

**INSTITUTO FEDERAL GOIANO – CÂMPUS CERES
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

GIOVANE ALVARES

Desenvolvimento de um aplicativo multiplataforma para gestão de demandas de infraestrutura da Prefeitura de Ceres

CERES - GO
2025

GIOVANE ALVARES

Desenvolvimento de um aplicativo multiplataforma para gestão de demandas de infraestrutura da Prefeitura de Ceres

Trabalho de curso apresentado ao curso de Sistemas de Informação do Instituto Federal Goiano – Campus Ceres, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rafael Divino Ferreira Feitosa.

**Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema Integrado de Bibliotecas do IF Goiano - SIBi**

Alvares, Giovane
A473d Desenvolvimento de um aplicativo multiplataforma para gestão
de demandas de infraestrutura da Prefeitura de Ceres / Giovane
Alvares. Ceres 2025.
50f. il.
Orientador: Prof. Dr. Rafael Divino Ferreira Feitosa.
Tcc (Bacharel) - Instituto Federal Goiano, curso de 0320203 -
Bacharelado em Sistemas de Informação - Ceres (Campus
Ceres).
I. Título.

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem resarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- Tese (doutorado)
 Dissertação (mestrado)
 Monografia (especialização)
 TCC (graduação)
 Artigo científico
 Capítulo de livro
 Livro
 Trabalho apresentado em evento

Produto técnico e educacional - Tipo: **Aplicativos de aquisição e análise de dados**

Nome completo do autor:

Giovane Alvares

Matrícula:

2018103202030100

Título do trabalho:

Desenvolvimento de um aplicativo multiplataforma para gestão de demandas de infraestrutura da Prefeitura de Ceres

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: Não Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: **12 / 04 / 2025**

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Ceres

Local

12 / 04 / 2025

Data

Documento assinado digitalmente



GIOVANE ALVARES
Data: 04/12/2025 19:25:07-0300
Verifique em <https://validar.iti.gov.br>

Ciente e de acordo:

Assinatura do(a) orientador(a)



Documento assinado digitalmente
RAFAEL DIVINO FERREIRA FEITOSA
Data: 05/12/2025 22:12:33-0300
Verifique em <https://validar.iti.gov.br>



ATA DE DEFESA DE TRABALHO DE CURSO

Aos vinte e sete dias do mês de novembro do ano de dois mil e vinte e cinco, realizou-se a defesa de Trabalho de Curso do acadêmico Giovane Álvares, do Curso de Bacharelado em Sistemas de Informação, matrícula 2018103202030100, cujo título é “Desenvolvimento de um aplicativo multiplataforma para gestão de demandas de infraestrutura da Prefeitura de Ceres”. A defesa iniciou-se às 09 horas e 44 minutos, finalizando-se às 10 horas e 20 minutos. A banca examinadora considerou o trabalho APROVADO com média 9,2 no trabalho escrito, média 9,8 no trabalho oral, apresentando assim média aritmética final de 9,5 pontos, estando o estudante APTO para fins de conclusão do Trabalho de Curso.

Após atender às considerações da banca e respeitando o prazo disposto em calendário acadêmico, o estudante deverá fazer a submissão da versão corrigida em formato digital (.pdf) no Repositório Institucional do IF Goiano – RIIF, acompanhado do Termo Ciência e Autorização Eletrônico (TCAE), devidamente assinado pelo autor e orientador.

Os integrantes da banca examinadora assinam a presente.

(Assinado Eletronicamente)
Dr. Rafael Divino Ferreira Feitosa

(Assinado Eletronicamente)
Me. Rangel Rigo

(Assinado Eletronicamente)
Dr. Daniel Seabra

Documento assinado eletronicamente por:

- **Rafael Divino Ferreira Feitosa, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 27/11/2025 10:44:28.
- **Daniel Seabra, ANALISTA DE TEC DA INFORMACAO**, em 27/11/2025 10:47:51.
- **Rangel Rigo, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 28/11/2025 09:07:46.

Este documento foi emitido pelo SUAP em 26/11/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 766980
Código de Autenticação: c4d0c93160



RESUMO

A infraestrutura pública urbana demanda ciclos contínuos de detecção, priorização e resposta a ocorrências como buracos em vias, falhas de iluminação e drenagem insuficiente; contudo, a dispersão de canais e a baixa rastreabilidade dos registros comprometem a transparência e a efetividade do atendimento. Nesse contexto, este trabalho propôs a implementação de uma solução digital para gestão de demandas de infraestrutura, composta por aplicativo móvel e serviço web em Flutter com painel administrativo e API em Django, concebida para padronizar categorias, capturar evidências (foto, descrição e geolocalização) e permitir acompanhamento por estados de atendimento. O objetivo da pesquisa é centralizar solicitações e fornecer trilhas de auditoria que apoiem a tomada de decisão e a prestação de contas. A modelagem abrangeu casos de uso, máquina de estados da requisição e esquema relacional, assegurando coerência semântica entre entidades e endpoints. Como resultados, foram entregues de forma funcional o aplicativo do cidadão (cadastro/autenticação; abertura de solicitações com foto e geolocalização; histórico e consulta de status) e a API integrada e documentada; o painel do gestor permaneceu como plano de melhoria futura. Embora o escopo originalmente proposto não tenha sido integralmente atingido devido a restrições de tempo, redução de equipe e troca de tecnologias, as evidências de uso confirmam a viabilidade técnica e institucional do canal.

Palavras-chave: governo digital; infraestrutura urbana; participação cidadã

ABSTRACT

Urban public infrastructure requires continuous cycles of detection, prioritization, and response to incidents such as potholes, lighting failures, and insufficient drainage; however, dispersed service channels and low record traceability undermine transparency and the effectiveness of service delivery. In this context, this study proposed implementing a digital solution for managing infrastructure requests, consisting of a mobile application and a web service in Flutter, together with an administrative panel and a Django-based API, designed to standardize categories, capture evidence (photo, description, and geolocation), and enable tracking through service states. The objective of the research is to centralize requests and provide audit trails that support decision-making and accountability. The modeling encompassed use cases, a request state machine, and a relational schema, ensuring semantic consistency between entities and endpoints. As outcomes, a functional citizen application (registration/authentication; submission of requests with photo and geolocation; history and status tracking) and an integrated, documented API were delivered, while the manager panel remains a target for future improvement. Although the originally proposed scope was not fully achieved due to time constraints, team reduction, and technology changes, usage evidence confirms the channel's technical and institutional feasibility.

Keywords: digital government; urban infrastructure; citizen participation

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 - Diagrama de caso de uso do aplicativo móvel..... | 18 |
| Figura 2 - Modelo de Entidade Relacionamento da API..... | 21 |
| Figura 3 - Diagrama de estados da requisição..... | 23 |
| Figura 4 - Corpo JSON da requisicao POST /request/create/..... | 26 |
| Figura 5 - Resposta JSON da requisicao POST /request/create/..... | 26 |
| Figura 6 - Resposta JSON da requisicao GET /request/<id>/..... | 27 |
| Figura 7 - Resposta JSON da requisicao GET /request/<id>/status/..... | 27 |
| Figura 8 - Diagrama de fluxo de telas do aplicativo..... | 28 |
| Figura 9 - Tela de splash..... | 29 |
| Figura 10 - Tela de login..... | 29 |
| Figura 11 - Tela de cadastro..... | 30 |
| Figura 12 - Termos de uso..... | 31 |
| Figura 13 - Exemplo de e-mail de confirmação enviado..... | 32 |
| Figura 14 - Recupera senha..... | 32 |
| Figura 15 - Exemplo de e-mail enviado para redefinição de senha..... | 32 |
| Figura 16 - Tela principal..... | 33 |
| Figura 17 - Menu de perfil..... | 33 |
| Figura 18 - Menu de perfil..... | 34 |
| Figura 19 - Alterar dados..... | 35 |
| Figura 20 - Menu de solicitar..... | 35 |
| Figura 21 - Tipos de serviços para atendimento (Versão 2.0)..... | 36 |
| Figura 22 - Cadastro de solicitações (Versão 2.0)..... | 37 |
| Figura 23 - Minhas solicitações (Versão 2.0)..... | 38 |
| Figura 24 - Detalhes da solicitação (Versão 2.0)..... | 38 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 - Documentação dos casos de uso..... | 20 |
| Tabela 2 - Rastreabilidade entre casos de uso e rotas da API..... | 21 |
| Tabela 3 - Dicionário de dados das entidades da API..... | 24 |
| Tabela 4 - Rotas de usuário implementadas..... | 26 |
| Tabela 5 - Rotas de solicitação implementadas..... | 26 |

LISTA DE ABREVIATURAS E SIGLAS

TICs: Tecnologias da informação e comunicação

API: Application Programming Interface

MVP: Minimum Viable Product

HTTP: Hypertext Transfer Protocol

REST: Representational State Transfer

MTV: Model-Template-View

ORM: Object-Relational Mapper

JWT: JSON Web Token

MER: Modelo Entidade-Relacionamento

IP: Internet Protocol

LGPD: Lei geral de proteção de dados pessoais

JSON: JavaScript Object Notation

SUMÁRIO

| | |
|--|-----------|
| 1. Introdução..... | 8 |
| 2. Fundamentação Teórica..... | 11 |
| 3. Metodologia de Execução..... | 15 |
| 4. Resultados..... | 20 |
| 5. Conclusão..... | 44 |
| 6. Referencias..... | 46 |

1. Introdução

Nas instituições públicas, processos burocráticos baseados em fluxos manuais e despadronizados tendem a comprometer a eficiência da comunicação interna e do atendimento à população (BRASIL, 2020). Esse cenário impacta a qualidade dos serviços, alonga prazos e dificulta o acesso a informações confiáveis. A ausência de um suporte digital integrado também dificulta o acompanhamento das demandas entre setores, reduzindo a agilidade e a efetividade no uso de recursos públicos. À luz da agenda brasileira de governo digital — que prioriza simplificação, integração e oferta de serviços digitais mais simples, acessíveis e a menor custo — a transformação digital apresenta-se como caminho para reduzir prazos e elevar a qualidade do atendimento (BRASIL, 2020). Dessa forma, torna-se necessário desenvolver uma solução que centralize e organize as demandas de infraestrutura, melhorando o fluxo de informações e fortalecendo a comunicação institucional.

A transformação digital na gestão pública tem se tornado uma prioridade global, visando modernizar processos e aprimorar a prestação de serviços, conforme observado por Purcino, Gomes e Ribeiro (2024). Utilizar métodos tradicionais, tais como documentos físicos, planilhas não integradas e e-mails não padronizados, abrem brechas para erros, perda de informação e duplicidade de tarefas. Sant'ana (2009) destaca que a utilização de recursos de Tecnologia da Informação pode propiciar um ambiente mais transparente e acessível, permitindo um olhar da sociedade sobre as ações e decisões da gestão pública. Além disso, a falta de mecanismos eficientes para acompanhar o desempenho institucional limita a capacidade de planejamento e avaliação estratégica dos serviços públicos. Nesse contexto, torna-se fundamental a implantação de tecnologias que padronizam e promovam a rastreabilidade e automação dos fluxos das demandas de infraestrutura requisitadas pelos cidadãos, assim tornando a gestão mais eficiente.

Nas cidades brasileiras, é comum que cidadãos enfrentem dificuldades ao tentar comunicar problemas relacionados à infraestrutura urbana, como lâmpadas de postes apagadas, buracos em vias públicas, entupimentos de bueiros e demais falhas nos serviços básicos. Tais tipos de solicitações geralmente são feitas por meio de ligações telefônicas, e-mail ou atendimentos presenciais, o que dificulta o registro sistematizado, o acompanhamento do andamento e a efetiva resolução dos

problemas. Tal realidade evidencia uma lacuna na gestão pública, que ainda carece de ferramentas digitais eficientes para gerenciar e responder de forma ágil e transparente as demandas da população. Janssen e Estevez (2013) evidenciam que a adoção de plataformas digitais na administração pública permite a otimização dos processos, reduzindo desperdícios e aumentando a capacidade de resposta às necessidades sociais. Adicionalmente, Repette et al. (2021) destaca o uso de plataformas digitais como um canal de pressão popular para tornar a administração pública mais colaborativa, eficiente e responsiva.

Marson, Mayer e Nogueira (2013) apontam que a comunicação interna ineficiente nas instituições públicas prejudica diretamente o fluxo de informações e compromete a eficiência dos serviços prestados. Dentre os tipos de comunicação, destacamos as demandas de cidadãos por reparos em postes de iluminação danificados, bueiros entupidos e buracos nas vias públicas. Tal fragilidade comunicacional gera atrasos no atendimento e dificulta o acesso do cidadão às informações relevantes. Souza (2023) reforça essa ideia ao apontar que a ausência de um sistema estruturado de comunicação nas instituições públicas afeta negativamente a gestão interna e o relacionamento com o público externo. Além disso, Martins e Alves (2023) evidenciam que a adoção de plataformas digitais integradas na gestão pública pode aumentar a eficiência organizacional e a transparência, gerando impactos positivos na execução de políticas públicas.

Ao unir recursos que permitem ao cidadão registrar e acompanhar solicitações de infraestrutura urbana, o aplicativo atua como um canal direto de comunicação entre o cidadão e a administração pública. Essa transparência no fluxo comunicacional reforça a confiança na população nas instituições além de estimular a participação social. Segundo Bertot, Jaeger e Grimes (2010), às tecnologias da informação e comunicação (TICs), quando bem implementadas, podem promover a transparência, fomentar o engajamento cívico e servir como ferramentas eficazes de responsabilização governamental. Nesse sentido, a solução digital desenvolvida busca não apenas organizar internamente os processos administrativos, mas também oferecer um meio acessível e eficiente para que a população interaja com o poder público de forma ativa.

Diante desse cenário, foi desenvolvido um aplicativo para fornecer suporte à Prefeitura do Município de Ceres no gerenciamento de demandas de infraestrutura dos cidadãos. O objetivo é centralizar as solicitações, padronizando os fluxos de

trabalho e automatizando processos. Por meio do aplicativo o cidadão pode solicitar reparos e serviços e infraestrutura pública como conserto de semáforos, desentupimento de bueiros e reparos nas vias públicas, podendo acompanhar o andamento de sua solicitação, tal fator contribui para uma gestão mais transparente e aproxima a administração pública do cidadão. Além disso, oferecer transparência e rastreabilidade das ações administrativas. Organizando e integrando os setores em uma única plataforma, visavam promover melhorias significativas na comunicação interna, otimizando a gestão de recursos e contribuindo para elevar a qualidade dos serviços prestados à população.

O presente trabalho teve como objetivo geral desenvolver uma solução em tecnologia capaz de centralizar e organizar as demandas de infraestrutura pública, com o intuito de melhorar a eficiência e qualidade dos serviços prestados ao cidadão. Para alcançar isso, foram definidos os seguintes objetivos específicos:

1. projetar um sistema para captar e registrar as demandas de infraestrutura enviadas pelos cidadãos;
2. assegurar acesso direto às funcionalidades, para ser utilizado por diversos tipos de usuários;
3. disponibilizar informações operacionais aos setores envolvidos, de modo a apoiar respostas rápidas e precisas.

2. Fundamentação Teórica

A transformação digital na administração pública tem sido apontada como um vetor fundamental para a modernização do setor público. Segundo Castells (2003), a sociedade em rede demanda novas formas de organização e resposta por parte das instituições, com forte apoio em tecnologias da informação. A digitalização permite o armazenamento centralizado, o rastreamento de processos e a geração de indicadores de desempenho, contribuindo significativamente para a eficiência da gestão. Di Marco e Terci (2022) analisam como os portais de transparência e acesso à informação facilitam o controle social, permitindo que os cidadãos acompanhem as ações da administração pública de forma mais eficaz.

Nos últimos anos, plataformas digitais de reporte de ocorrências passaram a canalizar observações cotidianas dos moradores para fluxos administrativos, convertendo-as em informações operacionais para gestão municipal (OFFENHUBER, 2015). Ao estruturarem os registros com categorias, geolocalização e estados de atendimento, essas soluções favorecem a triagem, a priorização e o monitoramento das respostas públicas. Como discute Offenhuber (2015), a organização dos dados e sua espacialização elevam a “legibilidade” da infraestrutura e tendem a encurtar o ciclo entre abertura e solução das demandas.

A adoção de plataformas digitais para a gestão de demandas urbanas também fortalece os princípios de participação cidadã e transparência pública. Quando os cidadãos podem registrar, acompanhar e receber retorno sobre solicitações em tempo real, o processo de prestação de contas se torna mais direto e visível, promovendo maior engajamento social. Conforme afirmam Janssen e Estevez (2013), sistemas de governo eletrônico não apenas aumentam a eficiência, mas também transformam a relação entre governo e sociedade, ao fomentar uma cultura de colaboração e corresponsabilidade.

Além disso, a centralização de dados sobre demandas recorrentes permite às prefeituras identificar padrões, priorizar investimentos e planejar ações preventivas com base em evidências concretas. Para Bertot, Jaeger e Grimes (2010), o uso estratégico de tecnologias da informação amplia a capacidade de resposta das instituições públicas e fortalece a governança democrática. Nesse contexto, o desenvolvimento de soluções digitais não se limita à automação de

processos, mas representa um instrumento fundamental para a modernização da gestão pública orientada ao cidadão.

Como desdobramento prático do movimento de centralização e uso de evidências, experiências com plataformas digitais de reporte georreferenciado de problemas urbanos ilustram o potencial dessa abordagem. Um exemplo é o FixMyStreet, uma plataforma online e de código aberto para reporte georreferenciado de problemas urbanos, concebida para que cidadãos registrem ocorrências e para que os governos locais acompanhem e respondam a esses chamados (MYSOCIETY, 2025).

Estudos com dados do FixMyStreet em Bruxelas mostram que a participação em plataformas de relato de problemas pode variar por perfil socioeconômico e território, o que exige métricas de equidade e decisões de design que mitiguem vieses de acesso (PAK; CHUA; MOERE, 2017). Complementarmente, em Kansas City (EUA), análise de registros identificou disparidades socioespaciais no comportamento de queixa com implicações para a justiça algorítmica em decisões orientadas por dados, reforçando a necessidade de desenho inclusivo e monitoramento de viés ao longo do ciclo do serviço (KONTOKOSTA; HONG, 2021).

Segundo Brito (2025), a transformação digital só produz valor público quando ancorada em indicadores comparáveis, orientados a resultados e passíveis de escrutínio social contínuo. Para serviços públicos digitais, isso implica mensurar desempenho e cobertura com métricas reproduutíveis, permitindo distinguir ganhos de eficiência de meras automações e iluminando assimetrias territoriais e de acesso. Ao enfatizar a avaliação sistemática como parte do próprio ciclo de política, Brito (2025) reforça que plataformas cívicas precisam de medidas mínimas de efetividade e de equidade para sustentarem legitimidade e aprendizado institucional.

No Brasil, estudos sobre o Colab (outra plataforma que permite ao cidadão reportar problemas na infraestrutura pública) e iniciativas locais indicam que plataformas digitais aproximam cidadão e prefeitura, ao mesmo tempo em que trazem dados estruturados para priorização e planejamento (CASTRO; PEREZ, 2023; OLIVEIRA, ET AL., 2021). Pesquisas e materiais técnicos sobre o 1746 Rio mostraram a institucionalização de um canal unificado de solicitações, com integração a rotinas internas e indicadores públicos, servindo de referência para outras capitais (SHIGENO, 2016).

No município de São Paulo, o SP156 é o canal oficial de atendimento ao cidadão — acessível por portal e aplicativo — que centraliza o registro e o acompanhamento de solicitações de serviços públicos. Além do atendimento, o programa disponibiliza dados abertos específicos sobre as solicitações registradas pelas populações, favorecendo a transparência e a reutilização de informações pelo poder público e pela sociedade (PREFEITURA DE SÃO PAULO, 2025).

A partir dessa base, estudos evidenciam como dados de solicitações alimentam o planejamento e a prestação de contas; esse uso deve estar ancorado em princípios de governo aberto (transparência, participação, colaboração) (QUINELLO; BONI, 2023). Esses achados convergem para três pilares: um canal único e padronizado para abertura e acompanhamento; dados estruturados e georreferenciados para gestão e indicadores; e feedback visível ao cidadão, essencial para confiança e participação contínua.

O canal móvel é particularmente relevante. Ao estar no bolso do cidadão, reduz barreiras de acesso ao serviço e permite registrar uma demanda “no local”, com foto e localização. Relatório de referência mostra ganhos de capilaridade, engajamento e eficiência quando aplicativos móveis são integrados aos fluxos internos de triagem e atendimento (GANPATI, 2015). Esses resultados dialogam com achados de meta estudos sobre participação digital: plataformas de “voz cidadã” tendem a gerar maior responsividade quando há integração institucional e devolutivas claras (NALCHIGAR; FOX, 2014).

Para que esses ganhos se sustentem além de pilotos, é necessário um arranjo arquitetural que garanta padronização, interoperabilidade e governança de dados entre canais (app do cidadão, painéis do gestor) e sistemas já existentes. Na literatura de setor público, a arquitetura corporativa funciona como camada de coordenação que organiza processos, dados e serviços, reduz redundâncias e viabiliza evolução incremental das soluções (VILELA-CURY et al., 2023).

Em governos, apps e serviços móveis apoiados em interfaces de programação de aplicações (APIs) claras reduzem custos transacionais e ampliam o engajamento, ao mesmo tempo em que viabilizam ecossistemas de inovação pública (GANPATI, 2015). A literatura de cidade-como-plataforma reforça que interoperabilidade e governança de dados são condições para consolidar serviços urbanos em camadas reutilizáveis, com ganhos de coordenação entre secretarias (REPETTE ET AL., 2021).

Segundo Vilela-Cury et al. (2023), a adoção de arquiteturas e padrões no setor público, com destaque para referenciais como a ePING, é habilitadora da integração entre sistemas, reduzindo redundâncias e possibilitando evolução coordenada de serviços. Vilela-Cury et al. (2023) ainda evidencia que padronização semântica, governança de dados e contratos bem definidos sustentam interoperabilidade, rastreabilidade e prestação de contas. Em plataformas que articulam múltiplos órgãos, essa base arquitetural é condição para comparabilidade de informações, confiabilidade das séries históricas e compartilhamento seguro de dados entre módulos e entidades governamentais.

Além do tipo, canal e arquitetura da solução, também é necessário se atentar à acessibilidade. Alencar et al. (2023), pontua que a experiência do usuário em serviços digitais depende de percursos comprehensíveis, feedback de estado e linguagem clara, elementos que afetam diretamente eficácia, eficiência e satisfação em tarefas concretas. Apontando também que a usabilidade não é atributo cosmético, mas condição para compreensão e realização de objetivos pelo cidadão, especialmente em contextos de diversa alfabetização digital. Em plataformas públicas, essa perspectiva reforça a centralidade de acessibilidade e legibilidade como requisitos de valor público, prevenindo barreiras de uso e incentivando a participação informada.

Como solução pública que lida com imagens, geolocalização e metadados de atendimento, o sistema deve observar a Lei Geral de Proteção de Dados Pessoais (LGPD): base legal para tratamento pelo poder público, minimização de dados, transparência ao titular e trilhas de auditoria para responsabilização (BRASIL, 2018). Isso se traduz, na prática, em consentimento/informação clara no app, políticas de retenção e controle de acesso por perfis no módulo gestor.

Segundo Brelàz, Crantschaninov e Bellix (2021), a institucionalização de governo aberto em nível municipal demanda mecanismos efetivos de transparência ativa e devolutiva ao cidadão, sob pena de plataformas participativas se limitarem a coletar manifestações sem produzir responsabilização. A credibilidade de iniciativas digitais decorre de rotinas públicas de explicação, monitoramento e aprendizado coletivo, com divulgação inteligível de critérios e resultados. Assim, a lógica da “accountability”(assumir responsabilidade por ações, decisões e resultados, com transparência e comprometimento) informa o desenho de serviços digitais voltados à resolução de demandas urbanas e à coordenação intersetorial.

3. Metodologia de Execução

Este capítulo descreve a metodologia de execução do projeto, originado a partir de demanda da Secretaria Municipal de Planejamento e Tecnologia de Ceres, com o objetivo de desenvolver um aplicativo para registro e gestão de solicitações de infraestrutura urbana.

A partir desta demanda, organizou-se no IF Goiano – Campus Ceres uma ampla equipe dedicada ao desenvolvimento da solução, com o propósito de traduzir necessidades operacionais em requisitos técnicos verificáveis. Para assegurar aderência ao contexto municipal, adotou-se uma abordagem iterativa leve, com reuniões semanais que periodicamente contaram com a participação de representantes da Secretaria para pactuar prioridades, revisar entregar e ajustar o escopo conforme restrições de recursos, prazo e dados disponíveis. Essa dinâmica possibilitou converter solicitações do serviço público em incrementos funcionais, mantendo alinhamento contínuo entre expectativas institucionais e a evolução do projeto.

Assim, os integrantes foram estruturados em três frentes — design, documentação e desenvolvimento — tais grupos constituíam-se de discentes, que faziam as atividades, e de docentes, que supervisionavam e ajudavam nas execuções das atividades. O autor do presente trabalho integrou a equipe de desenvolvimento, composta inicialmente por dois integrantes e, em etapa posterior, apenas pelo próprio autor. A equipe ficou responsável pela implementação do aplicativo em Flutter e da API em Django, cujo código foi desenvolvido quase integralmente pelo autor, com contribuições pontuais do outro integrante. Fazendo o uso das reuniões semanais também como pontos de checagem para definir prioridades, remover impedimentos e registrar decisões técnicas. Periodicamente também haviam reuniões com o representante da prefeitura para atualizar sobre o andamento do projeto e possíveis alterações do escopo.

A equipe de design ficou responsável por desenhar as telas de forma clara e intuitiva. A de documentação ficou responsável por documentar as telas, botões, rotas de API e seus respectivos funcionamentos. Por fim, a equipe de desenvolvimento tinha como objetivo juntar o design definido com a documentação imposta e desenvolver a aplicação móvel.

Para o desenvolvimento da aplicação adotou-se uma abordagem iterativa com ciclos de metas semanais alinhadas às restrições de tempo e experiência da equipe. Reuniões e pequenas entregas semanais se justificaram devido ao curto tempo disponível para a execução da implementação do aplicativo, dessa forma, pode-se ajustar o escopo e definir qual seria o melhor caminho a se tomar de acordo com a necessidade atual da equipe e do cronograma.

Havia também a equipe de testes, que seria formada por integrantes das outras equipes, esse grupo seria formado ao final do desenvolvimento do aplicativo para que as funções pudessem ser validadas garantindo a qualidade do produto final. Como o software não foi finalizado essa equipe não foi formada, os testes das funcionalidades foram realizados conforme eram desenvolvidos pela equipe de desenvolvimento.

Inicialmente o escopo do projeto era dividido em três partes: o aplicativo cidadão, o painel do gestor e o aplicativo das equipes municipais. Essas partes serão descritas nos próximos parágrafos.

A primeira parte consistiria-se do aplicativo móvel voltado à participação do cidadão, que facilitaria o registro de demandas relacionadas à infraestrutura pública como buracos em vias, falhas na iluminação, entupimentos de bueiros e outros problemas urbanos. A solução permitiria ao usuário informar a localização exata do ocorrido por meio de geolocalização, adicionar fotografias que evidenciam a situação e descrever o problema de forma detalhada, contribuindo para a triagem e priorização pelas equipes responsáveis. Adicionalmente, o usuário ainda poderia acompanhar o andamento da requisição e receber notificações sobre atualizações.

Como segunda parte, o painel do gestor, onde o responsável pela infraestrutura municipal poderia acompanhar as solicitações que estariam sendo enviadas, definir nível de prioridade do problema e alocar uma equipe para resolvê-las. O painel de gestão ainda contaria com gráficos contendo métricas relacionadas às requisições (tempo médio de resolução, quantidade de requisições por bairro, quantidade de requisições de cada tipo e etc.)

Por último, teria-se o aplicativo das equipes responsáveis por resolver as requisições. Sendo também uma aplicação móvel, a equipe de serviços da prefeitura receberia pelo celular as requisições em que deviam trabalhar, contendo os dados básicos para a execução da atividade como localização, imagens do problema e a descrição da natureza da requisição.

Devido a limitações e tempo, equipe e a complexidade do escopo inicial, mais a frente foi definido um Produto Mínimo Viável (MVP) para que se pudesse realizar uma entrega funcional. Dessa forma manteve-se o aplicativo do cidadão e o painel do gestor, removendo o aplicativo das equipes do escopo. Com esse MVP a prefeitura ainda poderia realizar a implantação do aplicativo internamente e avaliar seu desempenho e funcionalidade pelo painel do gestor.

Nas primeiras reuniões do primeiro mês foram definidos os papéis de cada integrante, a equipe de design definiu a utilização do Material Design (conjunto de diretrizes, componentes e ferramentas desenvolvido pelo Google) para o desenho das telas, assim como o logo e cores da aplicação. A equipe de documentação estudou o escopo e iniciou a documentação das funções dos botões e telas. Enquanto a equipe de desenvolvimento definiu a coleção de tecnologias, frameworks e componentes de infraestrutura.

Para o front-end definiu-se o uso de Flutter que tem como linguagem de programação o Dart, para o back-end Firebase e para o controle de versionamento do projeto o GitHub. Ainda no primeiro mês foram iniciados os estudos em flutter para possibilitar o desenvolvimento da solução.

O Flutter ganhou destaque por permitir a criação de aplicativos nativos para Android e iOS a partir de um único código-fonte, isso reduz o esforço de manutenção e facilita a integração com APIs HTTP/REST, notificações e serviços de back-end (Payne, 2019). Além disso, o Flutter organiza a interface por meio de uma árvore de widgets e um ciclo declarativo de atualização, o que simplifica a composição de telas e a gestão de estado. Em conjunto com recursos como hot reload e bibliotecas de componentes nativos (Material/Cupertino), essa abordagem encurta o ciclo de prototipagem e favorece a consistência visual entre plataformas (Payne, 2019).

No segundo mês com a finalização da modelagem do design das telas, iniciou-se o desenvolvimento do front-end, usando como base as telas desenhadas pelo time de design. Neste primeiro momento construiu-se uma demo do front-end com as telas em Flutter apenas de maneira visual e com a navegação proposta até o momento, ainda sem persistência de dados.

Seguindo para o terceiro mês trocou-se o Firebase pelo framework Django para o desenvolvimento do back-end. Naquele momento, a troca se justificou pois Firebase ainda era uma ferramenta nova e com pouca confiabilidade para

manutenção futura pois não se dispunha de ninguém com experiência nele, em contrapartida o Django já era bem consolidado e com muita documentação e padrões de segurança já inclusos em sua arquitetura e por ser utilizado pela reitoria, haviam pessoas a quem poderia-se pedir ajuda. Ao final do terceiro mês, após finalizar o desenvolvimento do core do front-end, se iniciam os estudos e paralelamente o desenvolvimento da API em Django. Foi neste momento que se calculou o cronograma e decidiu-se estabelecer o MVP citado anteriormente.

A utilização do Django, framework web baseado em Python, oferece robustez, segurança e agilidade no desenvolvimento, além de integração nativa com bancos de dados relacionais. Como framework web de alto nível, o Django adota o padrão Model-Template-View (MTV), o que favorece a separação de responsabilidades e a estabilidade do acoplamento entre módulos. Essa organização, somada a componentes já incluídos como Mapeador de Objeto-Relacional (ORM), roteamento e painel administrativo, acelera o desenvolvimento de aplicações com requisitos típicos de gestão de dados (LI; SUN, 2022).

No mês quatro, continuou-se o desenvolvimento da API, aprimorando as models, rotas e incluindo validações dos dados enviados no request. Também foi dado prosseguimento no aprimoramento da aplicação, realizando-se ,melhorias de desempenho nas telas já existentes e foram corrigidos erros encontrados. Ao final do quarto mês inicia-se a integração das duas partes com a persistência de dados e autenticação de usuário. Inicialmente tentou-se implementar autenticação por token JSON(JavaScript Object Notation) Web Tokens (JWT), mas devido ao cronograma apertado e aos erros encontrados, decidimos simplificar e utilizar o básico de autenticação com usuário e senha.

Ao quinto mês, continuava-se com ajustes no app, foram finalizadas as rotas de usuário e requisição, e iniciou-se o polimento do aplicativo e a correção de erros encontrados. Ao final do quinto mês se iniciou o desenvolvimento do painel do gestor web e foram implementados os tipos de usuário, confirmação de e-mail e troca de senha. Neste ponto, o aplicativo móvel do cidadão já estava totalmente funcional, contando com criação de conta, confirmação de email, restauração de senha, envio e acompanhamento das requisições com todas as informações necessárias.

Após isso, com o fim do semestre letivo, o projeto foi encerrado. Vários fatores colaboraram para o encerramento do projeto antes da finalização do desenvolvimento como desfalque da equipe, curto período para o desenvolvimento

do escopo e curva de aprendizado do back-end definido posteriormente. Tais dificuldades serão discutidas mais à frente no próximo capítulo.

4. Resultados

Este capítulo consolida a documentação do que foi efetivamente desenvolvido durante a execução do projeto (aplicativo em Flutter e API em Django); mostra as telas do aplicativo que foram entregues no protótipo descrevendo sua funcionalidade, botões e fluxos entre elas; examina as dificuldades enfrentadas no desenvolvimento do aplicativo do cidadão e em sua integração com a API abrangendo a visão funcional dos casos de uso, modelo de dados da API, fluxo de estados da requisição, especificações das rotas da API e navegação entre as telas do aplicativo.

A execução iniciou-se com escopo amplo e múltiplas frentes (design, documentação, aplicativo do cidadão, API, painel do gestor e aplicativo das equipes). Na prática, porém, a capacidade foi subdimensionada: cada frente contou com apenas dois discentes e, ao longo do semestre, o volume de trabalho concentrou-se em um grupo reduzido. Esse descompasso entre ambição e disponibilidade de pessoas impactou o ritmo das entregas.

O acúmulo de fatores limitantes e dificuldades tornou evidente a inviabilidade de concluir o escopo integral dentro do semestre. Em resposta, definiu-se um MVP centrado no ciclo mínimo de valor: aplicativo do cidadão, API e painel do gestor para recebimento e análise das solicitações. Mesmo assim, a integração entre equipes apresentou entraves operacionais (dificuldade de agenda e baixa aderência a rituais), reduzindo a frequência de alinhamentos e atrasando decisões de detalhe.

Houve, ainda, uma decisão arquitetural que elevou o risco: a substituição do Firebase por Django no meio do ciclo. Embora tecnicamente vantajosa em segurança, controle e interoperabilidade, a troca introduziu uma curva de aprendizado significativa e atrasou a disponibilização da API por aproximadamente seis semanas. Considerando o prazo acadêmico, essa escolha reduziu a janela útil para integração com o aplicativo.

Com esses fatores, escopo funcional entregue compreende: o registro de solicitações com foto, descrição e localização; autenticação/registro do cidadão; listagem e detalhamentos das requisições, com histórico de status. Esses fluxos compõem o recorte mínimo para a abertura e acompanhamento de demandas de infraestrutura urbana no dispositivo móvel.

A modelagem utilizada neste capítulo é composta pelo diagrama de casos de uso (interações do ator cidadão com o sistema), Modelo de entidade-relacionamento (MER) da API, fluxo de estados das solicitações e fluxo de telas do aplicativo. As figuras e tabelas correspondentes a estes itens são analisadas posteriormente neste capítulo.

O diagrama de casos de uso modela, em alto nível, as interações entre atores externos e o sistema representando os objetivos que esses atores buscam alcançar e os limites do sistema. Sua importância reside em delimitar escopo funcional, comunicar requisitos de maneira acessível a públicos técnicos e não técnicos, apoiar a priorização de funcionalidade e orientar validação/testes por meio da derivação de cenários e da verificação de cobertura funcional ao longo do ciclo de desenvolvimento.

As interações representadas no diagrama sintetizam a fronteira entre o ator cidadão e o sistema, evidenciando os fluxos implementados de abrir solicitação, listar/consultar solicitações enviadas e cadastrar/ativar conta.

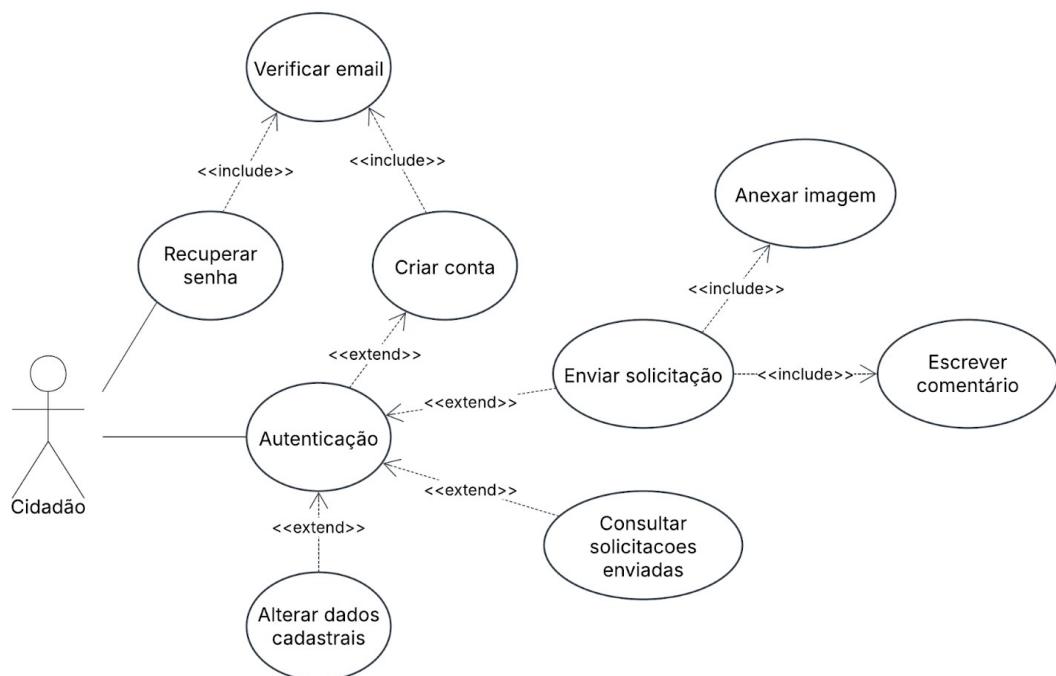


Figura 1 - Diagrama de caso de uso do aplicativo móvel

| ID | Caso | Autor | Objetivo | Entradas | Saída |
|-------|--------------------------|---------|---|---|--|
| UC-01 | Criar/ativar conta | Cidadão | Criar conta e concluir ativação via token | Dados cadastrais | Conta ativa e apta para autenticar |
| UC-02 | Recuperar senha | Cidadão | Redefinir senha via link por email | Email já cadastrado | Senha redefinida |
| UC-03 | Autenticar | Cidadão | Logar no sistema | CPF e senha | Acesso à página principal do app |
| UC-04 | Alterar dados cadastrais | Cidadão | Alterar informações do perfil do usuário | Dados que deseja alterar | Dados cadastrais atualizados |
| UC-05 | Enviar solicitação | Cidadão | Iniciar uma requisição | Imagen, localização e comentário | Solicitação criada com status inicial “em análise” |
| UC-06 | Anexar imagem | Cidadão | Anexar de 1 a 3 imagens necessárias para submeter a solicitação | Fotos capturadas no momento da abertura da requisição | Imagens anexadas a requisição |
| UC-07 | Escrever comentário | Cidadão | Escrever comentário na requisição | Descrição do problema encontrado | Comentário anexado a requisição |

Tabela 1 - Documentação dos casos de uso

A tabela a seguir mapeia cada caso de uso às rotas utilizadas para a realização dessas ações, todas as rotas apresentadas a seguir sucedem o prefixo da url da api “http://{IP_SERVIDOR}/{rota}”.

| UC | Método | Rota | Função |
|-------|--------|----------------------|---|
| UC-01 | POST | user/create/ | Criar o usuário do tipo “1 - cidadão” com estado inativo e envia email de confirmação para ativar a conta |
| UC-02 | POST | user/reset_password/ | Enviar email de redefinição de senha |
| UC-03 | GET | user/login/{cpf} | Validar as credenciais e retorna os dados do usuário caso exista |
| UC-04 | PUT | user/{id}/update | Atualizar informações do usuário |
| UC-05 | POST | request/create/ | Gravar uma nova requisição com status “1 - Em análise” |

Tabela 2 - Rastreabilidade entre casos de uso e rotas da API

A partir dos casos de uso, o Modelo Entidade-Relacionamento (MER) materializa o comportamento esperado em um vocabulário de dados estável: define entidades, atributos e relacionamentos com cardinalidades, fazendo a ponte entre a visão funcional e o esquema lógico que sustenta a API e as telas. Na prática, o MER orienta normalização, integridade referencial e domínios, alimenta o dicionário de dados e informa chaves/índices para consultas e auditoria. Com isso, cada fluxo mapeia de forma rastreável para operações persistentes, reduzindo ambiguidade e facilitando a evolução do protótipo.

O domínio da API desenvolvida organiza-se em torno das entidades Usuario, UserType, Request, RequestType, Status e StatusCode. Usuario (autor) possui um UserType e se Relaciona com Request que pertence a um RequestType e possui zero ou mais Status vinculados que possuem um StatusCode.

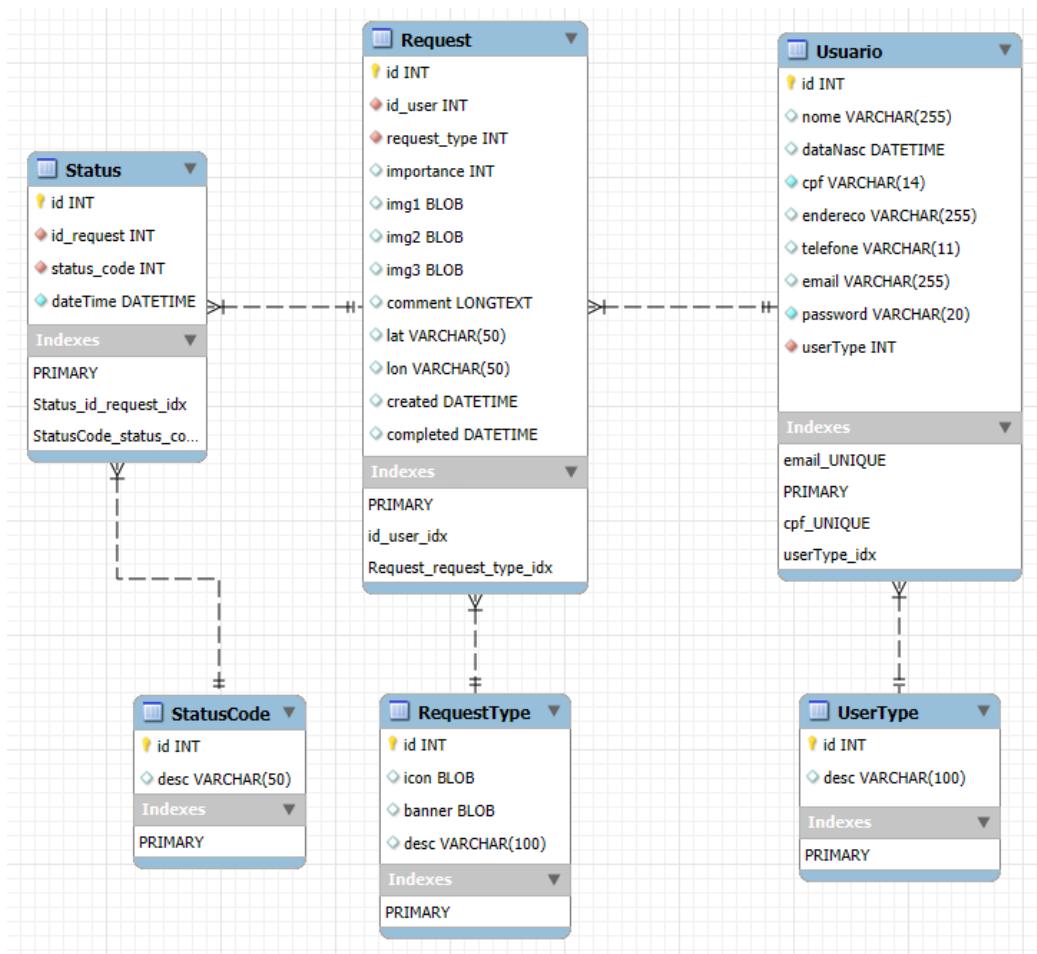


Figura 2 - Modelo de Entidade Relacionamento da API

A tabela a seguir descreve as entidades do banco de dados e seus atributos e funções.

| Entidade | Campo | Regras | Conteúdo |
|----------|----------|--------------------|----------------------------|
| Usuario | id | PK, Auto_increment | Chave identidade da tabela |
| | nome | | Nome do usuário |
| | dataNasc | | Data de nascimento |
| | cpf | Único | CPF |
| | endereco | | Endereço |
| | telefone | | Telefone |

| Entidade | Campo | Regras | Conteúdo |
|-------------|--------------|------------------------|---|
| Usuario | email | Único | E-mail |
| | password | | Senha criptografada |
| | userType | FK para UserType_id | Tipo do usuário |
| | date_joined | | Data e hora do cadastro |
| UserType | id | PK | Chave identidade da tabela |
| | desc | | Descrição |
| Request | id | PK, Auto_increment | Chave identidade da tabela |
| | id_user | FK para Usuario_id | Id do usuário que criou |
| | request_type | FK para RequestType_id | Tipo da solicitação |
| | importance | | Nível de prioridade |
| | img1 | | Fotos capturadas e enviadas no momento da criação da requisição |
| | img2 | | |
| | img3 | | |
| | comment | | Descrição da requisição |
| | lat | | Latitude da localização |
| | lon | | Longitude da localização |
| | created | | Data de criação |
| | completed | | Data da finalização |
| RequestType | id | PK | Chave identidade da tabela |
| | icon | | Base64 do ícone da solicitação |
| | banner | | Base64 do banner da solicitação |
| | desc | | Descrição do tipo solicitação |

| Entidade | Campo | Regras | Conteúdo |
|------------|-------------|-----------------------|-------------------------------------|
| Status | id | PK, Auto_increment | Chave identidade da tabela |
| | id_request | FK para Request_id | A qual requisição pertence o status |
| | status_code | FK para StatusCode_id | Código do status |
| | dateTime | | Data/hora do status |
| StatusCode | id | PK | Chave identidade da tabela |
| | desc | | Descrição |

Tabela 3 - Dicionário de dados das entidades da API

O MER e a tabela mostrados acima ilustram a estrutura de dados presentes na API desenvolvida. Entidades como UserType, RequestType e StatusCode assumem a função de tabela de referência, armazenando valores pré definidos para normalizar e controlar dados repetitivos. UserType armazena os tipos de usuário que podem ser: 1 - Cidadão, 2 - Gestor, 3 - Admin , 4 - Inativo, 5 - Bloqueado. Já a tabela StatusCode armazena os tipos de status possíveis para uma requisição que são: 1 - Em análise, 2 - Em andamento, 3 - Concluída, 4 - Rejeitada e 5 - Cancelada.

O ciclo de vida da solicitação é representado por uma máquina de estados, utilizada para validar transições e alimentar a tabela de Status. O diagrama explicita o caminho típico da abertura até a conclusão e os ramos de rejeição/cancelamento, garantindo previsibilidade no acompanhamento pelo cidadão.

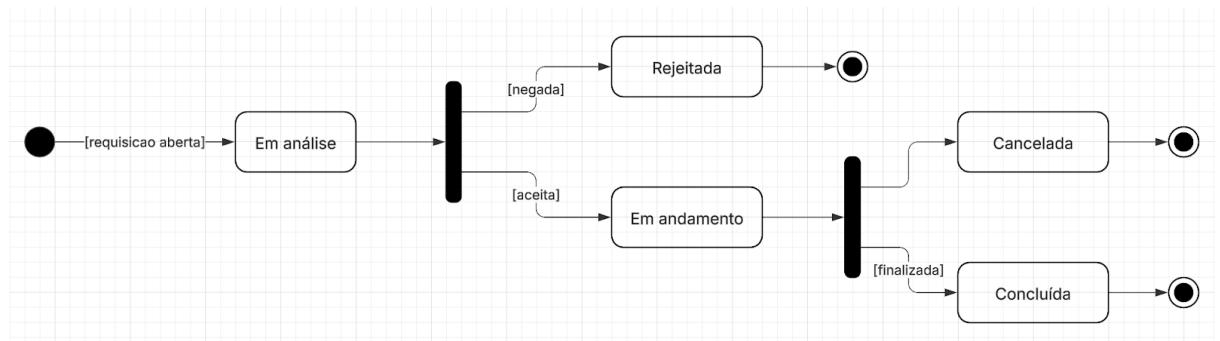


Figura 3 - Diagrama de estados da requisição

No desenvolvimento do aplicativo e da API, adotou-se o princípio da minimização de dados e definiu-se política de retenção compatível com a finalidade do tratamento. Embora a versão atual do app ainda não disponibilize um botão para exclusão de informações pessoais e desativação de conta, a API já prevê o procedimento: a rota “https://IP_SERVIDOR>/user/{id_usuario}/delete” executa a remoção lógica do titular, alterando o seu perfil para “4 – Inativo” e inativando o cadastro, preservando apenas metadados indispensáveis à comprovação de operações e segurança. Essa conformidade busca atender aos direitos do titular previstos na LGPD, sem comprometer a integridade operacional do sistema.

Os próximos parágrafos formalizaram os contratos de integração efetivamente disponíveis na API do protótipo.

As rotas estão organizadas por módulo de aplicação e publicadas no formato utilizado no projeto, a partir do endereço do servidor, que segue o seguinte padrão como base: “http://IP_SERVIDOR}/{modulo}/{rota}” .

O primeiro módulo é o de usuário que segue a url “http://ip_servidor>/user/{rota}”. Este módulo contém as rotas com todas as operações necessárias para cadastrar, atualizar, deletar e gerenciar usuários do sistema.

| Método | Rota | Função |
|--------|----------------------------|---------------------------------------|
| GET | / | Lista rotas do módulo |
| POST | /create/ | Criação de usuário |
| GET | /activate/<uidb64>/<token> | Ativação de conta via link/token |
| GET | /users/ | Listagem de usuários |
| GET | /login/<cpf> | Autenticação por credencial informada |
| PATCH | /<id>/change_password | Alteração de senha |
| PATCH | /<id>/update | Atualização do usuário |
| DELETE | /<id>/delete | Exclusão/desativação da conta |
| GET | /<id>/ | Dados de um usuário |

| Método | Rota | Função |
|--------|--------------------------|--|
| GET | /reset_password/ | Inicia o fluxo de redefinição de senha |
| GET | /reset_password_sent/ | Confirma o disparo de instruções de redefinição. |
| GET | /reset/<uidb64>/<token> | Validação de token de redefinição. |
| GET | reset_password_complete/ | Conclusão do fluxo de redefinição. |

Tabela 4 - Rotas de usuário implementadas

O segundo módulo é o de requisições que tem como base “http://{IP_SERVIDOR}/request/{rota}”. Este módulo contém todas as rotas necessárias para registrar, atualizar, desativar e acompanhar as requisições no sistema.

| Método | Rota | Função |
|--------|-------------------|---|
| GET | /reqs/ | Listagem geral de solicitações |
| POST | /create/ | Criação de nova solicitação |
| GET | /types/ | Catálogo de tipos de ocorrência |
| GET | /status/ | Consulta de status (visão de suporte) |
| GET | /status/user/<id> | Status das solicitações de um usuário específico. |
| GET | /status/code | Lista de códigos de status disponíveis (domínio). |
| POST | /create/status/ | Registro de evento de status (histórico). |
| PATCH | /<id>/update/ | Atualização da solicitação. |
| DELETE | /<id>/delete/ | Exclusão da solicitação. |
| GET | /<id>/ | Detalhe de uma solicitação. |
| GET | /<id>/status/ | Histórico de status da solicitação |
| GET | /<id>/user/ | Solicitações pertencentes ao usuário informado |

Tabela 5 - Rotas de solicitação implementadas

A seguir serão demonstrados alguns exemplos de consumo de rotas descritas anteriormente, sinalizando a rota, body em json enviado e o response recebido.

O Primeiro exemplo mostra o corpo em JSON do consumo da rota POST de registrar solicitacao “http://{IP_SERVIDOR}/request/create/” usando autenticacao via cabecalho, seguido pelo JSON do response com status HTTP 201Created.

```
{  
    "img1": "codigoBase64imagem1",  
    "img2": "codigoBase64imagem2",  
    "img3": "codigoBase64imagem3",  
    "comment": "Buraco na avenida, proximo a faixa de pedestres",  
    "lat": "-15.311300287282295",  
    "lon": "-49.603112939122674",  
    "id_user": 1,  
    "request_type": 1  
}
```

Figura 4 - Corpo JSON da requisicao POST /request/create/

```
{  
    "id": 2,  
    "importance": null,  
    "img1": "codigoBase64imagem1",  
    "img2": "codigoBase64imagem2",  
    "img3": "codigoBase64imagem3",  
    "comment": "Buraco na avenida, proximo a faixa de pedestres",  
    "lat": "-15.311300287282295",  
    "lon": "-49.603112939122674",  
    "created": "2025-10-12T05:18:02.156761Z",  
    "completed": null,  
    "id_user": 1,  
    "request_type": 1  
}
```

Figura 5 - Resposta JSON da requisicao POST /request/create/

Nesse caso o response retorna os dados do registro que acabou de ser criado da requisição, o campo “importance” se mantém nulo pois ele representa a criticidade do problema, o qual seria definido posteriormente pelo gestor.

O segundo exemplo mostra o corpo do response em JSON do consumo das rotas GET de detalhes da solicitação “[http://{IP_SERVIDOR}/request/<id>/](http://{IP_SERVIDOR}/request/<id>)” e histórico de status da solicitação “[http://{IP_SERVIDOR}/request/<id>/status/](http://{IP_SERVIDOR}/request/<id>/status)” ambas usando autenticação via cabeçalho e retornando status HTTP 200 OK.

```
{  
    "id": 2,  
    "importance": null,  
    "img1": "codigoBase64imagem1",  
    "img2": "codigoBase64imagem2",  
    "img3": "codigoBase64imagem3",  
    "comment": "Buraco na avenida, proximo a faixa de pedestres",  
    "lat": "-15.311300287282295",  
    "lon": "-49.603112939122674",  
    "created": "2025-10-12T05:18:02.156761Z",  
    "completed": null,  
    "id_user": 1,  
    "request_type": 1  
}
```

Figura 6 - Resposta JSON da requisição GET /request/<id>/

```
{  
    {  
        "id": 2,  
        "dateTime": "2025-10-12T05:26:34.470043Z",  
        "id_request": 1,  
        "status_code": 2  
    },  
    {  
        "id": 1,  
        "dateTime": "2025-10-12T05:26:23.447043Z",  
        "id_request": 1,  
        "status_code": 1  
    }  
}
```

Figura 7 - Resposta JSON da requisição GET /request/<id>/status/

Nesse exemplo após a requisição ter sido criada com status 1 - Em análise, ela foi atualizada para status 2 - Em andamento, tal atualização de status pode ser feita em visão administrativa pois todas requisições criadas pelo aplicativo automaticamente recebem status 1 - Em análise.

Os códigos 200 (OK) e 201 (Created) indicam sucesso nas requisições. Em situações de falha de comunicação com o servidor ou de envio de dados inválidos/inconsistentes, a API pode retornar códigos de erro (por exemplo, 400, 401, 403, 404, 500). Tais respostas exigem tratamento adequado para não degradar a experiência do usuário; no presente projeto, esse tratamento foi implementado majoritariamente no aplicativo.

A camada de interface foi orientada por princípios de usabilidade consolidados no campo do design de interação, com ênfase em consistência visual, visibilidade do estado do sistema, feedback imediato e prevenção de erros. Tais diretrizes alinham-se às heurísticas clássicas de usabilidade, favorecendo o aprendizado progressivo do usuário e reduzindo o retrabalho na navegação entre telas do aplicativo e do painel web. Essa base facilita também a rastreabilidade entre ação do usuário, resposta do sistema e estados da solicitação (status).

A navegação do sistema foi estruturada para reduzir percursos cognitivos e garantir que as tarefas principais (abrir, registrar e acompanhar solicitações) possam ser concluídas em poucos passos, com estados claramente perceptíveis. O mapa de telas do App Cidadão estabelece um fluxo principal linear, com rotas de retorno previsíveis e pontos de reentrada.

Para a versão final deste trabalho, algumas imagens originalmente capturadas na versão inicial do aplicativo foram substituídas por capturas de tela da versão 2.0, já contendo dados de exemplos representativos do uso real do sistema. Essa atualização teve como objetivo tornar mais clara a forma de utilização do aplicativo pelo cidadão, evidenciando o preenchimento dos campos, a visualização das solicitações em um cenário mais próximo da operação em ambiente municipal. Cabe destacar que a versão 2.0 corresponde a uma evolução do protótipo apresentado neste trabalho, desenvolvida em continuidade ao projeto por outra equipe, sem participação direta do autor deste artigo, sendo aqui utilizada apenas como apoio ilustrativo para demonstrar o fluxo de uso com dados reais.

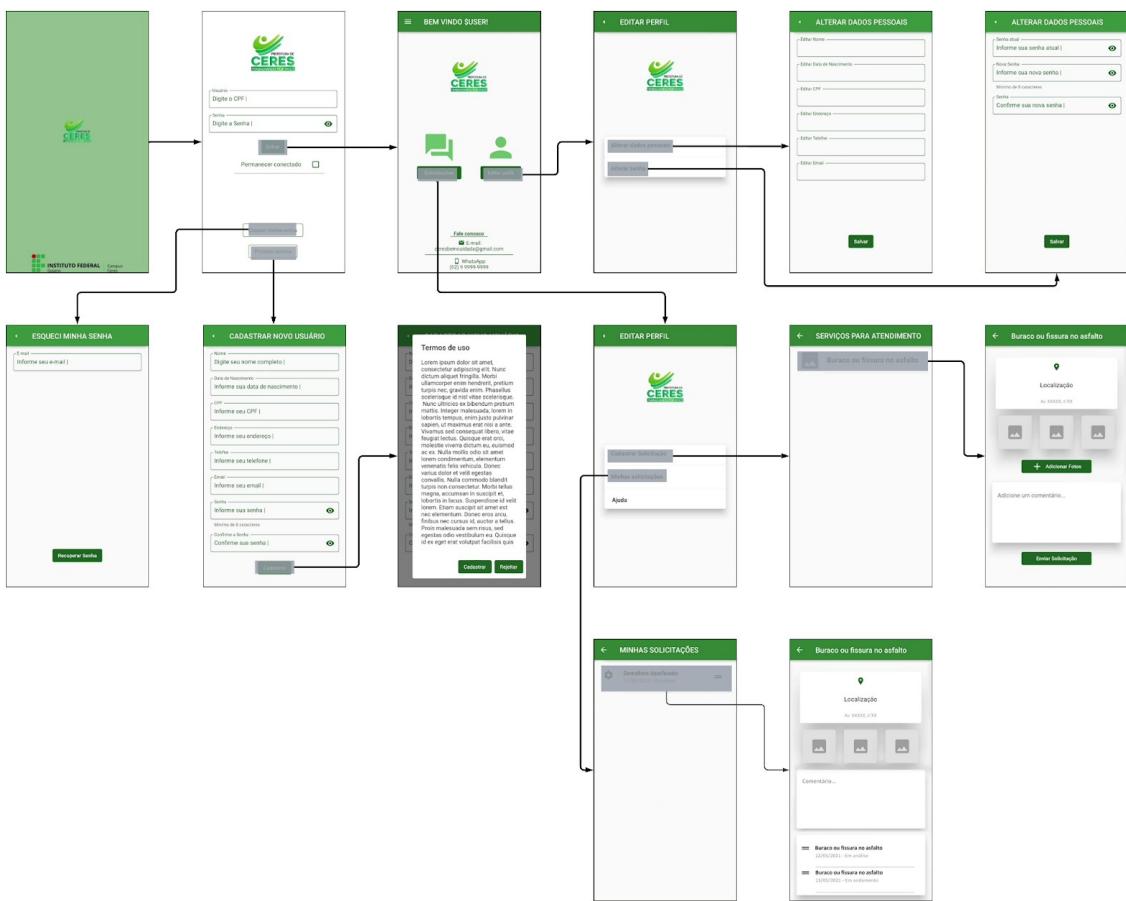


Figura 8 - Diagrama de fluxo de telas do aplicativo



Figura 9 - Tela de splash

A tela de splash é a primeira tela do aplicativo do cidadão, ela é exibida assim que o aplicativo é aberto e exibe os logotipos institucionais (IF Goiano – Campus Ceres e Prefeitura de Ceres). Esta tela tem a função de proporcionar ao

usuário uma experiência visual inicial enquanto o app carrega e inicializa os conteúdos, fornecendo uma transição suave entre a abertura do aplicativo e sua interface principal.



Figura 10 - Tela de login

A tela de login é a tela inicial da aplicação, exibida após a tela de splash quando todos os recursos do sistema já estão carregados. Nela pode-se realizar o login informando as credenciais e clicando no botão “Entrar”, o qual consome a rota GET “user/login/{cpf}/”. A partir dessa tela também podemos acessar as telas de redefinição de senha, pelo botão “Esqueci minha senha”, e a tela de cadastro, pelo botão “Primeiro acesso”.

As primeiras tentativas de autenticação baseada em token evidenciaram a inexperiência da equipe com o ecossistema escolhido. Dificuldades na configuração, na documentação aplicada ao caso e na integração cliente–servidor levaram à adoção temporária de um mecanismo mais simples (credenciais em requisição), priorizando a continuidade do desenvolvimento. Em paralelo, a modelagem com chaves estrangeiras no Django gerou retrabalho; para cumprir o cronograma, optou-se por abordagens mais diretas em alguns pontos, como armazenar até três imagens na própria entidade de solicitação, em vez de manter um relacionamento dedicado para anexos.

Figura 11 - Tela de cadastro

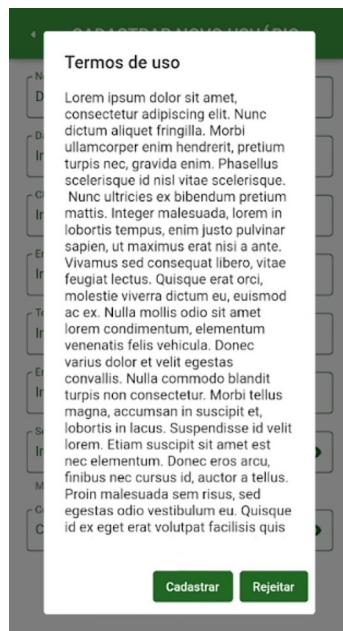


Figura 12 - Termos de uso

Na tela de cadastro pode-se realizar o cadastro de um usuário no sistema, deve-se informar obrigatoriamente um nome, data de nascimento, CPF válido, endereço, telefone, email válido e senha. Para a realização do cadastro, todos os campos são obrigatórios, email e CPF não podem ter sido cadastrados anteriormente e a senha deve ter no mínimo 8 caracteres.

Ao clicar no botão “Cadastrar”, caso alguma informação não passe nas validações, o respectivo campo ficará vermelho com a mensagem informando o que está errado. Após a inserção de todos os dados corretamente a janela de termos de

uso é exibida, exibindo os termos de uso os quais o usuário precisa concordar para finalizar o cadastro. Nesta versão do aplicativo os termos de uso ainda não estavam completamente definidos, mas eles tinham o intuito de deixar o usuário ciente do motivo que aquelas informações eram necessárias e como seriam utilizadas, transparecendo nosso compromisso com a LGPD.

Clicando no botão “Cadastrar” para aceitar os termos de uso, a rota POST “user/create/” é consumida, enviando em formato de JSON todas as informações para a API. O usuário recebe também um link no email informado para ativar a conta, após a ativação da conta a mesma está apta para logar no sistema de registrar requisições. Todos os usuários criados pelo aplicativo são do tipo “1 - Cidadão” e antes da realização da confirmação do email ficam com status inativo, impossibilitando o login no sistema.

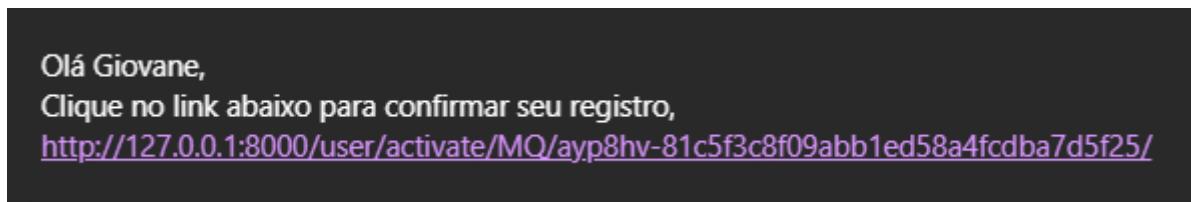


Figura 13 - Exemplo de e-mail de confirmação enviado

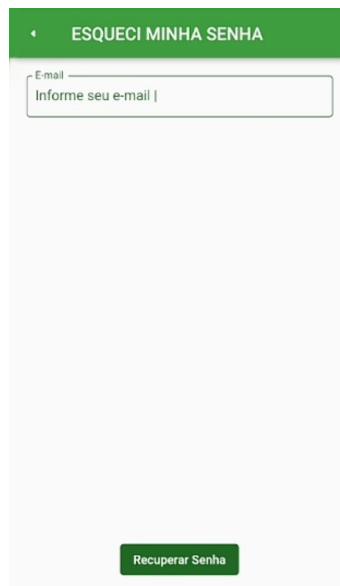


Figura 14 - Recupera senha

Esta tela deve ser utilizada caso o usuário já cadastrado esqueça sua senha ou login. Deve-se informar um email e ao clicar no botão “Recuperar Senha” caso o

email corresponda a uma conta existente a rota POST “user/{id}/reset_password/” é consumida, iniciando o processo de redefinição de senha. O processo consiste em o usuário receber um link via email, esse link abre uma tela para que o usuário informe a nova senha.

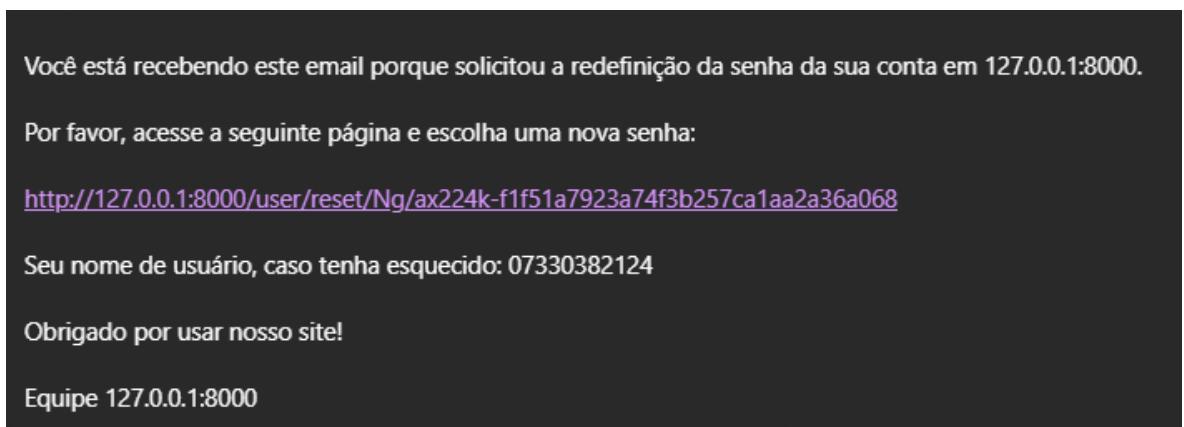


Figura 15 - Exemplo de e-mail enviado para redefinição de senha



Figura 16 - Tela principal

Após a autenticação, o usuário é direcionado à tela principal, que concentra os acessos aos menus Solicitações e Perfil. Nessa mesma tela, há atalhos de contato com a administração na área inferior: ao selecionar o ícone de envelope, o aplicativo de e-mail padrão é aberto com mensagem endereçada a um contato pré-configurado; ao selecionar o ícone de telefone, o WhatsApp é iniciado com uma nova conversa para um número pré-configurado. No estado atual do projeto, tanto o

e-mail quanto o número apresentados funcionam apenas como espaço reservado, pois os canais oficiais de atendimento ainda não foram definitivamente definidos.



Figura 17 - Menu de perfil

Esta tela serve apenas como intermediário entre a tela principal e as telas de edição das informações do usuário. Dando acesso tanto a tela de alteração de dados pessoais quanto a tela de alteração de senha.

Figura 18 - Menu de perfil

Nesta tela o usuário pode alterar alguns dos dados informados no momento do cadastro da conta, os dados do usuário autenticado já vêm carregados nos respectivos campos. Para a atualização das informações, todos os campos precisam estar preenchidos e seguem as mesmas validações da tela de cadastro de usuário. Ao clicar no botão “Salvar”, caso todos os dados estejam válidos a rota PATCH “user/{id}/update/” é consumida, enviando os dados para atualizar as informações do usuário.

The screenshot shows a user interface for changing personal data. At the top, there's a green header bar with the title "ALTERAR DADOS PESSOAIS". Below it, there are three input fields:

- A password field labeled "Senha atual" (Current Password) with the placeholder "Informe sua senha atual |".
- A password field labeled "Nova Senha" (New Password) with the placeholder "Informe sua nova senha |" and a note below it stating "Mínimo de 8 caracteres" (Minimum of 8 characters).
- A password field labeled "Confirmar nova senha" (Confirm New Password) with the placeholder "Confirme sua nova senha |".

At the bottom of the form is a green "Salvar" (Save) button.

Figura 19 - Alterar dados

A tela de alteração de senha permite ao usuário alterar a senha da conta, seguindo a mesma regra da tela de cadastro com o mínimo de oito caracteres. Ao clicar no botão “Salvar” a rota PATCH “user/{id}/change_password/” é consumida, alterando a senha do usuário.



Figura 20 - Menu de solicitar

Acessada através do botão “Solicitações” na tela principal, esta tela serve como intermediário entre a tela principal e as telas de cadastro de solicitações e minhas solicitações. Esta tela também daria acesso à tela de ajuda, mas nesta versão do aplicativo essa tela não foi desenvolvida.



Figura 21: Tipos de serviços para atendimento (Versão 2.0)

Nesta tela, o usuário escolhe o tipo de solicitação que deseja abrir. Exemplos de categorias incluem buraco em via, semáforo apagado e bueiro entupido. No estágio atual do projeto, a lista de tipos ainda não havia sido definida completamente.

Os rótulos, descrições e ícones exibidos são carregados dinamicamente a partir da tabela RequestType e podem ser modificados pela administração do sistema a qualquer momento. A lista apresentada no aplicativo é obtida por meio da rota GET “request/types/”, de modo que qualquer atualização realizada no backend passa a refletir na interface sem necessidade de alterações no código do aplicativo.

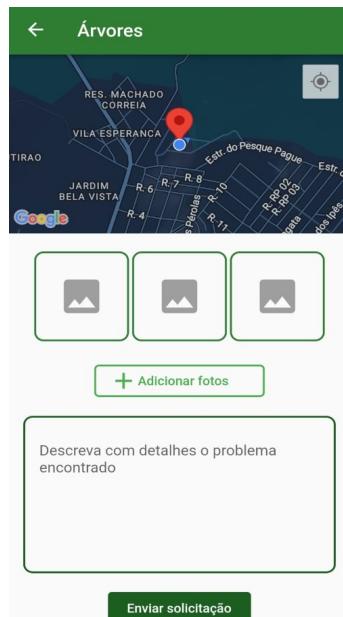


Figura 22 - Cadastro de solicitações (Versão 2.0)

Esta tela permite ao usuário cadastrar uma nova solicitação. Ao abri-la pela primeira vez, caso a permissão de localização ainda não tenha sido concedida, o aplicativo a solicita; uma vez autorizada, a posição é obtida automaticamente e um mapa exibe a localização aproximada do usuário.

Ao acionar o botão “Adicionar fotos”, se a permissão de câmera não estiver ativa, o aplicativo a requisita. Com a permissão concedida, abre-se a interface nativa da câmera para registrar imagens do problema. Após cada captura, é possível manter ou descartar a foto; até três imagens podem ser anexadas à solicitação.

Além das imagens e da geolocalização, é obrigatória a inclusão de um comentário descritivo do problema. Ao selecionar “Enviar solicitação”, o aplicativo

consume a rota POST “request/create/”, transmitindo em JSON as informações para persistência no backend. Todas as solicitações criadas pelo aplicativo são registradas com o status inicial “1 – Em análise”.



Figura 23 - Minhas solicitações (Versão 2.0)

A tela de Minhas solicitações pode ser acessada através do menu de solicitações, esta tela contém a lista de todas as solicitações criadas pelo usuário, cada item da lista exibe um ícone ilustrativo, o tipo da solicitação e o último status deste registro bem como a data do status. Essa lista de requisições do usuário é obtida por meio da rota GET “request/{id}/user/” que busca todas as solicitações de um usuário e também da rota GET “request/{id}/status/” que busca os status dessas solicitações.



Figura 24 - Detalhes da solicitação (Versão 2.0)

O acesso a esta tela ocorre a partir de Minhas Solicitações, ao selecionar um dos registros listados. Nela são apresentados os principais dados da ocorrência: localização (mapa/endereçamento aproximado), imagens anexadas, comentário descritivo e o histórico de status em ordem cronológica.

Trata-se de uma exibição somente de leitura no aplicativo. A edição ou alteração de status é restrita a perfis autorizados no ambiente administrativo (gestores/administradores). Assim, o usuário cidadão pode acompanhar a evolução da demanda, mas não modificar seus dados.

Vários fatores influenciaram na não finalização do escopo completo deste projeto. Dentre estes, ênfase recai sobre restrições organizacionais — em especial o desfalcamento da equipe e o prazo reduzido — e, em paralelo, sobre questões técnicas como autenticação e gestão de sessão, permissões de câmera e GPS, precisão de geolocalização, envio de imagens sob conectividade móvel, consolidação dos tipos de solicitação e coerência nas transições de status.

No planejamento, observou-se a ausência de decomposição do escopo em épicos, funcionalidades e estórias de usuário. A falta desse refinamento dificultou a priorização, a estimativa de esforço e a visualização de dependências. Reuniões diárias entre os subgrupos chegaram a ser cogitadas, mas não se consolidaram; as reuniões semanais, por si só, não foram suficientes para garantir alinhamento contínuo e remoção ágil de impedimentos.

Por fim, verificou-se concentração excessiva de responsabilidades no desenvolvimento: após a conclusão das etapas iniciais de design e documentação por outros grupos, o trabalho de implementação permaneceu, na prática, com número reduzido de integrantes, chegando a recair sobre uma única pessoa em períodos críticos. Esse quadro explica parte das escolhas pragmáticas adotadas e orienta as ações corretivas propostas nos tópicos seguintes.

Para corrigir o descompasso entre o planejado e capacidade, o escopo deveria ter sido decomposto desde o início em épicos, funcionalidades e estórias de usuário com critérios claros de aceite. Essa estrutura permitiria priorização por valor e esforço, visualização de dependências e definição antecipada de um MVP realista, compatível com a disponibilidade de pessoas e o calendário acadêmico.

No eixo organizacional, a comunicação se beneficiaria de encontros curtos e diários por subequipe, focados em progresso e impedimentos, combinados a uma reunião de integração semanal para decisões interdependentes. Atribuição explícita de responsabilidades por módulo, com matriz de papéis e critérios de realocação de integrantes entre os times de design, documentação e desenvolvimento, ajudaria a evitar gargalos quando houver baixa disponibilidade de integrantes. Registro de tarefas em quadro único, com limites de trabalho em progresso e canal de escalonamento para bloqueios, daria previsibilidade ao fluxo.

Quanto à escolha de tecnologias, mudanças de tecnologias utilizadas deveriam passar por registro de decisão arquitetural com análise de custo, risco e prazo. Um período curto de teste técnico, comparando soluções e validando um esqueleto funcional de API, reduziria a incerteza. Mantida a troca, seria recomendável um plano de capacitação dirigido (tutoriais, exemplos mínimos, sessões de pareamento) e a entrega de uma trilha mínima de autenticação e CRUD antes de ampliar o escopo.

Em síntese, a combinação de planejamento orientado a MVP, rituais de coordenação enxutos, decisões tecnológicas justificadas e fundamentos mínimos de engenharia teria reduzido atrasos e escolhas emergenciais. As ações corretivas aqui propostas oferecem um roteiro prático para as próximas iterações, fortalecendo a previsibilidade do projeto e a qualidade do que é entregue.

5. Conclusão

Este trabalho partiu do diagnóstico de fragilidades na comunicação entre cidadão e administração pública para demandas de infraestrutura urbana e propôs um sistema digital centrado no município, composto por aplicativo móvel e API, com rastreabilidade e padronização dos fluxos de atendimento. O objetivo de centralizar solicitações, reduzir assimetrias de informação e favorecer a transparência orientou as decisões de desenho e implementação descritas ao longo do texto.

A execução seguiu abordagem iterativa, com organização da equipe em frentes de design, documentação e desenvolvimento, ciclos semanais e interação periódica com o ente municipal. Diante das restrições de tempo e de capacidade, o escopo foi readequado para um MVP que preserva o ciclo mínimo de valor (cidadão registra, gestão recebe/analisa), mantendo entregáveis funcionais e passíveis de avaliação institucional.

Do ponto de vista técnico, foram entregues: (i) aplicativo do cidadão com abertura de requisições (foto, geolocalização, comentário), consulta e acompanhamento; (ii) API documentada por rotas estáveis para cadastro, autenticação, criação e consulta de solicitações; e (iii) domínio de dados com máquina de estados e catálogos (UserType, RequestType, StatusCode) que sustentam a consistência semântica da experiência. Tais elementos materializam a rastreabilidade tela–endpoint e confirmam a viabilidade tecnológica do protótipo.

Em termos de valor público, a solução alinha-se à fundamentação teórica ao articular padronização dos registros, georreferenciamento e devolutiva ao cidadão, pilares associados a ganhos de eficiência, participação e responsabilidade em serviços digitais de reporte urbano. O modelo adotado cria condições para indicadores operacionais (tempo de resolução, aderência a métricas, recorrência por local/tipo) e cívicos (adoção por canal, distribuição territorial), preparando o caminho para avaliações futuras em cenário real.

O percurso evidenciou limitações que explicam o não encerramento do escopo integral no período acadêmico, notadamente o desfalcamento de equipe, a curva de aprendizado decorrente da troca de tecnologias e a necessidade de simplificações táticas (por exemplo, autenticação e modelagem de anexos) para

preservar o avanço. Essas restrições foram analisadas e converteram-se em lições e ações corretivas para o próximo ciclo.

Como continuidade, recomenda-se: finalizar o painel do gestor e os mecanismos de despacho/fechamento; evoluir o aplicativo das equipes; consolidar a taxonomia de tipos; expor no app os controles de privacidade (desativação/eliminação) já existentes na API; aprimorar acessibilidade (rotulagem, foco e escala tipográfica); adotar estratégias offline-first para conectividade móvel, no qual o aplicativo deve continuar operando mesmo na ausência ou instabilidade de conexão, priorizando o armazenamento e o tratamento local das informações e realizando a sincronização com o servidor apenas quando a rede estiver disponível; e estabilizar padrões mínimos de engenharia (versionamento da API, testes e documentação OpenAPI) para reduzir retrabalho e facilitar integração com sistemas municipais. Esses passos preservam o núcleo validado e ampliam a maturidade do serviço.

Em síntese, o projeto demonstrou a factibilidade técnica e organizacional de um canal único e rastreável para demandas de infraestrutura urbana, confirmando sua aderência ao contexto local e às diretrizes de governo digital. Embora parcial, o conjunto entregue constitui base sólida para implantação piloto e para a obtenção de evidências quantitativas e qualitativas que orientem a evolução do produto e sua institucionalização na rotina municipal.

REFERÊNCIAS

ALENCAR, Danila Fernandes et al. **Experiência do usuário:** análise de usabilidade do ambiente virtual de aprendizagem e-campo (EMBRAPA). *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, Campinas, SP, v. 21, n. 00, p. e023007, 2023. DOI: 10.20396/rdbsci.v21i00.8671313.

BERTOT, J. C.; JAEGER, P. T.; GRIMES, J. M. **Using ICTs to create a culture of transparency: e-government and social media as openness and anti-corruption tools for societies.** *Government Information Quarterly*, v. 27, n. 3, p. 264-271, 2010. Disponível em: <https://doi.org/10.1016/j.giq.2010.03.001>.

BRELÀZ, Gabriela de; CRANTSCHANINOV, Tamara Ilinsky; BELLIX, Laila. **Open Government Partnership na cidade de São Paulo e o programa São Paulo Aberta:** desafios na difusão e institucionalização de uma política global. *Cadernos Gestão Pública e Cidadania*, v. 26, n. 84, p. 1-26, 2021. Disponível em: <https://doi.org/10.1590/1679-395120200023>.

BRASIL. Lei n. 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Diário Oficial da União, Brasília, DF, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 18 out. 2025.

BRASIL. Ministério da Economia. Secretaria de Governo Digital. **Interoperabilidade — Governo Digital.** Brasília, DF, 2020. Disponível em: https://www.gov.br/governodigital/pt-br/infraestrutura-nacional-de-dados/interoperabilidade/copy_of_interoperabilidade. Acesso em: 18 out. 2025

BRITO, Antônio. **Estado digital:** análise crítica sobre indicadores de transformação digital e sua aplicação no setor público. Brasília, DF: Ipea, 2025. (Texto para Discussão, 3097). Disponível em: <https://doi.org/10.38116/td3097-port>.

CASTELLS, Manuel. **A sociedade em rede**. 8. ed. rev. e ampl. São Paulo: Paz e Terra, 2003.

CASTRO, Jessica Lopes Brito de; PEREZ, Olivia Cristina. **Colab: formas de engajamento e participação de mulheres e jovens**. Ciência e Sustentabilidade, Juazeiro do Norte, v. 7, n. 1, p. 182-214, jan./jun. 2023. Disponível em: <https://doi.org/10.56837/ces.v7i01.1133>.

DI MARCO, Cláudio Augusto Ferreira; TERCI, Eliana Tadeu. **Transparência municipal e controle social: a visão dos Observatórios Sociais sobre os portais de transparência e acesso à informação**. Interações – Revista Internacional de Desenvolvimento Local, Campo Grande, v. 23, n. 2, p. 313-330, abr./jun. 2022. Disponível em: <https://doi.org/10.20435/inter.v23i2.2885>.

GANPATI, Sukumar. **Using mobile apps in government**. Washington, DC: IBM Center for The Business of Government, 2015. Disponível em: <https://www.businessofgovernment.org/report/using-mobile-apps-government>.

JANSSEN, Marijn; ESTEVEZ, Elsa. **Lean government and platform-based governance: doing more with less**. Government Information Quarterly, v. 30, supl. 1, p. S1-S8, 2013. Disponível em: <https://doi.org/10.1016/j.giq.2012.11.003>.

KONTOKOSTA, Constantine E.; HONG, Sun-Young. **Urban informatics in the science and practice of planning**. Journal of the American Planning Association, v. 87, n. 1, p. 1-15, 2021. Disponível em: <https://doi.org/10.1080/01944363.2020.1819387>.

MARSON, Priscila R.; MAYER, Verônica F.; NOGUEIRA, Heloísa G. P. **Comunicação interna no âmbito da gestão pública: o caso de uma autarquia pública federal brasileira**. REGE – Revista de Gestão, São Paulo, v. 20, n. 1, p. 43-60, jan./mar. 2013. Disponível em: <https://www.sciencedirect.com/science/article/pii/S180922761630220X>.

MARTINS, C. de S. F.; ALVES, L. R. G. **Um modelo conceitual para a análise de plataformas digitais na gestão pública brasileira.** Revista Brasileira de Direito Público, v. 1, n. 1, p. 1-26, 2024. Disponível em: <https://periodicos.unicesumar.edu.br/index.php/RBDP/article/view/12365>.

mysociety. **FixMyStreet:** report problems in your street., 2025. Disponível em: <https://www.fixmystreet.com>. Acesso em: 18 out. 2025.

NALCHIGAR, Soroosh; FOX, Mark S. **An Ontology for Open 311 Data.** Workshop on Semantic Cities: Beyond Open Data to Models, Standards and Reasoning, AAAI14, Quebec City, 2014. p. 3092-3098.

OFFENHUBER, Dietmar. **Infrastructure legibility – a comparative analysis of Open311-based citizen reporting.** Cambridge Journal of Regions, Economy and Society, 2015, v. 8, p. 93–112. DOI: 10.1093/cjres/rsu001 .

OLIVEIRA, Indira Dutra de Almeida Cabral de et al. **Participação cidadã como ferramenta de inovação no serviço público por meio da plataforma Colab.** ENCONTRO DE ADMINISTRAÇÃO PÚBLICA DA ANPAD, 9., 2022, On-line. 11 p. Disponível em: <https://anpad.com.br/uploads/articles/119/approved/0530e22dea41e24a039563139cdc215e.pdf> .

PAK, Brian; CHUA, Alton Y. K.; MOERE, Andrew Vande. **FixMyStreet Brussels: Socio-Demographic Inequality in Crowdsourced Civic Participation.** Journal of Urban Technology, v. 24, n. 2, p. 65–87. DOI:10.1080/10630732.2016.1270047.

PREFEITURA DE SÃO PAULO. **SP156** – Serviço de atendimento ao cidadão. São Paulo, 2025. Disponível em: <https://transparencia.prefeitura.sp.gov.br/dados-abertos-sp156/> . Acesso em: 18 out. 2025.

PURCINO, Claudevan Campos; GOMES, Heulállya Raissa Ferreira; RIBEIRO, Maria Ivanilse Calderon. **Transformação digital na gestão pública:** uma análise inicial sobre as expectativas e necessidades dos cidadãos em relação à transformação

digital. Revista Multidisciplinar do Nordeste Mineiro, v. 11, n. 1, 2024. DOI: 10.61164/rmn.m.v11i1.3020.

QUINELLO, Robson; BONI, Graziela. **Uma breve análise da cidade de São Paulo, através da plataforma 156, sob a perspectiva de urban facility management.** Engenharia Urbana em Debate, São Carlos, v. 4, n. 1, p. 91-108, 2023. Disponível em: <https://doi.org/10.59550/engurbdebate.v4i1.84>.

REPETTE, Palmyra et al. **The Evolution of City-as-a-Platform:** Smart Urban Development Governance with Collective Knowledge-Based Platform Urbanism. Land, v. 10,n. 1, 33. Disponível em: <https://doi.org/10.3390/land10010033>

SANT'ANA, Ricardo César Gonçalves. **Tecnologia e gestão pública municipal:** mensuração da interação com a sociedade. São Paulo: Editora Unesp; Cultura Acadêmica, 2009. Disponível em: <https://doi.org/10.7476/9788579830105> .

SHIGENO, Kelly Kiyumi. **Human Scapes:** o design de plataformas participativas móveis para a cidade digitalmente conectada. 2016. Dissertação (Mestrado em Arquitetura e Urbanismo) – Faculdade de Arquitetura e Urbanismo, Universidade de São Paulo, São Paulo, 2016. Disponível em: <https://doi.org/10.11606/D.16.2017.tde-16022017-124551> .

SOUZA, Maurício Lima e. **Comunicação interna na gestão pública de uma Instituição Federal de Ensino Superior:** um diagnóstico da UFES campus São Mateus. 2023. Dissertação (Mestrado Profissional em Gestão Pública) – Universidade Federal do Espírito Santo, Vitória, 2023. Disponível em: <https://dspace5.ufes.br/handle/10/16678> .

VILELA-CURY, Luisa et al. **Panorama da arquitetura corporativa na administração pública do Brasil.** Journal of Information Systems and Technology Management, São Paulo, v. 20, 2023. Disponível em: <https://doi.org/10.4301/S1807-1775202320001> .