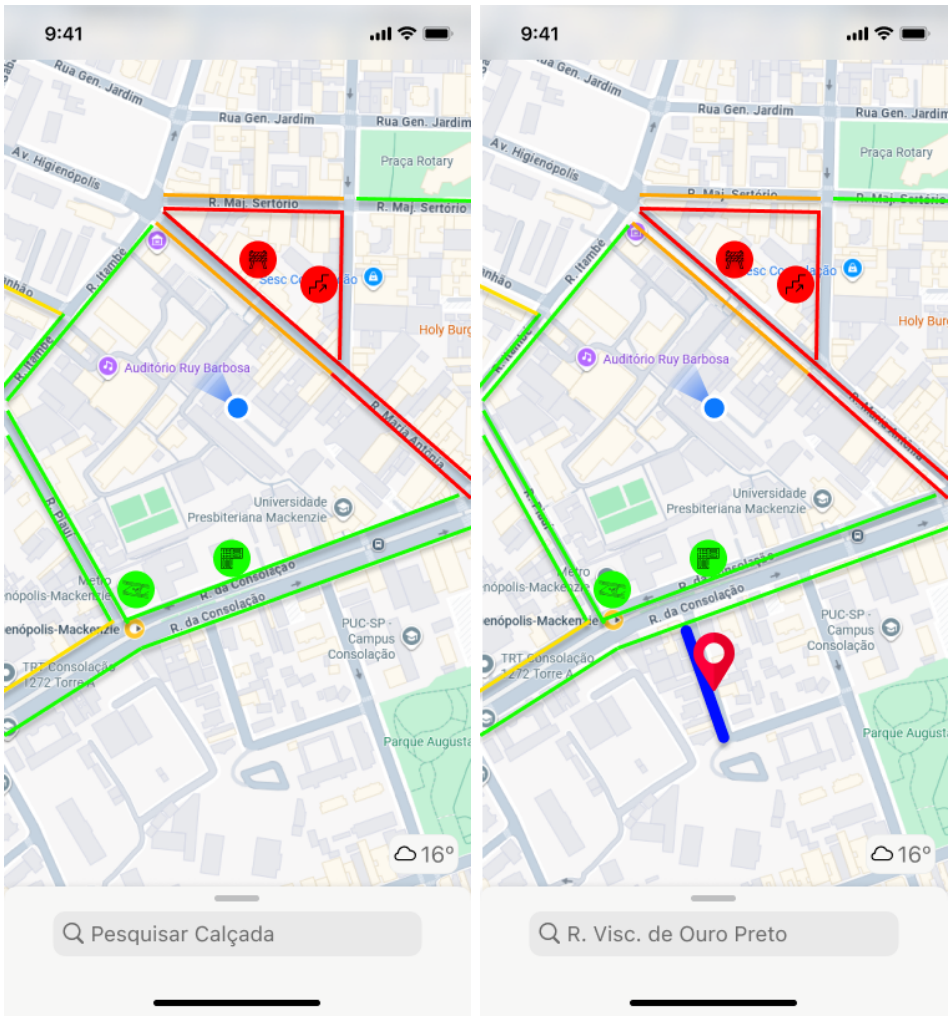

Figura 2: Fluxo inicial do caso de uso: Consultar calçadas.

Figura 3: Detalhes da calçada selecionada.

4.1.2 Caso de uso 2: Avaliar calçada

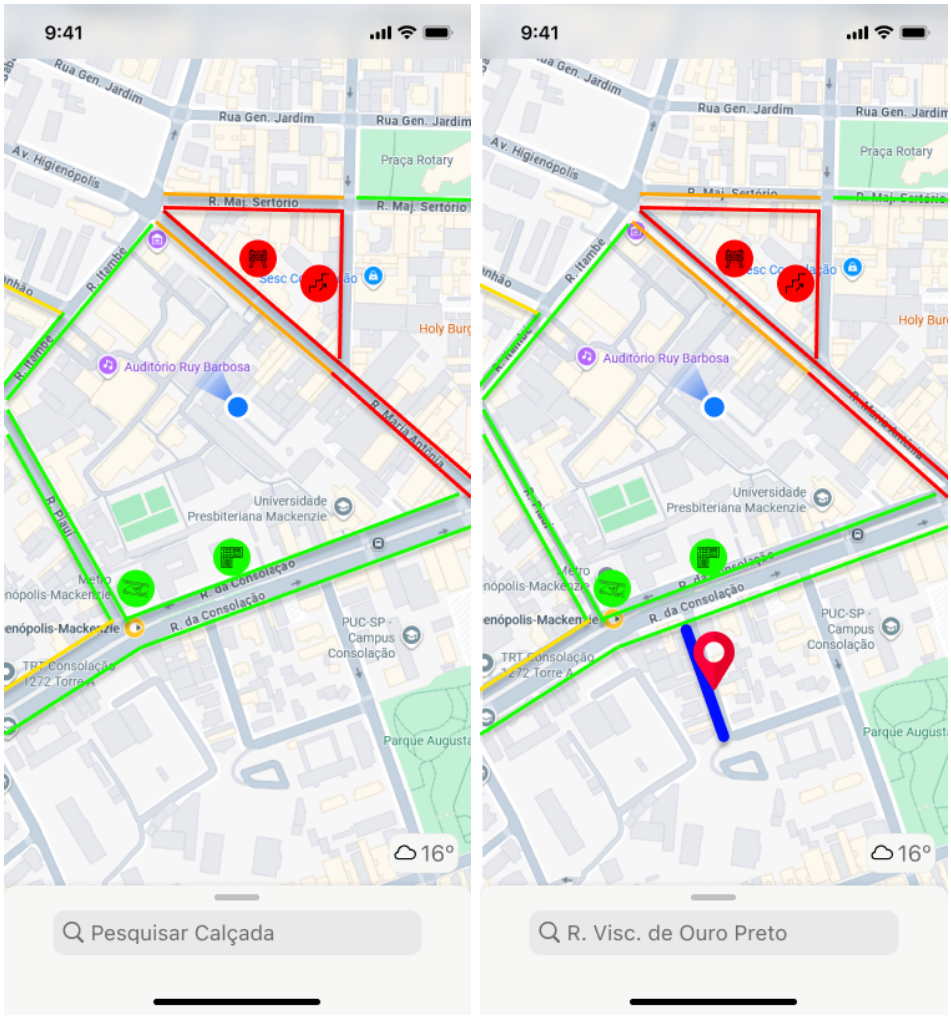

Figura 4: Fluxo inicial do caso de uso: Avaliar calçada.

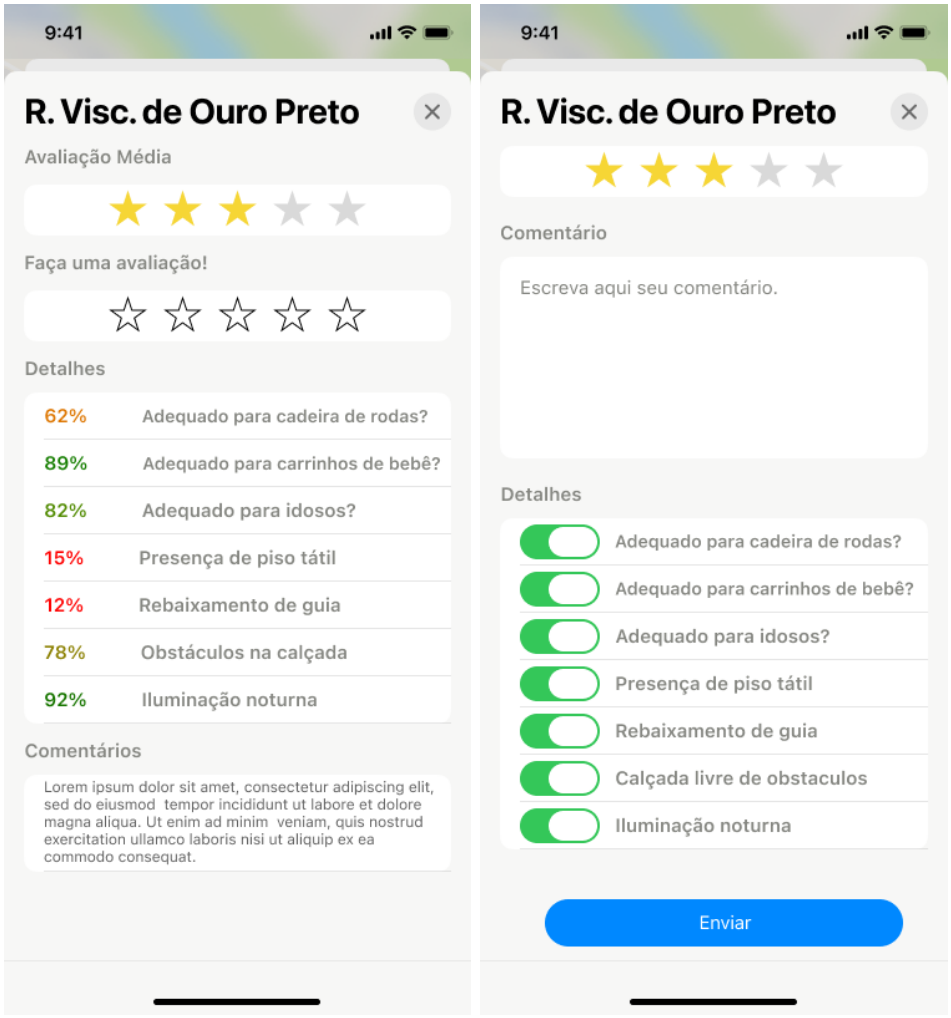

Figura 5: Detalhes da calçada e formulário de avaliação.

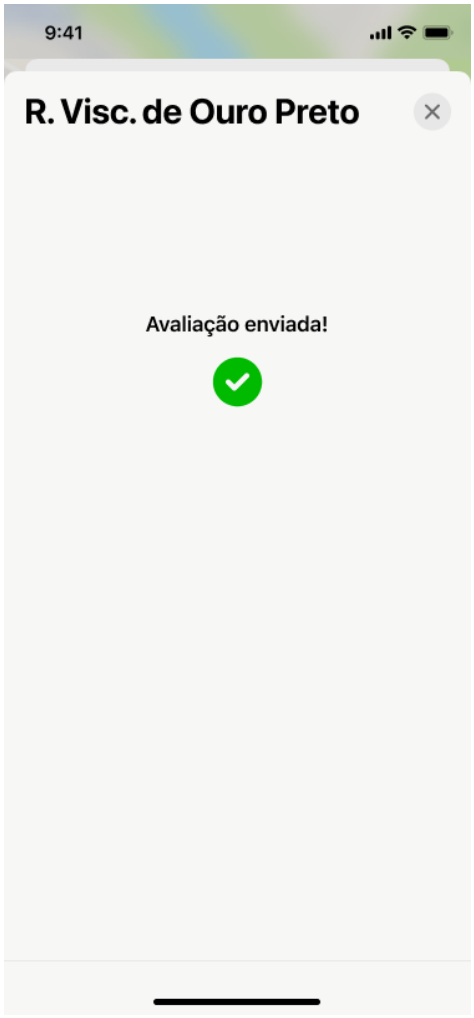

Figura 6: Confirmação da avaliação enviada.

4.1.3 Caso de uso 3: Gerar relatório

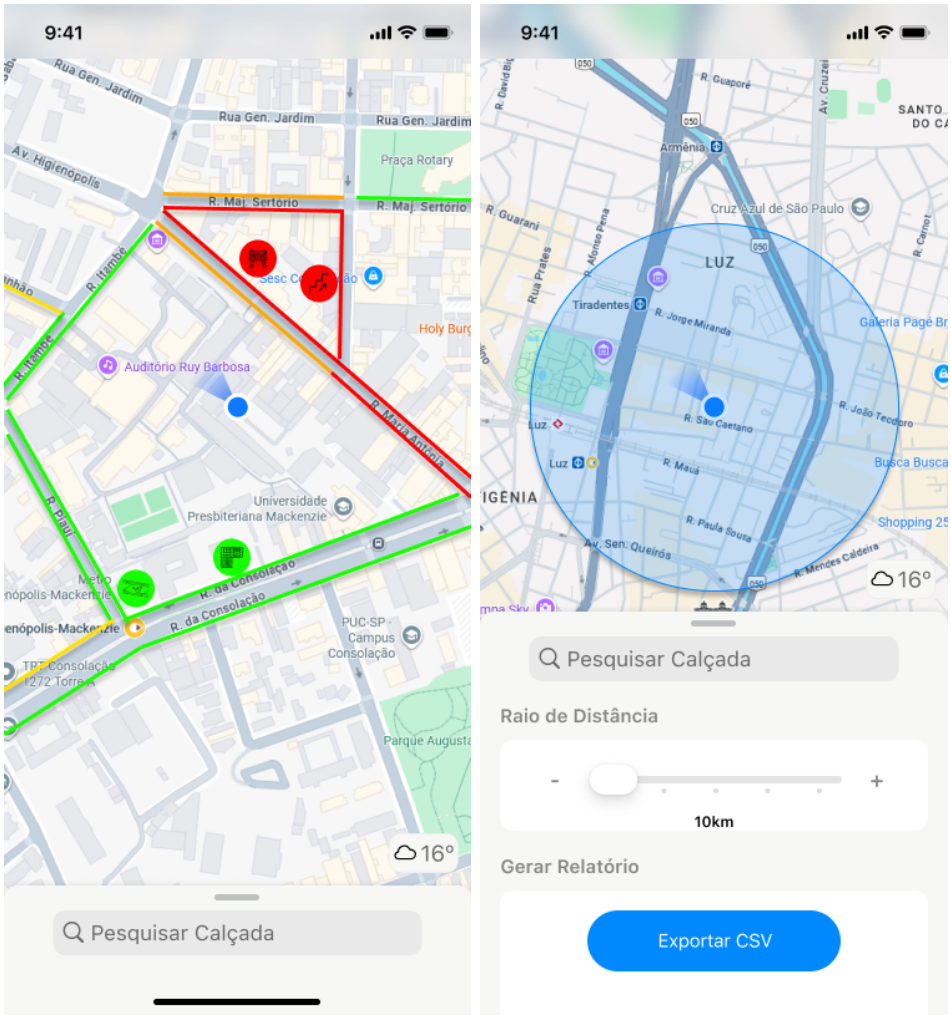

Figura 7: Acesso à funcionalidade de geração de relatórios.

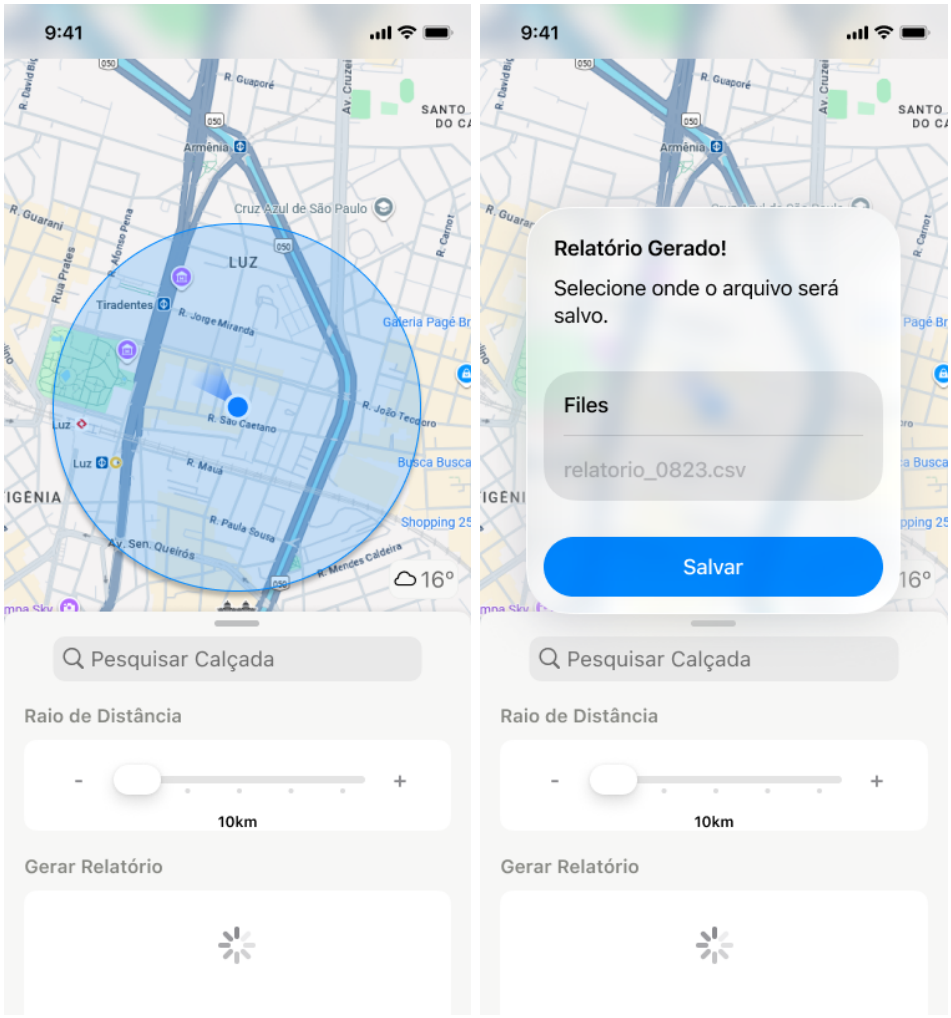

Figura 8: Configuração e visualização do relatório gerado.

5 Modelagem do Sistema

Nas próximas subseções se encontram os modelos UML do sistema. Incluindo: Diagramas de casos de uso, especificações, diagrama de classes de domínio e diagramas de sequência.

5.1 Diagrama de Casos de Uso

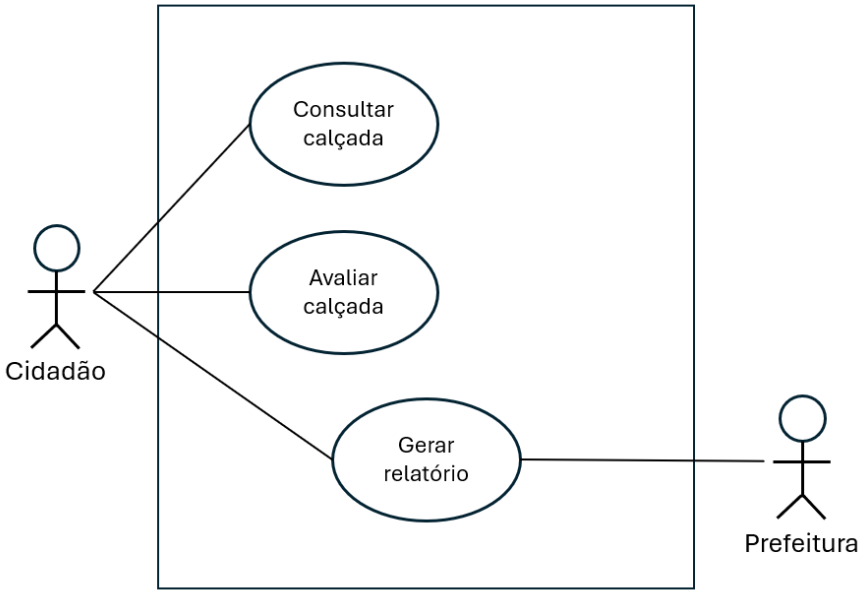

Figura 9: Diagrama de Casos de Uso do sistema.

5.2 Especificações de Casos de Uso

5.2.1 Caso de uso 1: Consultar calçadas

Elemento	Descrição
Ator	Cidadão (principal) OU Prefeitura (principal)
Pré-condições	1- Já existem calçadas avaliadas no sistema.
Fluxo básico	1- É aberto um grande mapa de São Paulo. 2- Cidadão seleciona algum local a sua escolha clicando diretamente em algum ponto do próprio mapa ou usando a ferramenta de busca. 3- Ao escolher, sistema mostra a região escolhida, com fotos, quantidade de estrelas e comentários feitos por outros usuários carregadas do banco de dados. 4- Ao terminar sua consulta, usuário fecha o sistema.
Fluxo alternativo	1a- Cidadão ingressa no sistema e digita seus credenciais (fazer login é opcional para esta atividade). 4a- Usuário clica em "mais fotos"para visualizar mais fotos da calçada escolhida. 4b- Usuário clica na opção "mais comentários"para visualizar mais comentários além dos iniciais. 4c- Usuário clica em "resumo de avaliações"abaixo da média de estrelas apresentada, para ver os números exatos de quantidades de estrelas que cada usuário deu.
Requisitos não funcionais	NF1 - NF4 - NF5 - NF6
Pós-condições	Nenhum estado do sistema é alterado, visto que o usuário não fez nenhuma avaliação

Tabela 3: Especificação do caso de uso: Consultar calçadas.

5.2.2 Caso de uso 2: Avaliar calçada

Elemento	Descrição
Atores	Cidadão (principal)
Pré-condições	1. A calçada alvo está localizável no mapa.
Fluxo básico	1. Ator seleciona uma calçada no mapa e clica em "Avaliar calçada". 2. Sistema exibe formulário com: nota (1–5 estrelas), comentário (opcional) e itens de acessibilidade (checkboxes: adequado para cadeira de rodas, adequado para carrinhos de bebê, adequado para idosos, presença de piso tátil, rebaixamento de guia, calçada livre de obstáculos, iluminação noturna). 3. Ator preenche os campos desejados e envia a avaliação. 4. Sistema valida os dados informados. 5. Sistema persiste a avaliação, associa à calçada, e atualiza métricas agregadas (ex.: média de estrelas, porcentagem de acessibilidade). 6. Sistema confirma o registro e exibe a avaliação na listagem da calçada.
Fluxo alternativo	4a. Validação falha: se a nota não for informada ou algum campo estiver inválido, o sistema exibe mensagem e solicita correção. 5b. Falha de persistência/timeout: o sistema informa erro e oferece opção de tentar novamente; nenhuma gravação parcial é publicada.
Requisitos não funcionais	NF1, NF2, NF3, NF4, NF5, NF6, NF7, NF8, NF9
Pós-condições	1. Avaliação registrada, associada à calçada, com métricas agregadas atualizadas e disponível para consulta pública.

Tabela 4: Especificação do caso de uso: Avaliar calçada.

5.2.3 Caso de uso 3: Gerar relatório

Elemento	Descrição
Atores	Prefeitura (principal), Associações de Bairro/ONGs (principal)
Pré-condições	1. Já existem registros de avaliações de calçadas no banco de dados.
Fluxo básico	1. Ator acessa a opção "Gerar relatório". 2. Sistema solicita parâmetros de filtro: ponto geográfico inicial (endereço ou coordenada) e raio de abrangência. 3. Ator define a área e confirma a solicitação. 4. Sistema consulta banco de dados e consolida as informações: calçadas avaliadas, média de notas, listagem de problemas reportados e uma coluna por cada categoria (ex.: largura insuficiente, obstáculos, falta de piso tátil, conservação). 5. Sistema gera relatório em formato tabular. 6. Sistema oferece opção de exportação em CSV. 7. Relatório é disponibilizado para download.
Fluxo alternativo	2a. Caso o ator insira parâmetros inválidos (raio negativo, ponto inexistente), o sistema exibe mensagem de erro e solicita correção. 3a. Caso não existam dados suficientes na região/intervalo definido, o sistema retorna mensagem: "Nenhuma informação encontrada para os critérios selecionados". 6a. Caso ocorra falha na geração de arquivo, o sistema informa o erro e mantém visualização na tela.
Requisitos não funcionais	NF1, NF2, NF3, NF4, NF5, NF7, NF8, NF9
Pós-condições	1. Relatório gerado e exibido ao ator. 2. Se exportado, arquivo em CSV é armazenado temporariamente no servidor e disponibilizado para download.

Tabela 5: Especificação do caso de uso: Gerar relatório.

5.3 Diagrama de Classe de Domínio

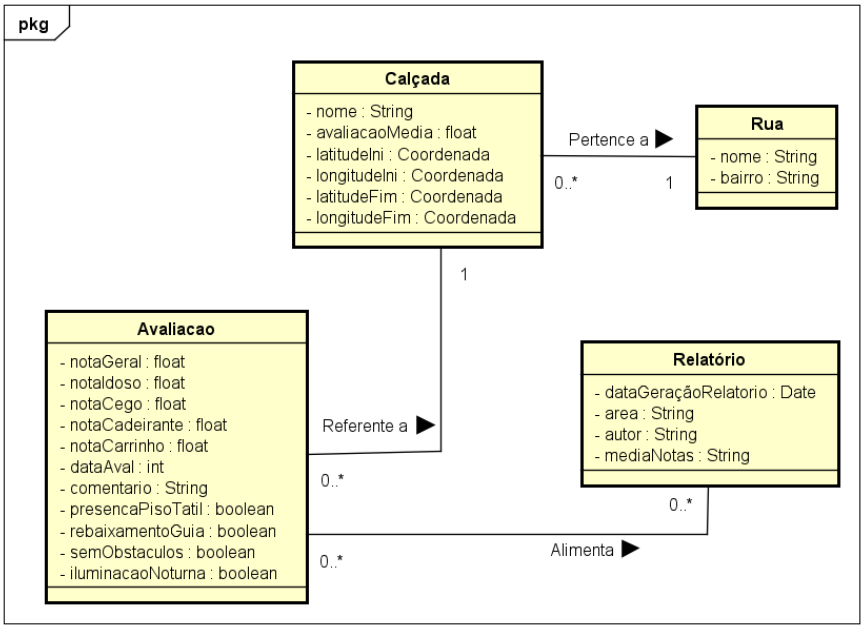

Figura 10: Diagrama de Classe de Domínio.

5.4 Diagramas de Sequência

5.4.1 Caso de uso 1: Consultar calçadas

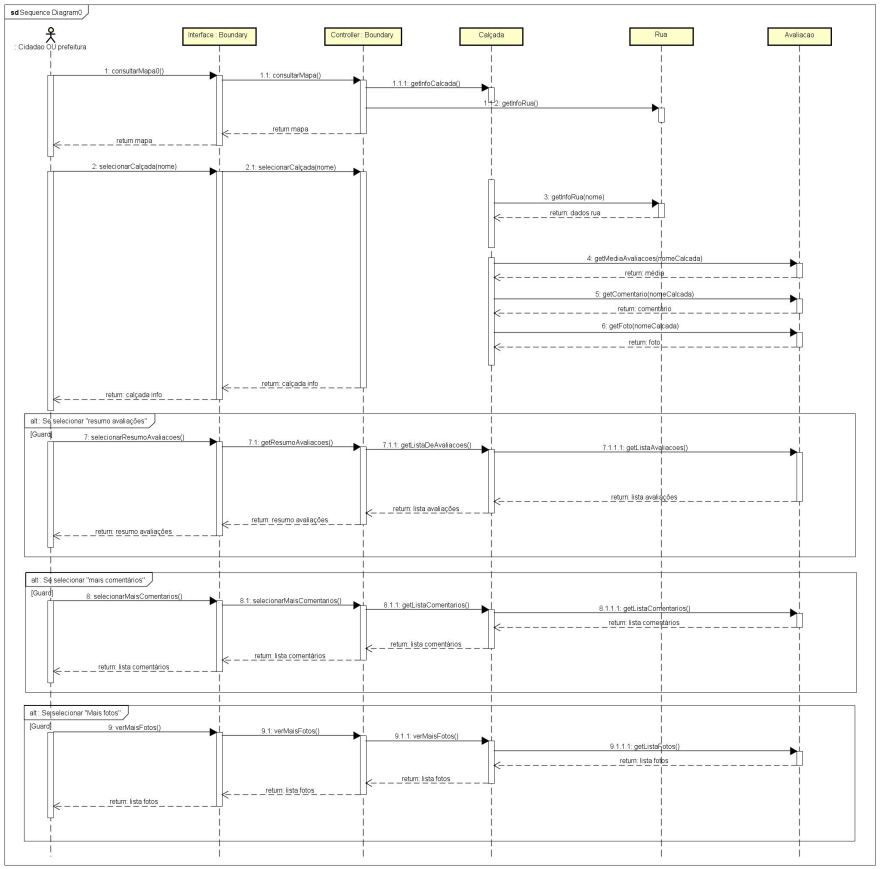

Figura 11: Diagrama de sequência: Consultar calçadas.

5.4.2 Caso de uso 2: Avaliar calçada

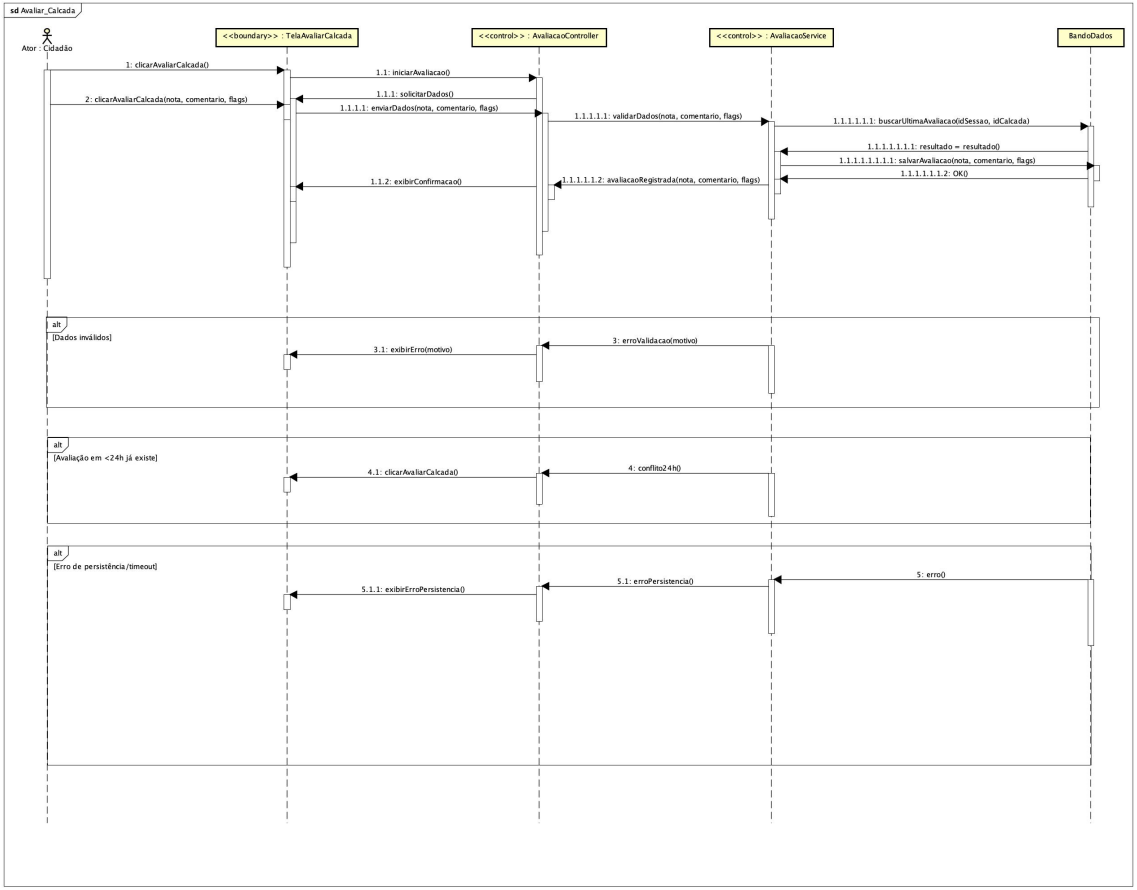

Figura 12: Diagrama de sequência: Avaliar calçada.

5.4.3 Caso de uso 3: Gerar relatório

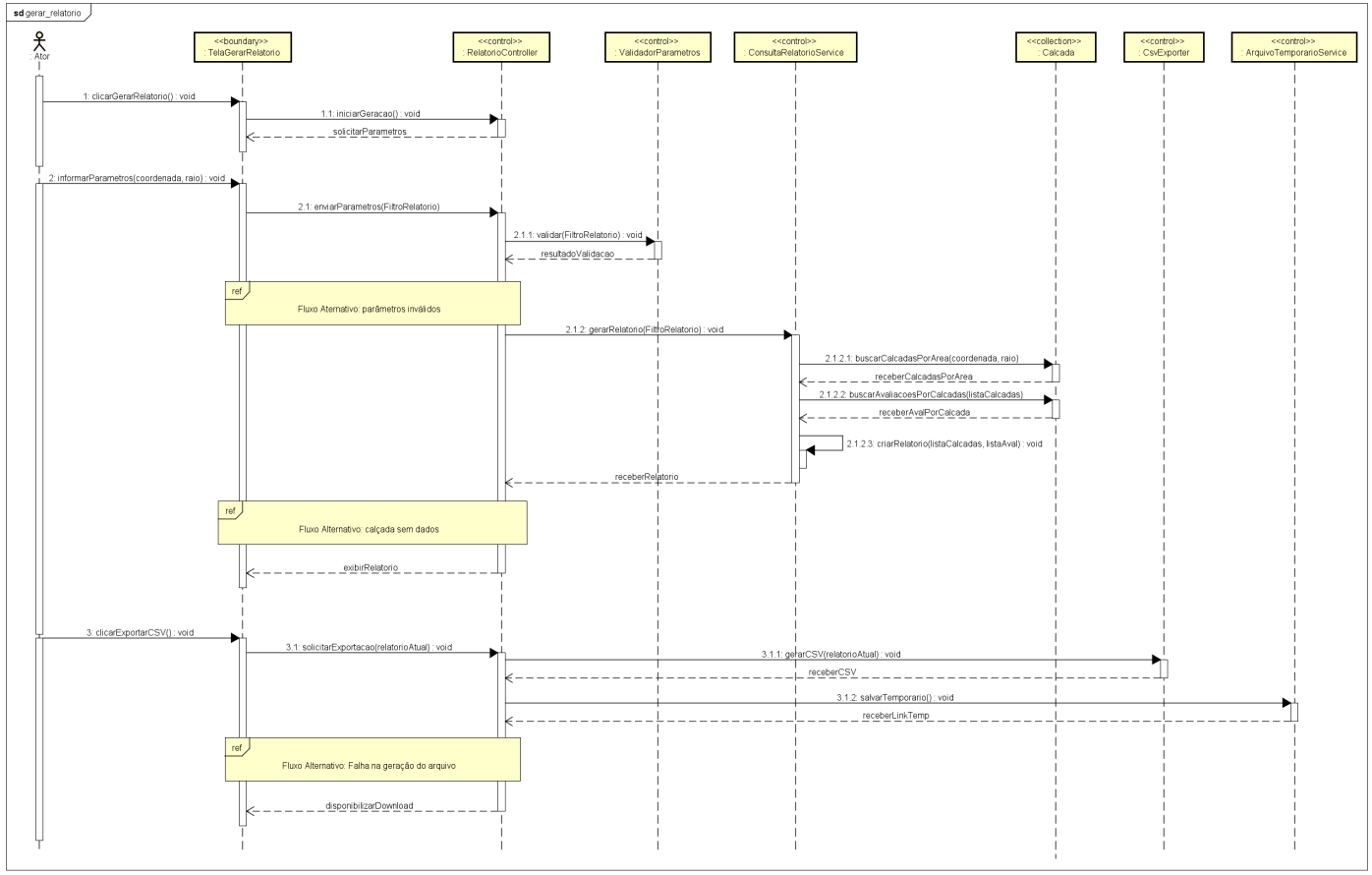

Figura 13: Diagrama de sequência: Gerar relatório.

6 Arquitetura do Sistema

6.1 Visão Geral da Arquitetura

A arquitetura do *Caminho Fácil* segue o estilo cliente-servidor em camadas, organizada para ser simples de implementar. O sistema é composto por:

- **Frontend**: HTML + CSS + Bootstrap para responsividade, com Leaflet.js para visualização de mapas.
- **Backend**: Spring Boot com API REST, estruturado em camadas (Controllers, Services, Repositories).
- **Banco de Dados**: PostgreSQL, acessado via Spring Data JPA.
- **APIs externas**: OpenStreetMap para tiles de mapa e Nominatim (ou Overpass) para geocodificação e recuperação de ruas.
- **Infraestrutura**: Deploy em containers Docker (app + banco), pipeline CI/CD no GitHub Actions e hospedagem em nuvem (AWS).

6.1.1 Diagrama geral

```
[Navegador]
    HTML + Bootstrap + Leaflet.js
         (requisições HTTP)
     [API REST - Spring Boot]
         (Spring Data JPA)
     [Banco de Dados PostgreSQL]

  [APIs Externas: Nominatim / OpenStreetMap]
```

6.2 Camadas do Sistema

Camada	Tecnologias	Responsabilidade principal
Frontend	HTML, CSS, Bootstrap, JS, Leaflet.js	Exibir mapa, interface responsiva, interação com usuários, envio/recebimento de dados
Backend	Java + Spring Boot (MVC)	Expor APIs REST, aplicar regras de negócio, validar dados, orquestrar persistência
Persistência	PostgreSQL + Spring Data JPA	Armazenar ruas, usuários e avaliações; consultas eficientes via repositórios
APIs externas	OpenStreetMap + Nominatim	Buscar ruas por nome, obter trechos em GeoJSON, usar mapas e geocodificação gratuita
Infraestrutura	Docker, GitHub Actions, AWS	Padronizar ambientes, automação CI/CD, deploy em nuvem

6.3 Fluxos por Funcionalidade

6.3.1 1. Ver ruas no mapa (carregamento inicial)

Objetivo: exibir ruas próximas à posição do usuário.

1. Frontend consulta posição atual via `navigator.geolocation`.
2. Requisição à API Nominatim/Overpass retorna ruas em formato GeoJSON.
3. Frontend desenha ruas no mapa com Leaflet.js.
4. Dados podem vir direto da API externa ou via backend (Adapter).

6.3.2 2. Avaliar uma rua

Objetivo: registrar avaliação de acessibilidade.

1. Usuário clica em uma rua no mapa.
2. Frontend exibe formulário (nota 1–5, checkboxes de acessibilidade, comentário opcional).
3. Frontend envia `POST /avaliacoes`.

4. Backend valida dados, verifica restrições (ex.: 1 avaliação por usuário/24h).
5. Backend persiste no banco. Se a rua não existir, insere novo registro com ID OSM.
6. Backend retorna confirmação, frontend atualiza mapa.

6.3.3 3. Consultar avaliações

Objetivo: exibir avaliações existentes.

1. Frontend envia `GET /avaliacoes?bbox=minLat,minLon,maxLat,maxLon`.
2. Backend consulta banco filtrando pelo retângulo geográfico.
3. Retorno inclui nome da rua, ID OSM, coordenadas, nota média, métricas agregadas.
4. Frontend desenha ícones coloridos no mapa.

6.3.4 4. Gerar relatório por região

Objetivo: consolidar métricas em raio de busca.

1. Usuário seleciona ponto no mapa e raio (ex.: 500m).
2. Frontend envia `GET /relatorio?lat=... & lon=... & raio=...`.
3. Backend busca avaliações próximas (fórmula de Haversine).
4. Calcula médias e estatísticas (ex.: % de ruas com piso tátil).
5. Retorna JSON ou CSV.
6. Frontend exibe relatório em tabela ou gráfico.

6.3.5 5. Buscar rua pelo nome

Objetivo: localizar rapidamente uma rua.

1. Usuário digita o nome.
2. Frontend consulta Nominatim.
3. Resposta contém coordenadas da rua.
4. Mapa centraliza na rua e exibe dados.

6.4 Padrões Utilizados

6.4.1 Arquiteturais

- **MVC em camadas**: separação entre interface (Frontend), lógica (Controllers/Services) e modelo (Banco).
- **RESTful APIs**: comunicação simples e padronizada.
- **Cliente-servidor desacoplado**: front independente do back, permitindo evolução separada.

6.4.2 De Projeto

- **DTOs (Data Transfer Objects)**: envio de dados enxutos para o front.
- **DAO/Repositórios com JPA**: consultas declarativas sem SQL manual.
- **Adapter**: encapsular chamadas externas (Nominatim/OSM).
- **Builder**: geração de relatórios customizados.

6.5 Justificativas de Escolha

Item	**Escolha feita**	**Justificativa**
Backend	Spring Boot + MVC	Framework robusto, usado no mercado, favorece boas práticas e didática
Banco de dados	PostgreSQL	Gratuito, confiável, compatível com dados geográficos
Frontend	HTML + Bootstrap + Leaflet.js	Simples, responsivo, rápido de implementar e ideal para mapas
APIs externas	OpenStreetMap + Nominatim	Gratuitas, bem documentadas, suporte a geocodificação
Padrão arquitetural	MVC + REST	Clareza, modularidade e facilidade de testes
Organização do código back	Controllers, Services, Repositories	Favorece divisão de responsabilidades e manutenção futura
Infraestrutura	Docker + GitHub Actions + AWS	Facilita CI/CD, padroniza ambiente, garante deploy acessível em nuvem