



Aplicativo Pró-Inclusão e seu impacto na promoção da inclusão social de cidadãos em situação de rua

Willian Cesar de Sena Melo

JUIZ DE FORA
JANEIRO, 2026

Aplicativo Pró-Inclusão e seu impacto na promoção da inclusão social de cidadãos em situação de rua

WILLIAN CESAR DE SENA MELO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: André Luiz de Oliveira

JUIZ DE FORA
JANEIRO, 2026

APLICATIVO PRÓ-INCLUSÃO E SEU IMPACTO NA PROMOÇÃO DA INCLUSÃO SOCIAL DE CIDADÃOS EM SITUAÇÃO DE RUA

Willian Cesar de Sena Melo

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

André Luiz de Oliveira
Doutor em Ciência da Computação

Igor de Oliveira Knop
Doutor em Ciência da Computação

Gleiph Ghiotto Lima de Menezes
Doutor em Ciência da Computação

JUIZ DE FORA

21 DE JANEIRO, 2026

Aos meus pais, por todo o amor, força e incentivo que sempre me deram.

Resumo

O crescimento do número de pessoas em situação de rua no Brasil impõe desafios significativos a voluntários e organizações de apoio, que frequentemente operam com métodos manuais e descentralizados, sem instrumentos digitais integrados para registro, acompanhamento e coordenação de atendimentos (IPEA, 2023). Este trabalho propõe o *Pró-Inclusão*, uma plataforma composta por aplicativo móvel e API *web* com o objetivo de apoiar o registro estruturado de pessoas em situação de rua, a criação e o gerenciamento de pedidos de ajuda e o pareamento geográfico entre usuários que desejam oferecer auxílio e solicitações abertas nas proximidades. O desenvolvimento foi orientado pelos princípios da Engenharia de Usabilidade de Nielsen e por diretrizes de Interação Humano-Computador (IHC), com ênfase em usabilidade, experiência do usuário, acessibilidade e geolocalização. A avaliação do sistema foi conduzida por meio de testes funcionais e avaliação heurística das interfaces, adotando as dez Heurísticas de Usabilidade de Nielsen. Os testes funcionais não evidenciaram falhas críticas nos fluxos principais do sistema, a avaliação heurística identificou violações de severidade variada, em especial relacionadas a prevenção de erros, visibilidade do estado do sistema e consistência, para as quais foram elaboradas recomendações de melhoria. Os resultados indicam que a plataforma é tecnicamente viável e apresenta boa aderência aos princípios de usabilidade avaliados, configurando uma proposta computacional consistente para apoiar ações sociais voltadas à população em situação de rua.

Palavras-chave: inclusão social; pessoas em situação de rua; interação humano-computador; usabilidade; avaliação heurística; geolocalização; aplicativos móveis; tecnologia social.

Abstract

The growing number of homeless individuals in Brazil poses significant challenges to volunteers and support organizations, which frequently rely on manual, decentralized processes and lack integrated digital tools for the registration, monitoring, and coordination of assistance efforts. This work proposes Pró-Inclusão, a platform composed of a mobile application and a web API designed to support the structured registration of homeless individuals, the creation and management of help requests, and the geographic matching between users willing to offer assistance and open requests in their vicinity. The development was guided by Nielsen's Usability Engineering principles and Human-Computer Interaction (HCI) guidelines, with emphasis on usability, user experience, accessibility, and geolocation. System evaluation was conducted through functional testing and heuristic evaluation of the interfaces, adopting Nielsen's ten Usability Heuristics. Functional tests revealed no critical failures in the main system workflows, heuristic evaluation identified violations of varying severity, particularly related to error prevention, visibility of system status, and consistency, for which improvement recommendations were developed. Results indicate that the platform is technically viable and shows good adherence to the evaluated usability principles, constituting a consistent computational proposal to support social actions aimed at the homeless population.

Keywords: social inclusion; homeless people; human-computer interaction; usability; heuristic evaluation; geolocation; mobile applications; social technology.

Agradecimentos

Agradeço primeiramente aos meus pais, pelo apoio incondicional, pelo incentivo aos meus estudos e por sempre acreditarem no meu potencial. Sem o amor, a compreensão e a dedicação de vocês, esta conquista não seria possível.

Agradeço também ao meu orientador, pela orientação atenta, pelos conselhos valiosos e pela confiança depositada ao longo do desenvolvimento deste trabalho. Sua dedicação foi fundamental para a concretização deste projeto.

Estendo meus agradecimentos a todos os professores do departamento, que contribuíram para a minha formação acadêmica por meio de suas aulas, experiências e apoio ao longo da graduação.

Índice

Lista de Figuras	7
Lista de Tabelas	9
Lista de Abreviações	10
1 Introdução	11
1.1 Problema de Pesquisa	13
1.2 Objetivo Geral	13
1.3 Objetivos Específicos	13
1.4 Metodologia Geral	14
1.5 Organização do Documento	14
2 Fundamentação Teórica	15
2.1 Qualidade de Uso	15
2.1.1 Usabilidade	15
2.1.2 Experiência do Usuário	16
2.1.3 Acessibilidade	17
2.1.4 Comunicabilidade	18
2.2 Heurísticas de Usabilidade de Nielsen	18
2.2.1 Visibilidade do estado do sistema	18
2.2.2 Correspondência entre sistema e mundo real	19
2.2.3 Controle e liberdade do usuário	20
2.2.4 Consistência e padrões	21
2.2.5 Prevenção de erros	21
2.2.6 Reconhecimento em vez de memorização	22
2.2.7 Flexibilidade e eficiência de uso	23
2.2.8 Design estético e minimalista	23
2.2.9 Ajudar usuários a reconhecer, diagnosticar e recuperar-se de erros	24
2.2.10 Ajuda e documentação	25
2.3 Método de Avaliação Heurística	26
2.4 Diretrizes de Usabilidade para Dispositivos Móveis	28
2.5 Sistemas de Informação Geográfica e Geolocalização Móvel	28
2.6 Considerações Finais	29
3 Metodologia e Levantamento de Requisitos	30
3.1 Visão Geral da Metodologia	30
3.2 Escopo do Sistema	32
3.3 Características dos Usuários	33
3.3.1 Usuário Básico	33
3.3.2 Voluntário	33
3.4 Cenários de Uso Representativos	33
3.5 Levantamento de Requisitos	34
3.5.1 Requisitos Funcionais	34
3.5.2 Especificação de Casos de Uso Selecionados	35

3.5.3	Requisitos Não Funcionais	36
3.6	Modelagem de Tarefas	37
3.7	Design de IHC Aplicado	41
3.8	Considerações Finais	43
4	Arquitetura, Implementação e Telas do Sistema Pró-Inclusão	44
4.1	Arquitetura Geral da Plataforma	44
4.2	Modelagem de Dados	46
4.3	Back-end - Estrutura e Fluxos	48
4.3.1	Autenticação e Autorização	48
4.3.2	Módulo de Usuários	48
4.3.3	Módulo de Moradores	48
4.3.4	Módulo de Pedidos de Ajuda	49
4.3.5	Módulo de Ofertas de Ajuda	49
4.3.6	Módulo de Listas Gerais	49
4.4	Trechos de Código Selecionados	49
4.4.1	Consulta Geoespacial com Prisma e PostGIS	49
4.4.2	Regra de Negócio para Bloqueio de Ofertas Duplicadas	50
4.4.3	Captura Contínua de Localização no Aplicativo Móvel	52
4.5	Aplicativo Móvel - Estrutura e Funcionalidades	54
4.5.1	Autenticação	57
4.5.2	Cadastro de Morador	57
4.5.3	Mapa e Visualização de Moradores	66
4.5.4	Perfil do Morador	68
4.5.5	Pedidos de Ajuda	69
4.5.6	Detalhes de Oferta Aceita	73
4.5.7	Ofertas Concluídas	74
4.5.8	Minhas Ofertas	75
4.5.9	Finalização do Cadastro	76
4.5.10	Perfil do Usuário	77
4.6	Considerações Finais	78
5	Avaliação do Sistema	79
5.1	Metodologia de Avaliação	79
5.2	Testes Funcionais	81
5.2.1	Roteiros de Teste	81
5.3	Avaliação Heurística	83
5.3.1	Heurísticas Aplicadas	83
5.3.2	Problemas Identificados	84
5.4	Recomendações de Melhoria	84
5.5	Considerações Finais	85
6	Conclusão	86
6.1	Síntese dos Resultados	86
6.2	Contribuições do Trabalho	87
6.3	Limitações	87
6.4	Trabalhos Futuros	88
6.5	Considerações Finais	88
	Bibliografia	90

Lista de Figuras

2.1	Colmeia da Experiência do Usuário. Fonte: (AGNI, 2012).	17
2.2	Heurística 1: Visibilidade do estado do sistema. Fonte: (Nielsen Norman Group, 1994).	19
2.3	Heurística 2: Correspondência entre sistema e mundo real. Fonte: (Nielsen Norman Group, 1994).	20
2.4	Heurística 3: Controle e liberdade do usuário. Fonte: (Nielsen Norman Group, 1994).	20
2.5	Heurística 4: Consistência e padrões. Fonte: (Nielsen Norman Group, 1994).	21
2.6	Heurística 5: Prevenção de erros. Fonte: (Nielsen Norman Group, 1994).	22
2.7	Heurística 6: Reconhecimento em vez de memorização. Fonte: (Nielsen Norman Group, 1994).	22
2.8	Heurística 7: Flexibilidade e eficiência de uso. Fonte: (Nielsen Norman Group, 1994).	23
2.9	Heurística 8: Design estético e minimalista. Fonte: (Nielsen Norman Group, 1994).	24
2.10	Heurística 9: Ajudar usuários a reconhecer, diagnosticar e recuperar-se de erros. Fonte: (Nielsen Norman Group, 1994).	25
2.11	Heurística 10: Ajuda e documentação. Fonte: (Nielsen Norman Group, 1994).	25
2.12	Etapas do método de avaliação heurística. Fonte: autoria própria.	27
3.1	Fluxograma da metodologia com ciclos de implementação e prototipação evolutiva.	32
3.2	Análise Hierárquica de Tarefas para cadastrar morador.	38
3.3	Análise Hierárquica de Tarefas para criar pedido de ajuda.	39
3.4	Análise Hierárquica de Tarefas para ofertar ajuda.	40
3.5	Análise Hierárquica de Tarefas para explorar o mapa.	41
3.6	Interface implementada da navegação principal do aplicativo.	42
3.7	Interface implementada para criação de pedidos de ajuda.	43
4.1	Visão geral da arquitetura da plataforma <i>Pró-Inclusão</i> . Fonte: autoria própria.	45
4.2	Diagrama Entidade-Relacionamento do banco de dados <i>Pró-Inclusão</i> . Fonte: autoria própria.	47
4.3	Fluxo de telas de autenticação e encaminhamento para a aplicação. Fonte: autoria própria.	54
4.4	Fluxo de telas para criação e cadastro de morador. Fonte: autoria própria.	55
4.5	Fluxo de telas para solicitar e ofertar ajuda (perfil voluntário). Fonte: autoria própria.	56
4.6	Fluxo de telas para ofertar ajuda (perfil básico). Fonte: autoria própria.	56
4.7	Tela de <i>login</i> do usuário. Fonte: autoria própria.	57
4.8	Interface implementada para cadastro de informações pessoais do morador. Fonte: autoria própria.	58
4.9	Interface implementada para registrar situação de rua e contexto. Fonte: autoria própria.	59

4.10	Interface implementada para seleção de localização no mapa. Fonte: autoria própria.	60
4.11	Interface implementada listando localizações do morador. Fonte: autoria própria.	61
4.12	Interface implementada para cadastro de familiar. Fonte: autoria própria.	62
4.13	Interface implementada com lista de familiares cadastrados. Fonte: autoria própria.	63
4.14	Interface implementada para registrar serviços sociais. Fonte: autoria própria.	64
4.15	Interface implementada para captura da foto de perfil. Fonte: autoria própria.	65
4.16	Interface implementada para registro de consentimento em áudio ou vídeo. Fonte: autoria própria.	66
4.17	Mapa com moradores carregados por <i>viewport</i> . Fonte: autoria própria.	67
4.18	Detalhes e ações disponíveis ao selecionar um morador no mapa. Fonte: autoria própria.	68
4.19	Perfil completo do morador. Fonte: autoria própria.	69
4.20	Interface implementada para criação de pedido de ajuda. Fonte: autoria própria.	70
4.21	Interface implementada para gerenciamento de pedidos do usuário. Fonte: autoria própria.	71
4.22	Interface implementada com pedidos de ajuda disponíveis. Fonte: autoria própria.	72
4.23	Interface implementada para acompanhar ofertas recebidas em um pedido. Fonte: autoria própria.	73
4.24	Detalhes e contato do voluntário na oferta aceita. Fonte: autoria própria.	74
4.25	Interface implementada com histórico de ofertas concluídas. Fonte: autoria própria.	75
4.26	Interface implementada com ofertas realizadas pelo usuário. Fonte: autoria própria.	76
4.27	Interface implementada para finalização do cadastro do usuário. Fonte: autoria própria.	77
4.28	Interface implementada do perfil do usuário. Fonte: autoria própria.	78

Lista de Tabelas

3.1	Requisitos funcionais do sistema <i>Pró-Inclusão</i>	35
3.3	Especificação do caso de uso associado ao requisito RF08 - Criar pedido de ajuda.	35
3.5	Especificação do caso de uso associado ao requisito RF12 - Oferecer ajuda a pedido aberto.	36
3.7	Requisitos não funcionais do sistema <i>Pró-Inclusão</i>	36
5.1	Roteiros de teste funcional para fluxos principais do sistema.	81
5.3	Problemas encontrados durante a avaliação heurística.	84

Lista de Abreviações

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	Interface de Programação de Aplicações
CRAS	Centro de Referência de Assistência Social
CREAS	Centro de Referência Especializado de Assistência Social
DCC	Departamento de Ciência da Computação
HCI	Human-Computer Interaction
HTA	Análise Hierárquica de Tarefas
HTTP	Hypertext Transfer Protocol
IHC	Interação Humano-Computador
MVC	Model-View-Controller
ORM	Mapeamento Objeto-Relacional
REST	Representational State Transfer
RF	Requisito Funcional
RNF	Requisito Não Funcional
S3	Simple Storage Service
SSO	Single Sign-On
TICs	Tecnologias de Informação e Comunicação
UFJF	Universidade Federal de Juiz de Fora
URL	Uniform Resource Locator
UX	Experiência do Usuário

1 Introdução

A expansão do acesso à Internet móvel, a popularização de *smartphones* e o amadurecimento de serviços baseados em localização transformaram profundamente a maneira como indivíduos interagem com o espaço urbano e com serviços digitais. Soluções móveis deixaram de ser meros canais informativos para se tornarem ferramentas ativas de mediação social, capazes de conectar pessoas, promover colaboração e apoiar ações humanitárias. No contexto brasileiro, esse potencial tecnológico se torna especialmente relevante diante de dados que apontam para o crescimento do número de pessoas em situação de rua e da carência de instrumentos eficazes para identificar necessidades, registrar atendimentos e organizar iniciativas voluntárias (IPEA, 2023).

A ausência de sistemas integrados e confiáveis pode resultar em dificuldades como duplicidade de esforços, falta de informação atualizada sobre indivíduos atendidos, registros fragmentados em diferentes organizações e baixa capacidade de identificar situações emergenciais. Nesse contexto, voluntários tendem a encontrar obstáculos para localizar moradores de rua em grandes centros urbanos, registrar detalhes relevantes sobre sua situação e coordenar pedidos de ajuda de forma transparente e ágil. Tais limitações comprometem a eficiência de ações sociais e dificultam a continuidade de atendimentos.

Para ilustrar esse cenário, pode-se considerar a rotina de uma voluntária que atua em uma ação social recorrente em uma região central. Sem apoio tecnológico, o registro das pessoas atendidas tende a ocorrer em anotações dispersas em cadernos ou planilhas pessoais, dificultando a atualização de dados, a identificação de retornos frequentes e a coordenação com outros grupos. Em dias de maior demanda, a voluntária precisa relembrar, a partir da memória, onde determinados moradores costumam permanecer e quais pedidos de ajuda ainda não foram atendidos, o que favorece duplicidade de esforços e a invisibilização de casos mais críticos. Com uma plataforma georreferenciada como a proposta neste trabalho, o mesmo cenário poderia contar com um mapa atualizado de moradores cadastrados, histórico de pedidos de ajuda e registro de ofertas realizadas, facilitando a consulta coordenada das necessidades abertas nas proximidades.

Nesse contexto, tecnologias móveis com suporte a geolocalização oferecem caminhos promissores ao permitir a coleta estruturada de dados, a consulta contextualizada por proximidade, o registro de informações multimídia e a interação entre diferentes perfis de usuários. Além de potencialmente viabilizar uma logística mais eficiente de atendimentos e doações, espera-se que a combinação de registros georreferenciados e perfis detalhados contribua para a continuidade dos vínculos e para o planejamento de ações recorrentes em territórios específicos. Entretanto, tais tecnologias exigem rigor na aplicação de princípios de Interação Humano-Computador (IHC), que orientam a criação de interfaces intuitivas, acessíveis e cognitivamente adequadas aos perfis dos usuários envolvidos, desde cidadãos com pouca familiaridade tecnológica até voluntários experientes (PREECE; ROGERS; SHARP, 2015; DIX et al., 2004). Princípios clássicos de usabilidade, como visibilidade, consistência e prevenção de erros, e estudos contemporâneos sobre experiência do usuário, oferecem base conceitual fundamental para o projeto de soluções dessa natureza (NIELSEN, 1993; NIELSEN; MOLICH, 1990; NIELSEN; BUDIU, 2012).

Este Trabalho de Conclusão de Curso (TCC) em Ciência da Computação propõe o *Pró-Inclusão*, uma plataforma digital composta por um aplicativo móvel e uma API *web* para apoiar ações de identificação, registro e acompanhamento de pessoas em situação de rua. A solução busca permitir que voluntários cadastrem perfis de moradores, registrem localizações georreferenciadas e organizem pedidos de ajuda, ao mesmo tempo em que cidadãos possam explorar necessidades próximas, oferecer ajuda e acompanhar o andamento das ofertas.

A arquitetura do protótipo integra um *back-end* (NestJS, Prisma e PostgreSQL com PostGIS para consultas espaciais) e um aplicativo móvel (Expo/React Native), com autenticação, armazenamento de mídias e integração com serviços de geocodificação. Essas escolhas tecnológicas são descritas como parte da materialização do artefato, mas o foco do trabalho é investigar como uma abordagem computacional pode apoiar o processo de coleta, organização e consulta de informações no domínio de ações sociais voltadas à população em situação de rua.

1.1 Problema de Pesquisa

Os métodos utilizados por organizações sociais e voluntários para apoiar pessoas em situação de rua tendem a ser predominantemente manuais, descentralizados e informais. Essa realidade aponta para a ausência de uma ferramenta unificada que permita registrar dados completos de moradores, organizar pedidos de ajuda, conectar voluntários a necessidades próximas e garantir segurança e consentimento no tratamento das informações.

A partir desse cenário, formula-se o problema desta pesquisa:

Como melhorar o processo de coleta de informações de pessoas em situação de rua e aproximar cidadãos e entidades de apoio?

1.2 Objetivo Geral

Propor e avaliar uma abordagem computacional para apoiar a coleta estruturada e georreferenciada de informações sobre pessoas em situação de rua e favorecer a articulação entre cidadãos, voluntários e entidades de apoio, considerando requisitos de usabilidade, privacidade e consentimento no tratamento de dados sensíveis.

1.3 Objetivos Específicos

- Caracterizar o domínio do problema e o processo de registro e atendimento, identificando atores, informações essenciais e limitações observadas em abordagens manuais e descentralizadas.
- Definir requisitos de informação e de qualidade de uso (usabilidade e experiência do usuário), além de requisitos de privacidade e consentimento adequados ao tratamento de dados sensíveis.
- Propor um modelo de dados e fluxos de interação que suportem cadastro, atualização e consulta por proximidade, com ênfase na rastreabilidade de atendimentos e na redução de duplicidade de esforços.
- Projetar as interfaces e construir um protótipo funcional da plataforma *Pró-Inclusão*

para materializar a proposta e viabilizar sua avaliação.

- Avaliar o protótipo quanto à usabilidade e à adequação ao contexto de uso, utilizando avaliação heurística e testes funcionais baseados em cenários.
- Analisar os resultados da avaliação e derivar recomendações de melhoria para o processo e para a solução proposta.

1.4 Metodologia Geral

A metodologia segue as etapas da Engenharia de Usabilidade segundo Nielsen (NIELSEN, 1993), incluindo levantamento do contexto, definição de escopo, caracterização de usuários, especificação de requisitos e avaliação. Como parte da materialização da proposta, realiza-se a construção iterativa do protótipo (projeto de interfaces e implementação), permitindo aplicar as diretrizes de IHC ao longo do desenvolvimento e, ao final, avaliar o artefato em cenários representativos.

1.5 Organização do Documento

O Capítulo 2 apresenta a fundamentação teórica que sustenta o trabalho, com foco em Interação Humano-Computador, usabilidade, experiência do usuário, acessibilidade e avaliação heurística. O Capítulo 3 descreve a metodologia e o levantamento de requisitos, contextualizando o domínio do problema e os fluxos de uso considerados. O Capítulo 4 detalha a solução proposta e sua materialização em protótipo, abordando arquitetura e telas principais. O Capítulo 5 apresenta o processo e os resultados da avaliação. Por fim, o Capítulo 6 discute as conclusões, limitações e perspectivas de trabalhos futuros.

2 Fundamentação Teórica

Neste Capítulo são apresentados os conceitos necessários à compreensão das contribuições deste trabalho de conclusão de curso. Na Seção 2.1 são apresentados os critérios de Qualidade de Uso relevantes ao projeto do aplicativo *Pró-Inclusão*, incluindo usabilidade, experiência do usuário, acessibilidade e comunicabilidade. Na Seção 2.2 são apresentadas as Heurísticas de Usabilidade de Nielsen, utilizadas como referência para o projeto e a avaliação das interfaces do aplicativo. Na Seção 2.3 é apresentado o Método de Avaliação Heurística adotado neste trabalho. Na Seção 2.4 são apresentadas as Diretrizes de Usabilidade para Dispositivos Móveis que orientaram as decisões de *design* ao longo do desenvolvimento. Na Seção 2.5 são apresentados os fundamentos de Sistemas de Informação Geográfica e Geolocalização Móvel, que sustentam as funcionalidades de mapeamento e consulta por proximidade do aplicativo. Finalmente, na Seção 2.6 são apresentadas as considerações finais deste Capítulo. No Capítulo 3 são descritas as etapas de metodologia e levantamento de requisitos do aplicativo *Pró-Inclusão*.

2.1 Qualidade de Uso

A qualidade de uso (*quality in use*) refere-se ao grau em que um sistema permite que usuários alcancem objetivos específicos de forma eficaz, eficiente e satisfatória em um contexto de uso definido (International Organization for Standardization, 2018). Em aplicações móveis voltadas a ações sociais, como o *Pró-Inclusão*, o contexto é determinante: usuários podem operar com atenção dividida, em ambientes externos, sob pressão de tempo e, muitas vezes, com conectividade instável.

2.1.1 Usabilidade

Usabilidade pode ser entendida como a facilidade de aprender e utilizar um sistema, bem como a satisfação decorrente do uso (NIELSEN, 1993). No âmbito da engenharia de usabilidade, é comum decompor o conceito em fatores como facilidade de apren-

dizado (*learnability*), eficiência (*efficiency*), facilidade de memorização (*memorability*), segurança/prevenção de erros (*safety*) e satisfação (*satisfaction*) (NIELSEN, 1993).

No *Pró-Inclusão*, esses fatores se manifestam em tarefas críticas, como cadastrar um morador, registrar consentimento, criar pedidos de ajuda e oferecer ajuda a partir de listagens georreferenciadas. Interfaces que exigem memorização excessiva, múltiplos passos ou não deixam claro o estado de uma operação (por exemplo, “pedido criado”, “oferta enviada”) tendem a degradar a usabilidade, sobretudo em cenários de uso em campo.

2.1.2 Experiência do Usuário

A experiência do usuário (UX) envolve uma avaliação subjetiva dos efeitos do uso do sistema, incluindo emoções e percepções como confiança, frustração, sensação de controle e satisfação (PREECE; ROGERS; SHARP, 2015; SHNEIDERMAN et al., 2016; NORMAN, 2013). Embora relacionada à usabilidade, UX não se limita a ela: um sistema pode ser funcional, mas gerar insegurança, desconforto ou baixa confiança em razão de linguagem inadequada, estética confusa ou falta de feedback.

Para ilustrar esse caráter multidimensional, a Figura 2.1 apresenta a chamada “colmeia” da UX, que organiza a experiência em dimensões complementares — como ser útil, utilizável, desejável, encontrável, acessível, credível e valioso — destacando que a percepção final do usuário emerge do equilíbrio entre esses aspectos.



Figura 2.1: Colmeia da Experiência do Usuário. Fonte: (AGNI, 2012).

Em aplicações móveis, elementos frequentemente associados à UX incluem: (i) um projeto de interface que favoreça reconhecimento de ações e interpretação de respostas; (ii) oferta de funcionalidades estritamente necessárias; (iii) organização natural da informação e do fluxo; (iv) estratégia adequada de conteúdo (texto, imagens e multimídia) para comunicar ações e estados; e (v) flexibilidade para diferentes dispositivos e condições de uso (NIELSEN; BUDIUI, 2012; PREECE; ROGERS; SHARP, 2015).

2.1.3 Acessibilidade

Acessibilidade diz respeito à capacidade de um sistema ser utilizado por pessoas com diferentes capacidades e limitações, removendo barreiras que dificultem acesso, compreensão e interação (W3C, 2023; International Organization for Standardization, 2018). Além de deficiências permanentes, aplicações móveis devem considerar limitações temporárias impostas pelo contexto (por exemplo, atenção dividida, luminosidade elevada, ruído, uso com uma mão e conectividade instável) (NIELSEN; BUDIUI, 2012).

No *Pró-Inclusão*, isso implica oferecer textos legíveis, controles com área de toque adequada, mensagens de erro claras e fluxos que tolerem interrupções (como perda de rede ao enviar mídia). Também demanda cuidado com linguagem e com a privacidade, uma

vez que o sistema lida com dados sensíveis de uma população vulnerável.

2.1.4 Comunicabilidade

Comunicabilidade refere-se à capacidade da interface de transmitir, de forma efetiva, as intenções de *design* e o “como usar” embutido nas escolhas de interação, rótulos, mensagens e organização das telas (SOUZA, 2005; BARBOSA; SILVA, 2010). Sob essa perspectiva, a interface atua como um meio de comunicação entre quem projetou o sistema e quem o utiliza: quando a comunicação falha, o usuário tende a interpretar incorretamente funções, estados e restrições.

No contexto do *Pró-Inclusão*, a comunicabilidade é especialmente relevante para evitar ambiguidades em ações como “ofertar ajuda”, “aceitar oferta”, “encerrar pedido” e “registrar consentimento”. A ausência de feedback e de pistas semânticas pode levar a erros com impacto social (por exemplo, duplicidade de ofertas, perda de registro e exposição indevida de dados).

2.2 Heurísticas de Usabilidade de Nielsen

As heurísticas de Nielsen constituem um conjunto de princípios gerais para orientar projeto e avaliação de interfaces, amplamente utilizados por sua aplicabilidade prática e por cobrirem problemas recorrentes em sistemas interativos (NIELSEN, 1993; NIELSEN; MOLICH, 1990). A seguir, as dez heurísticas são sintetizadas com ênfase no tipo de decisão de *design* que elas influenciam no *Pró-Inclusão*.

2.2.1 Visibilidade do estado do sistema

O sistema deve manter os usuários informados, em tempo adequado, sobre o que está acontecendo. Em aplicações móveis isso inclui carregamentos, confirmações de salvamento e estados de sincronização (por exemplo, quando uma oferta foi enviada, mas ainda está pendente por instabilidade de rede) (NIELSEN, 1993).

A Figura 2.2 ilustra uma interface com elementos de estado destacados, reforçando que o usuário precisa perceber rapidamente o que está acontecendo, o que foi selecionado

e qual é a situação atual de uma ação ou contexto.



Figura 2.2: Heurística 1: Visibilidade do estado do sistema. Fonte: (Nielsen Norman Group, 1994).

2.2.2 Correspondência entre sistema e mundo real

A linguagem e os conceitos devem refletir o domínio de uso, evitando termos técnicos desnecessários. Para o *Pró-Inclusão*, isso envolve nomenclaturas claras de perfis, pedidos, ofertas e serviços sociais, bem como descrições que façam sentido para voluntários e cidadãos (NIELSEN, 1993).

A Figura 2.3 representa a ideia de traduzir conceitos do mundo real para a interface, usando metáforas e rótulos familiares, de modo que o usuário compreenda ações e informações sem precisar aprender jargões do sistema.



Figura 2.3: Heurística 2: Correspondência entre sistema e mundo real. Fonte: (Nielsen Norman Group, 1994).

2.2.3 Controle e liberdade do usuário

O usuário deve conseguir desfazer ações e sair de estados indesejados sem custo elevado. Isso se traduz em permitir cancelar operações, editar registros, voltar etapas de formulários e recuperar-se de erros sem perder dados já preenchidos (NIELSEN, 1993).

A Figura 2.4 evidencia controles de navegação e de desfazer/refazer, destacando que a interface deve oferecer “saídas” claras (por exemplo, voltar, cancelar e editar) para que o usuário mantenha sensação de controle durante o uso.

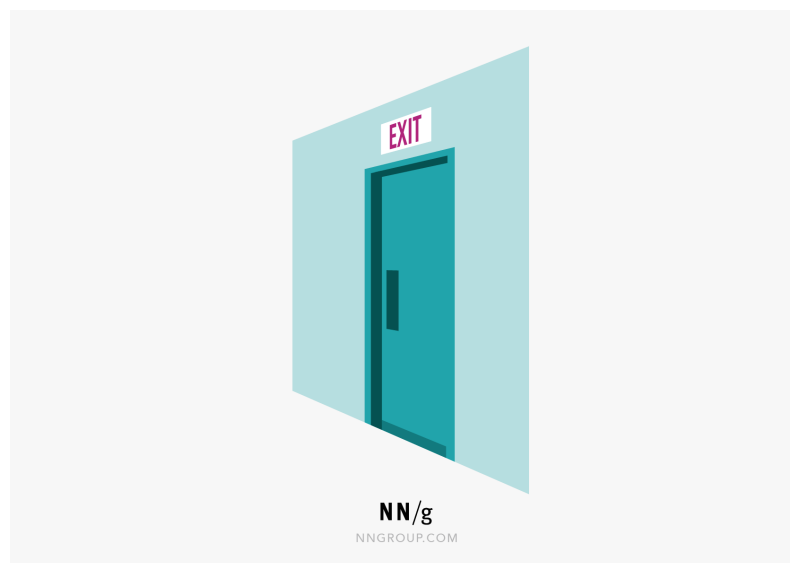


Figura 2.4: Heurística 3: Controle e liberdade do usuário. Fonte: (Nielsen Norman Group, 1994).

2.2.4 Consistência e padrões

Títulos, botões, ícones e fluxos devem ser consistentes entre telas. A consistência reduz esforço cognitivo e facilita aprendizagem, especialmente quando o aplicativo possui diferentes perfis de uso (básico e voluntário) (NIELSEN, 1993).

A Figura 2.5 ilustra a repetição de padrões visuais e de interação, enfatizando que elementos semelhantes devem se comportar de forma semelhante, reduzindo incerteza e erros ao alternar entre telas e funcionalidades.

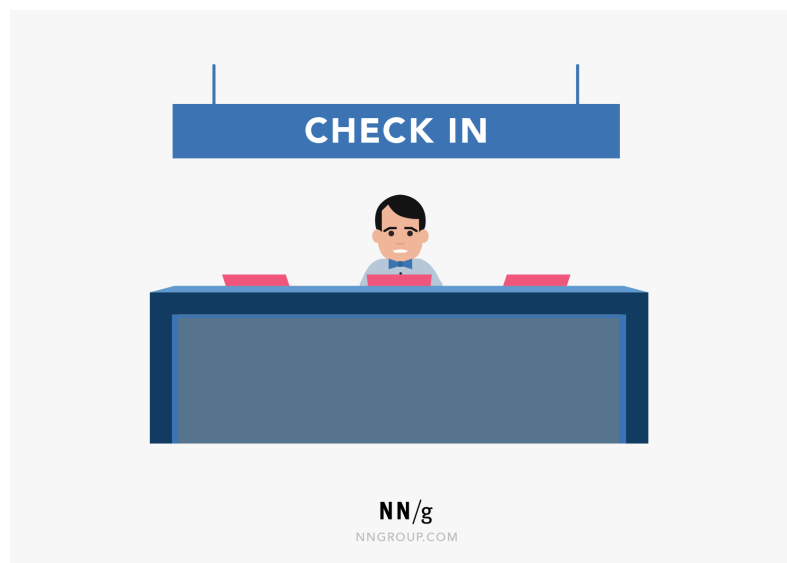


Figura 2.5: Heurística 4: Consistência e padrões. Fonte: (Nielsen Norman Group, 1994).

2.2.5 Prevenção de erros

Mais do que boas mensagens de erro, o sistema deve reduzir a chance de erro ocorrer. Exemplos são validações de campos obrigatórios, confirmações em ações irreversíveis (como encerrar um pedido) e bloqueios para evitar duplicidade de oferta no mesmo pedido (NIELSEN, 1993).

A Figura 2.6 destaca mecanismos de prevenção, como validações e confirmações, reforçando que a interface deve orientar o usuário antes do erro acontecer, evitando consequências indesejadas.

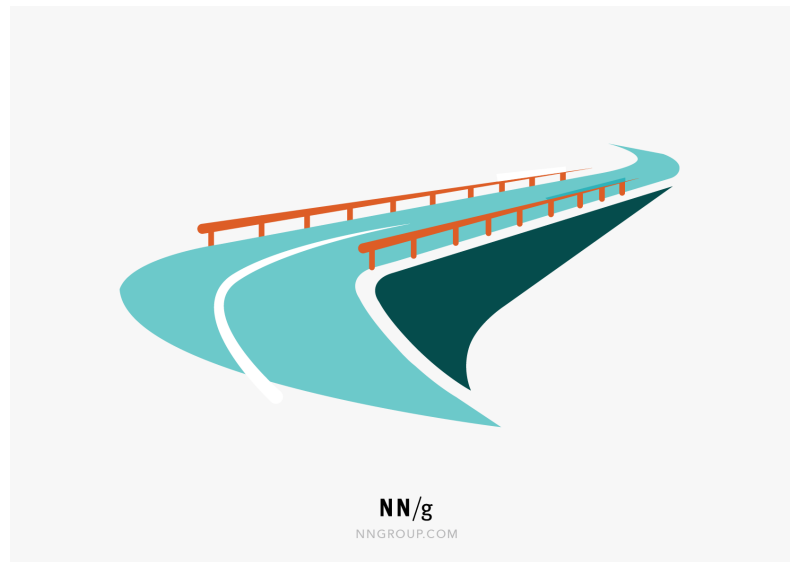


Figura 2.6: Heurística 5: Prevenção de erros. Fonte: (Nielsen Norman Group, 1994).

2.2.6 Reconhecimento em vez de memorização

Opções e informações devem estar visíveis, reduzindo a necessidade de lembrar passos ou significados. Listas, rótulos e indicações contextuais (por exemplo, status do pedido, distância e prioridade) são essenciais em cenários de uso com atenção dividida (NIELSEN, 1993).

A Figura 2.7 sugere o uso de pistas visuais e opções disponíveis na interface, enfatizando que o usuário deve reconhecer ações e caminhos possíveis a partir do que vê, em vez de memorizar comandos ou sequências.

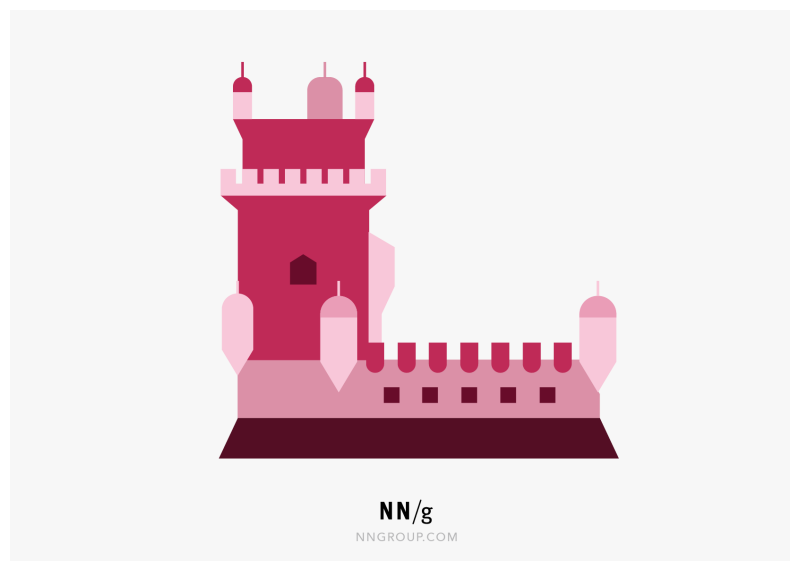


Figura 2.7: Heurística 6: Reconhecimento em vez de memorização. Fonte: (Nielsen Norman Group, 1994).

2.2.7 Flexibilidade e eficiência de uso

O sistema deve atender iniciantes e usuários frequentes, oferecendo atalhos e fluxos eficientes. No *Pró-Inclusão*, isso envolve reduzir passos em tarefas repetidas (cadastro de morador, criação de pedido), reutilizar dados e oferecer preenchimento assistido quando possível (NIELSEN, 1993; NIELSEN; BUDIUI, 2012).

A Figura 2.8 ilustra recursos que aceleram tarefas, como atalhos e ações rápidas, destacando que a interface deve permitir que usuários experientes executem fluxos recorrentes com menos passos, sem prejudicar a compreensão de iniciantes.

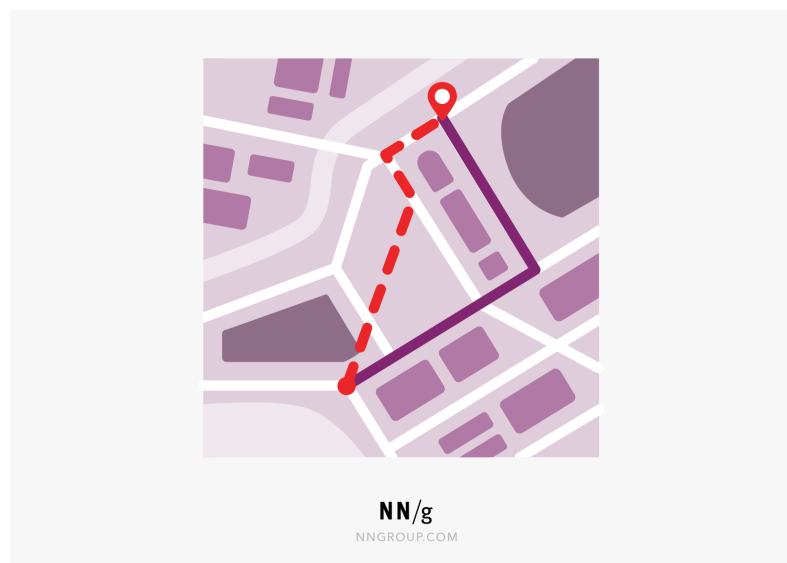


Figura 2.8: Heurística 7: Flexibilidade e eficiência de uso. Fonte: (Nielsen Norman Group, 1994).

2.2.8 Design estético e minimalista

Interfaces devem evitar informações irrelevantes. Em telas móveis, excesso de elementos compete por atenção e reduz legibilidade; a priorização de conteúdo crítico (estado do pedido, ações principais, localização) melhora a tomada de decisão (NIELSEN, 1993; KRUG, 2014).

A Figura 2.9 representa uma interface com poucos elementos essenciais, reforçando que a organização visual deve privilegiar o que é mais importante para a tarefa e reduzir ruído informacional.



Figura 2.9: Heurística 8: Design estético e minimalista. Fonte: (Nielsen Norman Group, 1994).

2.2.9 Ajudar usuários a reconhecer, diagnosticar e recuperar-se de erros

Mensagens de erro devem ser claras, indicar a causa e orientar correção. Para dados sensíveis e formulários longos, isso inclui destacar campos com problema e preservar conteúdo já inserido (NIELSEN, 1993).

A Figura 2.10 ilustra mensagens e indicações de problema que ajudam o usuário a entender o que ocorreu e como corrigir, reforçando a necessidade de orientação acionável e linguagem compreensível.



Figura 2.10: Heurística 9: Ajudar usuários a reconhecer, diagnosticar e recuperar-se de erros. Fonte: (Nielsen Norman Group, 1994).

2.2.10 Ajuda e documentação

Mesmo sistemas bem projetados se beneficiam de orientações curtas, especialmente em funcionalidades raras ou sensíveis (por exemplo, consentimento, privacidade e papéis de usuário). Ajuda contextual e textos curtos de orientação reduzem erros e aumentam confiança (NIELSEN, 1993).

A Figura 2.11 evidencia a presença de recursos de ajuda e documentação, reforçando que instruções curtas e contextualizadas devem estar disponíveis quando necessário, sem interromper o fluxo principal de uso.

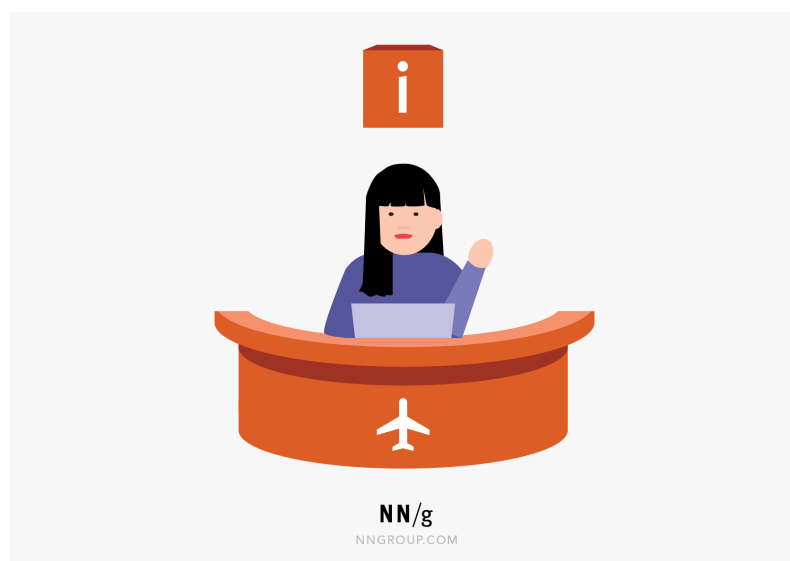


Figura 2.11: Heurística 10: Ajuda e documentação. Fonte: (Nielsen Norman Group, 1994).

2.3 Método de Avaliação Heurística

A avaliação heurística é um método de inspeção no qual avaliadores percorrem telas e fluxos comparando-os a um conjunto de heurísticas, registrando violações e oportunidades de melhoria (NIELSEN; MOLICH, 1990; NIELSEN, 1993). Em aplicações móveis, esse método é valioso por exigir pouca infraestrutura e permitir identificar problemas de usabilidade antes (ou em paralelo) a testes com usuários.

O processo de avaliação pode ser organizado em etapas de preparação, coleta e interpretação, e síntese dos resultados com recomendações.

A Figura 2.12 sintetiza essas etapas e explicita como os registros de violações são coletados, consolidados e transformados em recomendações priorizadas. O diagrama organiza o fluxo em três macrofases: (i) **Preparação**, na qual se define escopo, tarefas, avaliadores, heurísticas e o modelo de registro; (ii) **Coleta e Interpretação**, em que avaliadores percorrem telas e executam tarefas para registrar violações com contexto e evidências; e (iii) **Síntese e Recomendações**, etapa em que os achados são consolidados, deduplicados e classificados por severidade para orientar priorização de correções e recomendações.

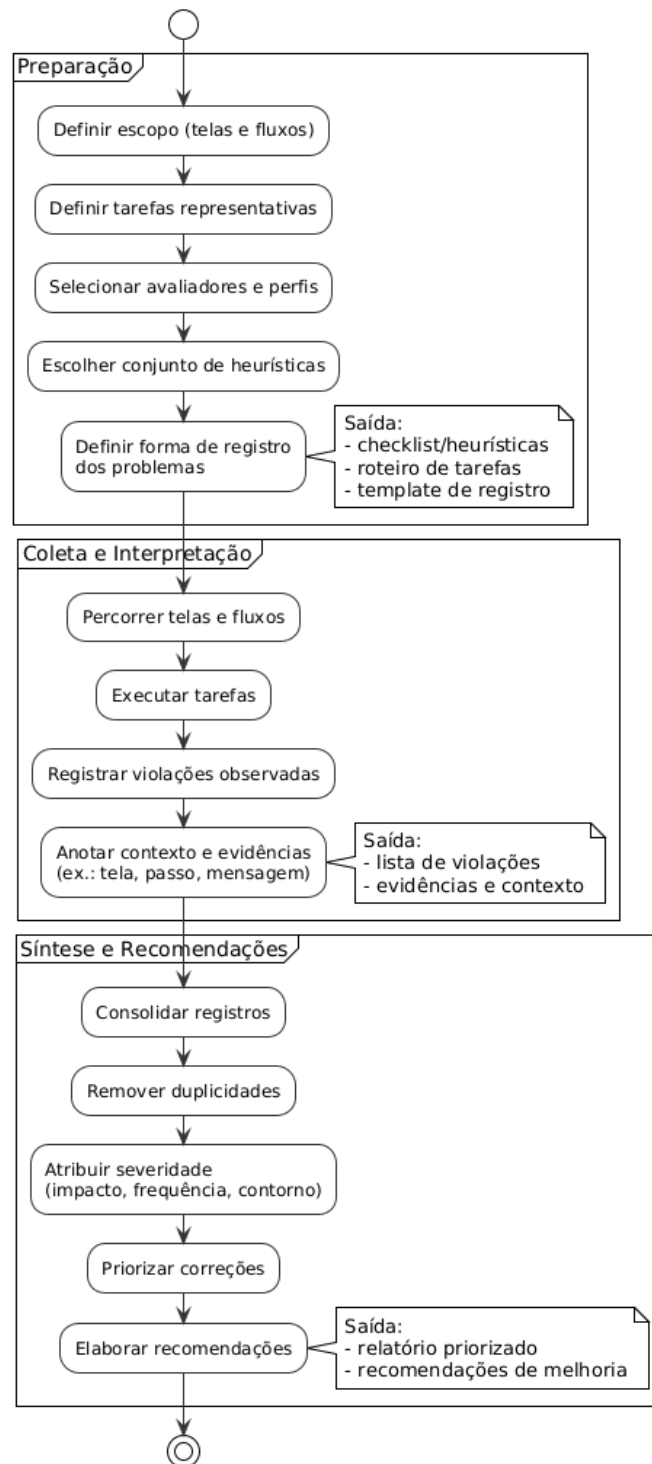


Figura 2.12: Etapas do método de avaliação heurística. Fonte: autoria própria.

Na etapa de preparação, define-se o escopo (telas e fluxos), o perfil dos avaliadores, o conjunto de heurísticas e a forma de registro dos problemas. Na coleta e interpretação, avaliadores executam tarefas representativas, registrando violações, contexto e evidências. Por fim, na síntese, as violações são consolidadas, removem-se duplicidades e atribui-se severidade, apoiando priorização de correções (NIELSEN, 1993).

Uma prática comum é classificar severidade considerando impacto na tarefa, frequência potencial e dificuldade de contorno, resultando em categorias como baixa, média e alta (ou escalas mais detalhadas quando necessário) (NIELSEN, 1993). Este trabalho adota uma classificação qualitativa para apoiar a discussão e a priorização de melhorias.

2.4 Diretrizes de Usabilidade para Dispositivos Móveis

Aplicações móveis operam sob restrições específicas: telas pequenas, entrada por toque, interrupções constantes e uso em movimento. Diretrizes de usabilidade móvel destacam a necessidade de minimizar carga cognitiva, reduzir passos em tarefas frequentes, garantir alvos de toque adequados e oferecer feedback rápido a cada ação (NIELSEN; BUDIUI, 2012; KRUG, 2014).

No *Pró-Inclusão*, essas diretrizes são críticas para fluxos com formulários extensos e para a exploração de mapa, pois o usuário precisa alternar entre visão geográfica e detalhes de registros. Além disso, a aplicação deve comunicar claramente permissões de localização e práticas de privacidade, uma vez que a geolocalização é parte central da experiência e envolve riscos se tratada de forma inadequada.

2.5 Sistemas de Informação Geográfica e Geolocalização Móvel

Sistemas de Informação Geográfica (SIG) combinam dados georreferenciados, análise espacial e visualizações cartográficas para apoiar decisões envolvendo localização, distância e distribuição espacial (LONGLY et al., 2015). Em aplicações móveis, a geolocalização integra sensores de posição, serviços de mapas e dados do domínio da aplicação para ordenar e filtrar informações por proximidade, região visível no mapa (*viewport*) e áreas de interesse.

Para garantir desempenho em consultas recorrentes (movimentos no mapa e alterações de *zoom*), utiliza-se indexação espacial e funções geográficas especializadas. Índices como R-Trees e variações restringem o conjunto de geometrias consideradas em

cada consulta, reduzindo custo computacional de operações de distância e interseção (LONGLEY et al., 2015; PROJECT, 2024).

No *Pró-Inclusão*, moradores em situação de rua e pedidos de ajuda são representados por pontos geográficos e recuperados por consultas que filtram resultados dentro da *viewport* atual ou ordenam registros por distância em relação à posição do usuário. Para isso, o *back-end* utiliza PostgreSQL com extensão PostGIS, que fornece tipos geométricos, funções espaciais e índices específicos para esse tipo de consulta (PROJECT, 2024).

2.6 Considerações Finais

A fundamentação teórica apresentada neste capítulo estabelece as bases para as decisões de *design* e para a avaliação do *Pró-Inclusão*. A noção de qualidade de uso, aliada às heurísticas de Nielsen e ao método de avaliação heurística, orienta a análise das interfaces. As diretrizes de usabilidade móvel e os fundamentos de geolocalização sustentam escolhas de interação e implementação discutidas nos capítulos seguintes.

3 Metodologia e Levantamento de Requisitos

Neste Capítulo são apresentadas a metodologia e as etapas de levantamento de requisitos do aplicativo *Pró-Inclusão*. Na Seção 3.1 é apresentada a Visão Geral da Metodologia adotada, baseada na Engenharia de Usabilidade. Na Seção 3.2 é definido o Escopo do Sistema, delimitando as funcionalidades centrais e os elementos fora do escopo. Na Seção 3.3 são apresentadas as Características dos Usuários identificados, com seus perfis e necessidades. Na Seção 3.4 são apresentados Cenários de Uso Representativos que ilustram situações típicas de operação do sistema. Na Seção 3.5 é apresentado o Levantamento de Requisitos, com a especificação de requisitos funcionais e não funcionais. Na Seção 3.6 é apresentada a Modelagem de Tarefas, que representa os fluxos de atividades dos usuários no sistema. Na Seção 3.7 é apresentado o Design de IHC Aplicado, descrevendo as diretrizes de Interação Humano-Computador seguidas no projeto das interfaces. Finalmente, na Seção 3.8 são apresentadas as considerações finais deste Capítulo. No Capítulo 4 são descritas a arquitetura, a implementação e as telas do sistema *Pró-Inclusão*.

3.1 Visão Geral da Metodologia

A metodologia utilizada neste trabalho baseia-se nos princípios da Engenharia de Usabilidade, que organiza o processo de desenvolvimento em etapas sistemáticas voltadas à compreensão de usuários, atividades e contexto de uso (NIELSEN, 1993; PREECE; ROGERS; SHARP, 2015). Essa abordagem busca reduzir retrabalho, garantir aderência às necessidades reais e fundamentar decisões de *design* em evidências empíricas.

O processo aplicado pode ser resumido nas seguintes etapas:

1. **Levantamento do contexto:** identificação do problema social, mapeamento do fluxo de atuação dos voluntários e análise da dinâmica de pedidos de ajuda.
2. **Definição do escopo e metas de *design*:** delimitação das funcionalidades essenciais e critérios mínimos de usabilidade.

3. **Caracterização dos usuários:** identificação de perfis de uso, capacidades, limitações e necessidades.
4. **Levantamento e especificação de requisitos:** definição de Requisitos Funcionais (RF) e Não Funcionais (RNF).
5. **Modelagem de tarefas:** representação das atividades e fluxos realizados pelos usuários no sistema.
6. **Implementação e Prototipação Evolutiva:** construção direta das interfaces de alta fidelidade para validação funcional.
7. **Avaliação:** testes funcionais e avaliação heurística, descritos no Capítulo 5.

O uso de React Native/Expo permitiu ciclos curtos de entrega, nos quais *design* visual e implementação ocorreram de forma integrada. As interações foram testadas diretamente em dispositivos móveis a cada incremento, reduzindo a necessidade de *wireframes* estáticos e favorecendo ajustes contínuos baseados em princípios de usabilidade.

A Figura 3.1 apresenta o fluxograma da metodologia adotada, destacando o encadeamento entre análise e especificação (contexto e requisitos), modelagem de tarefas (HTA), e o ciclo de prototipação evolutiva que integra desenvolvimento de *back-end* e *front-end*.

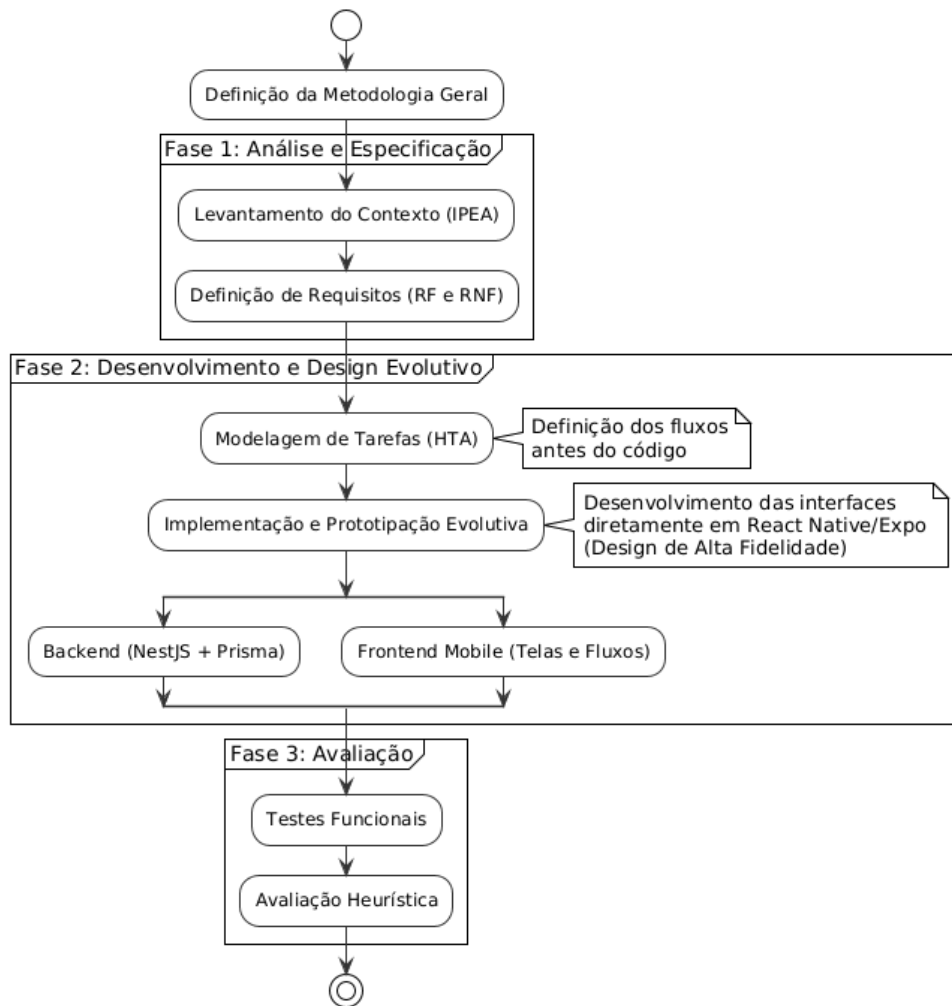


Figura 3.1: Fluxograma da metodologia com ciclos de implementação e prototipação evolutiva.

3.2 Escopo do Sistema

O sistema *Pró-Inclusão* foi projetado para apoiar voluntários e cidadãos em ações sociais relacionadas a pessoas em situação de rua. O escopo definido inclui as seguintes capacidades centrais:

- registro estruturado de moradores com dados pessoais, histórico, familiares, localizações e serviços sociais;
- criação e gerenciamento de pedidos de ajuda associados a moradores cadastrados;
- oferta de ajuda por outros usuários, com acompanhamento do status;
- ordenação geográfica das informações com base na posição do usuário;

- autenticação unificada e controle de permissões por papéis;
- integração com serviços externos como Google Places e armazenamento S3.

Elementos fora do escopo incluem: automação de políticas públicas, integração com bases governamentais e serviços de emergência, análises preditivas e sistemas de recomendação avançados.

3.3 Características dos Usuários

Os usuários identificados foram classificados em dois perfis principais, com necessidades e capacidades distintas. A administração desses perfis (por exemplo, a promoção de um usuário básico para voluntário) é realizada externamente, por meio do painel do serviço de autenticação Clerk, não havendo um papel de administrador dentro do próprio aplicativo móvel.

3.3.1 Usuário Básico

Usuários comuns que desejam explorar pedidos de ajuda próximos e oferecer auxílio quando possível. Frequentemente possuem pouca experiência com ferramentas de gestão, necessitando interfaces simples e diretas.

3.3.2 Voluntário

Usuários responsáveis por cadastrar moradores, criar pedidos de ajuda e acompanhar atendimentos. Possuem papel ativo no fluxo de trabalho e demandam mais funcionalidades do que usuários básicos.

3.4 Cenários de Uso Representativos

Antes do detalhamento dos requisitos, foram construídos cenários narrativos que exemplificam situações típicas de uso do sistema. Esses cenários auxiliaram na identificação de necessidades, na validação do escopo e na priorização de funcionalidades, alinhando-se às

recomendações de levantamento de requisitos orientado a tarefas (PREECE; ROGERS; SHARP, 2015).

No primeiro cenário, considera-se Maria, voluntária vinculada a uma organização social que realiza atendimentos semanais em uma região central. Durante uma abordagem, Maria identifica um novo morador em situação de rua, coleta informações sobre sua trajetória, registra a existência de familiares e mapeia locais em que ele costuma permanecer. Sem o uso do sistema, essas informações seriam anotadas em papel ou em registros pessoais, com alto risco de perda e dificuldade de compartilhamento com outros voluntários. Com o *Pró-Inclusão*, Maria autentica-se no aplicativo, preenche as etapas do cadastro de morador, salva localizações relevantes no mapa, associa serviços sociais já utilizados e, ao final, registra um pedido de ajuda específico (por exemplo, itens de higiene e cobertores) vinculado ao perfil do morador.

No segundo cenário, considera-se João, usuário básico que deseja contribuir com ações sociais em seu trajeto diário. Ao abrir o aplicativo, João permite o compartilhamento de sua localização e visualiza, em uma lista ordenada por distância, pedidos de ajuda abertos na região. Ele seleciona um pedido, verifica detalhes sobre o morador e o tipo de auxílio solicitado e decide ofertar ajuda. O sistema registra a oferta, impede que João faça ofertas duplicadas para o mesmo pedido e atualiza as visualizações do criador do pedido, que poderá aceitar, rejeitar ou encerrar a oferta após a conclusão da ajuda. Esses cenários ilustram como os requisitos funcionais derivam de situações concretas de uso e de necessidades identificadas junto aos perfis de usuários.

3.5 Levantamento de Requisitos

A partir da análise do contexto, atividades dos usuários e fluxos identificados, foram definidos os Requisitos Funcionais (RF) e Não Funcionais (RNF) do sistema.

3.5.1 Requisitos Funcionais

Tabela 3.1: Requisitos funcionais do sistema *Pró-Inclusão*

ID	Descrição	Prioridade
RF01	Permitir autenticação via Google/Clerk.	Essencial
RF02	Sincronizar dados de perfil do usuário com o serviço Clerk.	Importante
RF03	Exigir perfil completo para permitir ofertas de ajuda.	Importante
RF04	Cadastrar e atualizar perfis de moradores em situação de rua.	Essencial
RF05	Registrar dados pessoais, histórico, serviços sociais, familiares e foto do morador.	Essencial
RF06	Registrar múltiplas localizações georreferenciadas do morador.	Importante
RF07	Armazenar consentimento em áudio ou vídeo para criação de perfis.	Importante
RF08	Criar pedidos de ajuda associados a moradores.	Essencial
RF09	Permitir cancelamento ou atualização de pedidos de ajuda.	Importante
RF10	Listar pedidos de ajuda próximos ao usuário.	Essencial
RF11	Ordenar pedidos por distância, prioridade e prazo.	Desejável
RF12	Oferecer ajuda a pedidos abertos.	Essencial
RF13	Bloquear ofertas duplicadas para o mesmo pedido.	Importante
RF14	Permitir aceitar ou rejeitar ofertas feitas.	Essencial
RF15	Encerrar ofertas aceitas com conclusão ou cancelamento.	Importante
RF16	Exibir listas de apoio como tipos de ajuda, parentesco, escolaridade etc.	Importante
RF17	Integrar com API Google Places para validação de locais e <i>viewports</i> .	Desejável
RF18	Armazenar mídias em S3/MinIO.	Importante
RF19	Exibir mapa com marcadores de moradores.	Importante
RF20	Realizar buscas geoespaciais com PostGIS.	Importante
RF21	Exibir detalhes completos de moradores, pedidos e ofertas.	Importante
RF22	Enviar localização contínua do usuário para ordenação dos dados.	Importante

3.5.2 Especificação de Casos de Uso Selecionados

Alguns requisitos funcionais apresentam maior complexidade de fluxo e impactam diretamente a experiência dos usuários. Nesta subseção são detalhados dois casos de uso representativos, associados aos requisitos RF08 (Criar pedidos de ajuda associados a moradores) e RF12 (Oferecer ajuda a pedidos abertos), descrevendo atores, pré-condições, fluxo principal, fluxos alternativos e pós-condições.

Tabela 3.3: Especificação do caso de uso associado ao requisito RF08 - Criar pedido de ajuda.

Item	Descrição
Identificador	RF08 - Criar pedido de ajuda associado a morador.
Atores	Voluntário autenticado.
Pré-condições	(i) Usuário autenticado como voluntário; (ii) Morador previamente cadastrado e selecionado na interface.

Tabela 3.5: Especificação do caso de uso associado ao requisito RF12 - Oferecer ajuda a pedido aberto.

Item	Descrição
Identificador	RF12 - Oferecer ajuda a pedido aberto.
Atores	Usuário autenticado (perfil básico ou voluntário).
Pré-condições	(i) Usuário autenticado; (ii) Pedido de ajuda em estado “aberto”; (iii) Usuário não ser o criador do pedido; (iv) Inexistência de oferta ativa anterior do mesmo usuário para o mesmo pedido.
Fluxo principal	1. O usuário acessa a lista de pedidos disponíveis; 2. Seleciona um pedido de interesse e visualiza seus detalhes; 3. Aciona a opção “Oferecer ajuda”; 4. Informa, quando necessário, observações complementares ou condições para o auxílio; 5. Confirma a oferta; 6. O sistema verifica restrições (duplicidade, estado do pedido e identidade do criador); 7. O sistema registra a oferta em estado “pendente” para avaliação do criador do pedido.
Fluxos alternativos	3a. Pedido encerrado entre a listagem e a oferta: o sistema informa que o pedido não aceita mais ofertas e sugere atualização da lista; 6a. Oferta duplicada detectada: o sistema impede o registro e exibe mensagem explicando a restrição; 6b. Criador tenta ofertar ao próprio pedido: o sistema bloqueia a ação e apresenta mensagem de erro.
Pós-condições	(i) Oferta registrada e associada ao pedido e ao usuário ofertante; (ii) Pedido passa a exibir a nova oferta na visão do criador, que poderá aceitá-la, rejeitá-la ou aguardar outras ofertas.

3.5.3 Requisitos Não Funcionais

Tabela 3.7: Requisitos não funcionais do sistema *Pró-Inclusão*

ID	Descrição
RNF01	O sistema deve apresentar resposta em até 3 segundos para consultas comuns.
RNF02	Devem ser utilizados protocolos seguros para autenticação e transferência de dados.
RNF03	As APIs devem seguir arquitetura REST.
RNF04	O aplicativo deve funcionar adequadamente em redes móveis instáveis.
RNF05	A interface deve seguir diretrizes de acessibilidade.
RNF06	O sistema deve registrar logs para fins de auditoria.

3.6 Modelagem de Tarefas

A modelagem de tarefas é utilizada para organizar e compreender as ações realizadas pelos usuários, buscando explicitar objetivos, subtarefas e dependências entre atividades. Neste trabalho, tarefas principais foram representadas por meio de Análise Hierárquica de Tarefas (HTA), técnica amplamente discutida na literatura de IHC (DIX et al., 2004). As Figuras 3.2, 3.3, 3.4 e 3.5 detalham, respectivamente, os passos essenciais para cadastrar um morador, criar um pedido, ofertar ajuda e explorar o mapa. Cada diagrama explicita o que é etapa obrigatória, o que é decisão do usuário e onde o sistema precisa oferecer feedback e prevenção de erros.

Exemplos de tarefas modeladas são apresentados a seguir, com suas respectivas representações em HTA.

Cadastrar morador

O diagrama da Figura 3.2 organiza as sete macroetapas do cadastro, deixando evidente a sequência esperada (dados pessoais, situação de rua, localização, rede de apoio e consentimento) e apoiando a definição de validações e mensagens de estado ao longo do fluxo.

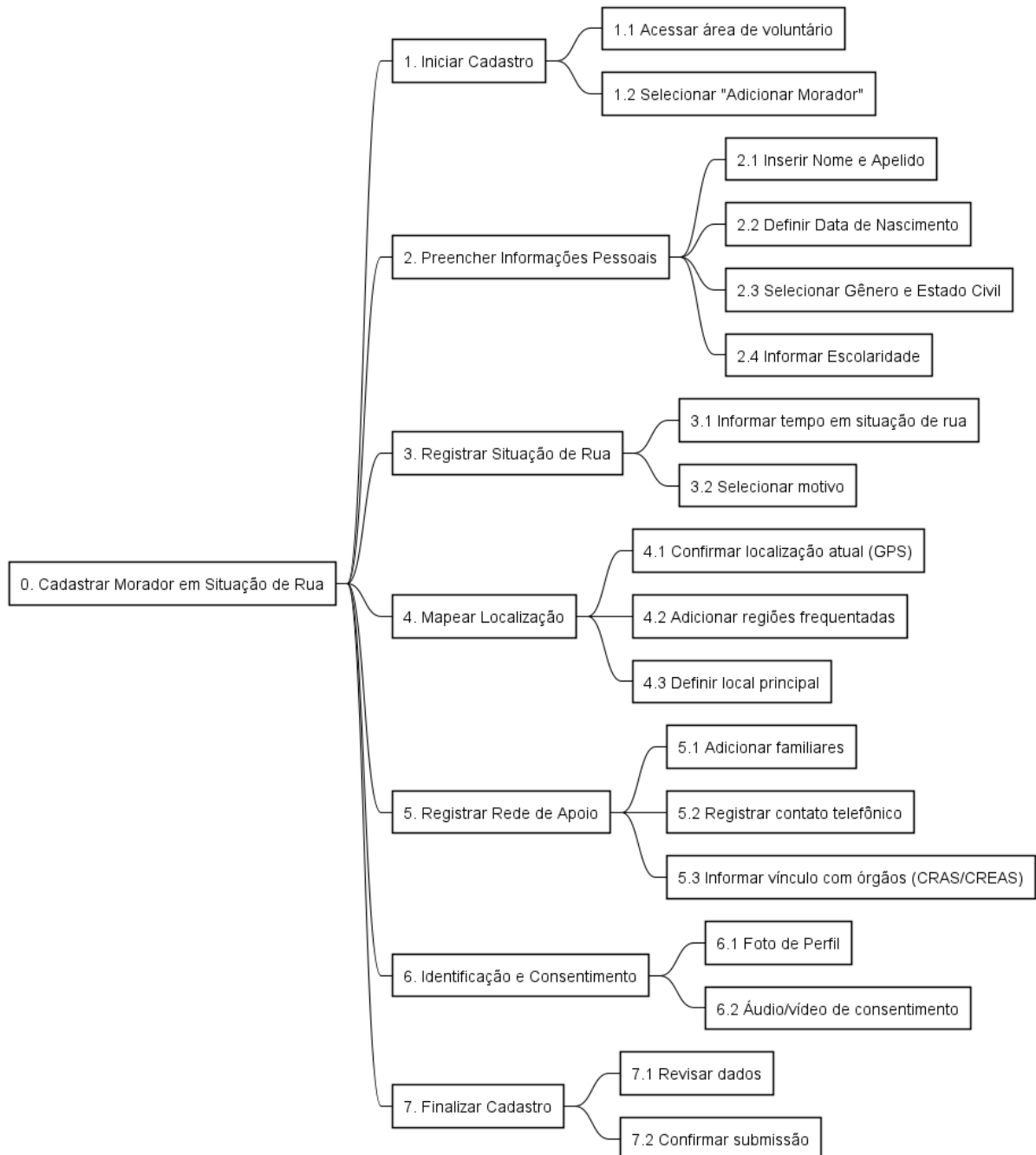


Figura 3.2: Análise Hierárquica de Tarefas para cadastrar morador.

Criar pedido de ajuda

A Figura 3.3 descreve a construção de um pedido a partir de um morador vinculado, enfatizando decisões de prioridade, prazo e logística (local de entrega), que impactam diretamente a ordenação por distância e o acompanhamento de ofertas.

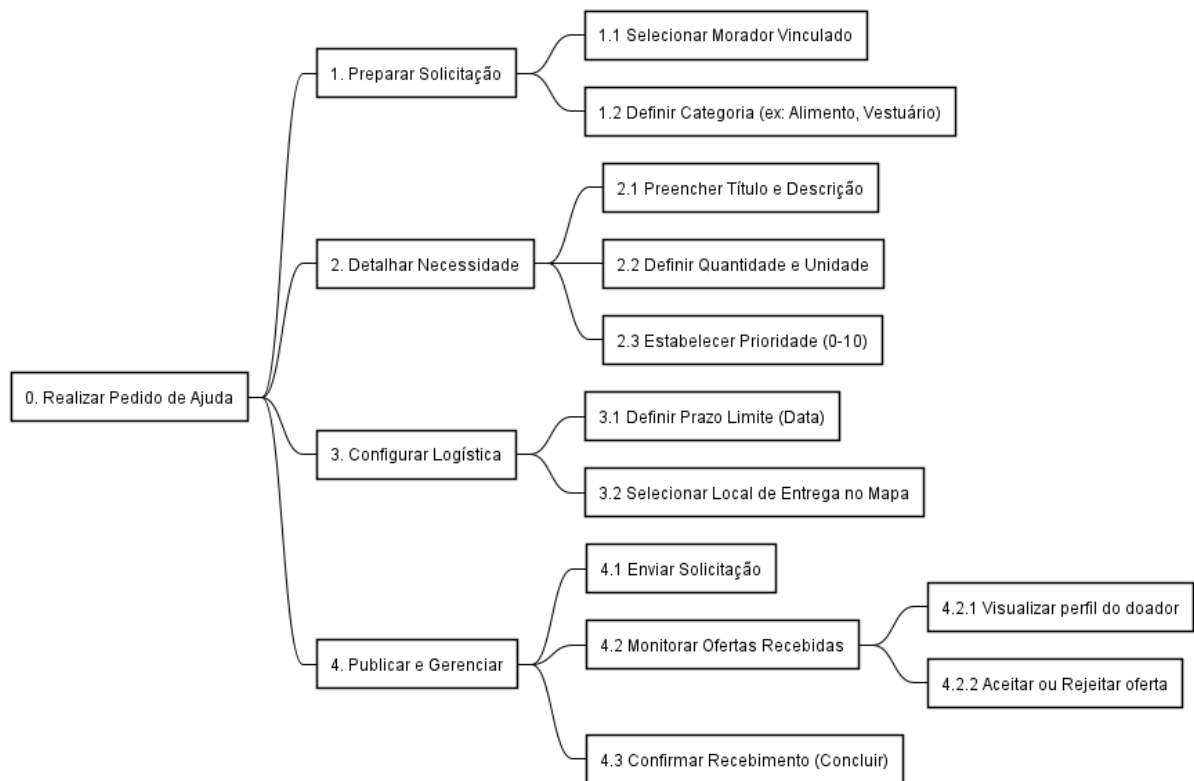


Figura 3.3: Análise Hierárquica de Tarefas para criar pedido de ajuda.

Ofertar ajuda

A Figura 3.4 modela o percurso do usuário ao selecionar um pedido e registrar uma oferta, destacando pontos em que o sistema deve prevenir duplicidade e garantir clareza de status (pendente, aceita, rejeitada).



Figura 3.4: Análise Hierárquica de Tarefas para ofertar ajuda.

Explorar mapa

A Figura 3.5 evidencia o ciclo recorrente de navegação no mapa (movimentar, filtrar, selecionar e abrir detalhes), reforçando a necessidade de desempenho em consultas por *viewport* e feedback rápido ao atualizar marcadores.

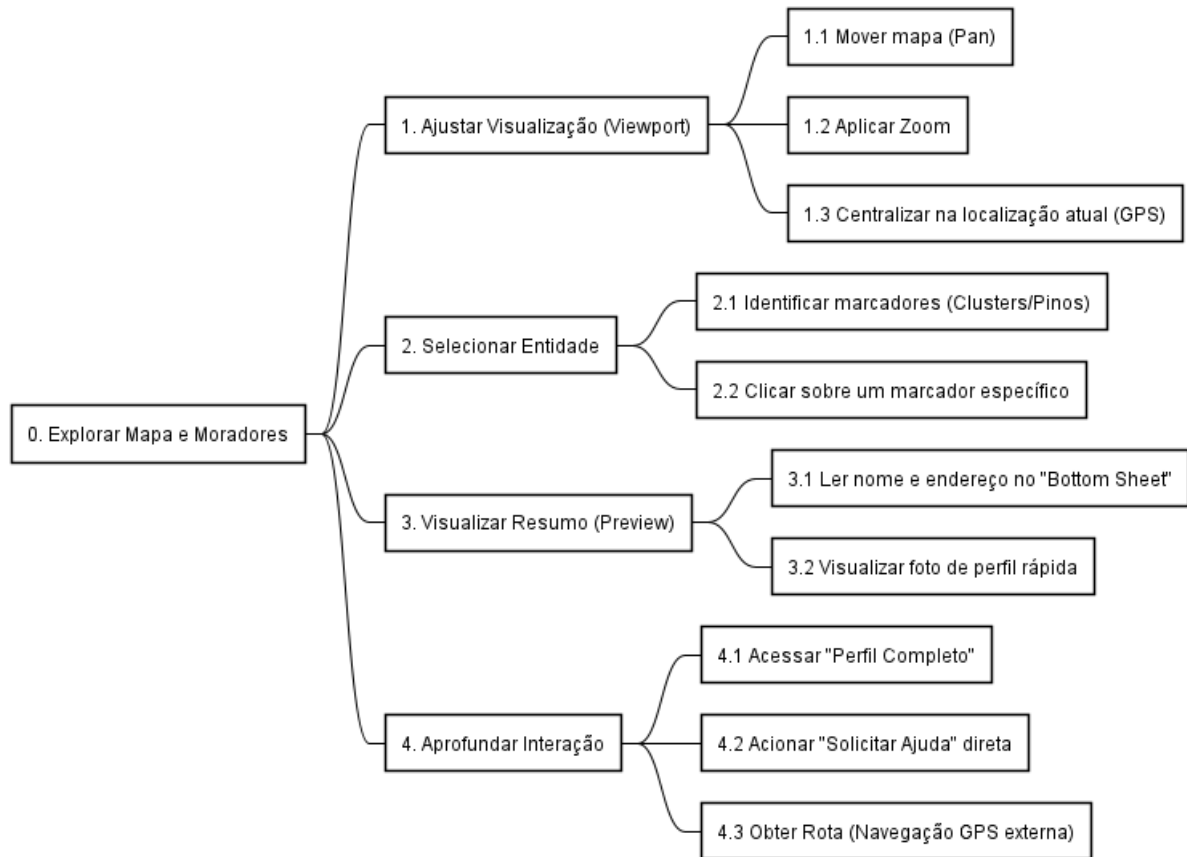


Figura 3.5: Análise Hierárquica de Tarefas para explorar o mapa.

3.7 Design de IHC Aplicado

O Design de IHC foi conduzido como uma prototipação evolutiva de alta fidelidade. A agilidade do React Native/Expo permitiu desenvolver diretamente protótipos funcionais, eliminando a necessidade de *wireframes* estáticos e validando fluxos reais em dispositivos. Cada incremento de código foi revisado à luz das heurísticas de usabilidade e diretrizes de uso em dispositivos móveis, priorizando visibilidade do estado do sistema, consistência e prevenção de erros durante a implementação (NIELSEN, 1993; NIELSEN; BUDI, 2012).

As principais práticas adotadas foram:

- ciclos curtos de implementação com testes imediatos em aparelho físico ou simulador;

- revisões heurísticas a cada incremento para verificar visibilidade do estado do sistema, consistência e prevenção de erros;
- padronização de componentes e linguagem para reduzir esforço cognitivo e aumentar comunicabilidade;
- reuso de componentes e *tokens* de estilo para manter alinhamento visual entre fluxos de moradores, pedidos e ofertas;
- mapeamento do fluxo de navegação diretamente no código de rotas, reduzindo transições desnecessárias e passos cognitivos.

As Figuras 3.6 e 3.7 apresentam exemplos das interfaces de alta fidelidade utilizadas como artefatos de *design* durante o ciclo de desenvolvimento.

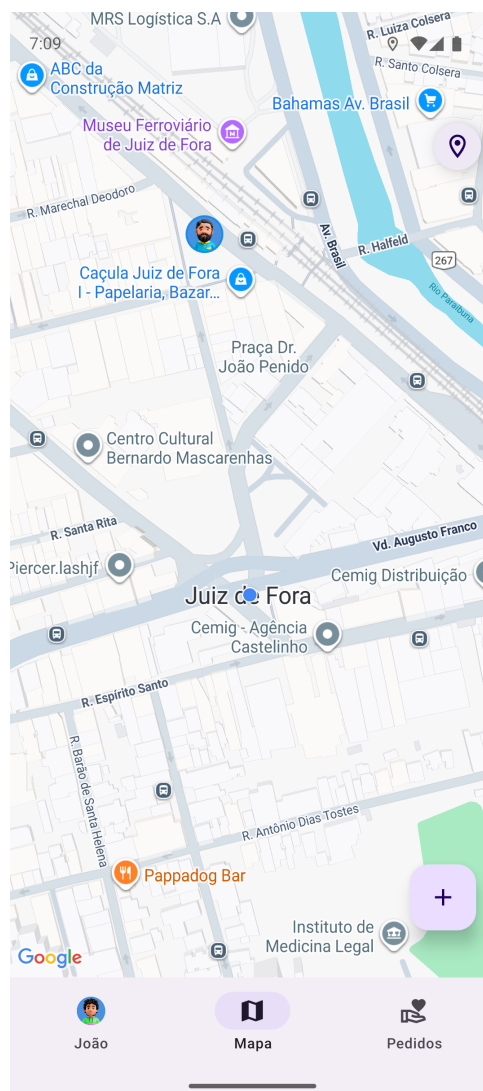
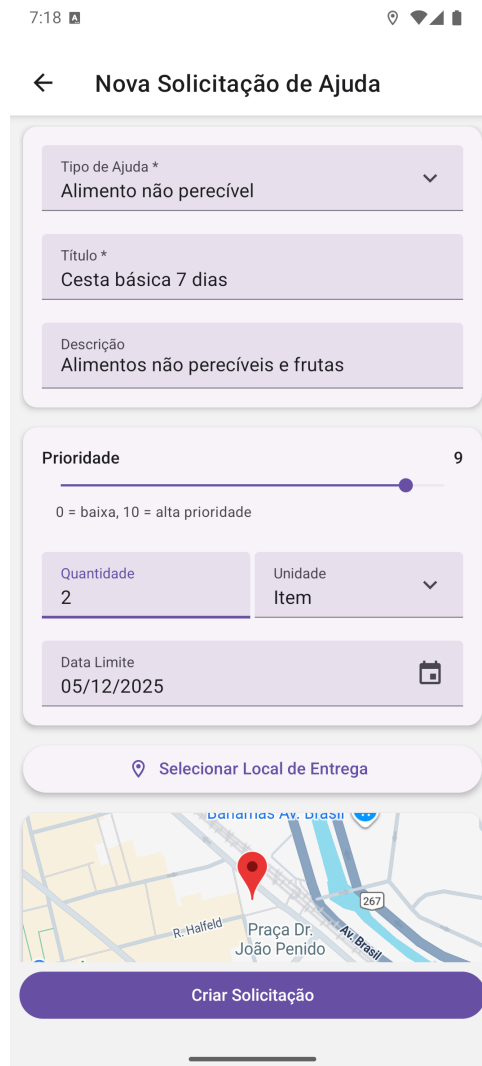


Figura 3.6: Interface implementada da navegação principal do aplicativo.



The screenshot shows a mobile application interface for creating a request for help. At the top, the status bar displays the time 7:18 and various system icons. The app title is "Nova Solicitação de Ajuda". The form includes the following fields and controls:

- Tipo de Ajuda ***: A dropdown menu with the selected option "Alimento não perecível".
- Título ***: A text input field containing "Cesta básica 7 dias".
- Descrição**: A text input field containing "Alimentos não perecíveis e frutas".
- Prioridade**: A slider control set to 9, with a scale from 0 (baixa) to 10 (alta prioridade).
- Quantidade**: A text input field containing "2".
- Unidade**: A dropdown menu with the selected option "Item".
- Data Limite**: A date picker showing "05/12/2025".
- Selecionar Local de Entrega**: A button with a location pin icon.
- Mapa**: A map showing the selected location at "Praça Dr. João Penido" near "R. Halfeld" and "Av. Brasil".
- Botão de Ação**: A large purple button labeled "Criar Solicitação".

Figura 3.7: Interface implementada para criação de pedidos de ajuda.

3.8 Considerações Finais

A metodologia apresentada neste capítulo guiou todas as fases de desenvolvimento da plataforma *Pró-Inclusão*, garantindo alinhamento entre objetivos, necessidades dos usuários e decisões de interface. Os requisitos definidos servirão de base para a implementação descrita no capítulo seguinte.

4 Arquitetura, Implementação e Telas do Sistema Pró-Inclusão

Neste Capítulo são descritos a arquitetura, a implementação e as telas do sistema *Pró-Inclusão*. Na Seção 4.1 é apresentada a Arquitetura Geral da Plataforma, com seus componentes principais e integrações externas. Na Seção 4.2 é apresentada a Modelagem de Dados, com as entidades e relações do banco de dados. Na Seção 4.3 são descritos a estrutura e os fluxos do *Back-end*, detalhando os módulos da API. Na Seção 4.4 são apresentados Trechos de Código Seleccionados que ilustram decisões técnicas relevantes da implementação. Na Seção 4.5 são descritas a estrutura e as funcionalidades do Aplicativo Móvel, com a apresentação das telas desenvolvidas. Finalmente, na Seção 4.6 são apresentadas as considerações finais deste Capítulo. No Capítulo 5 é apresentada a avaliação do sistema *Pró-Inclusão*.

4.1 Arquitetura Geral da Plataforma

A plataforma é composta por dois módulos principais: a API *web* (*backend*) e o aplicativo móvel (*frontend*). Ambos se comunicam por meio de requisições REST seguras, com autenticação realizada via Clerk, seguindo recomendações de arquiteturas orientadas a recursos (FIELDING, 2000).

A Figura 4.1 apresenta a visão geral da arquitetura e explicita como os principais componentes se conectam: autenticação (Clerk), integrações externas (Google Places), persistência (PostgreSQL/PostGIS) e armazenamento de mídias (MinIO/S3). As setas representam o fluxo de autenticação (obtenção/validação de *token*) e o fluxo de dados (requisições REST e consultas/persistência), deixando claro onde ocorrem validações e onde são realizadas consultas geoespaciais.

Os principais componentes são:

- **API Back-end:** implementada em NestJS, responsável por regras de negócio,

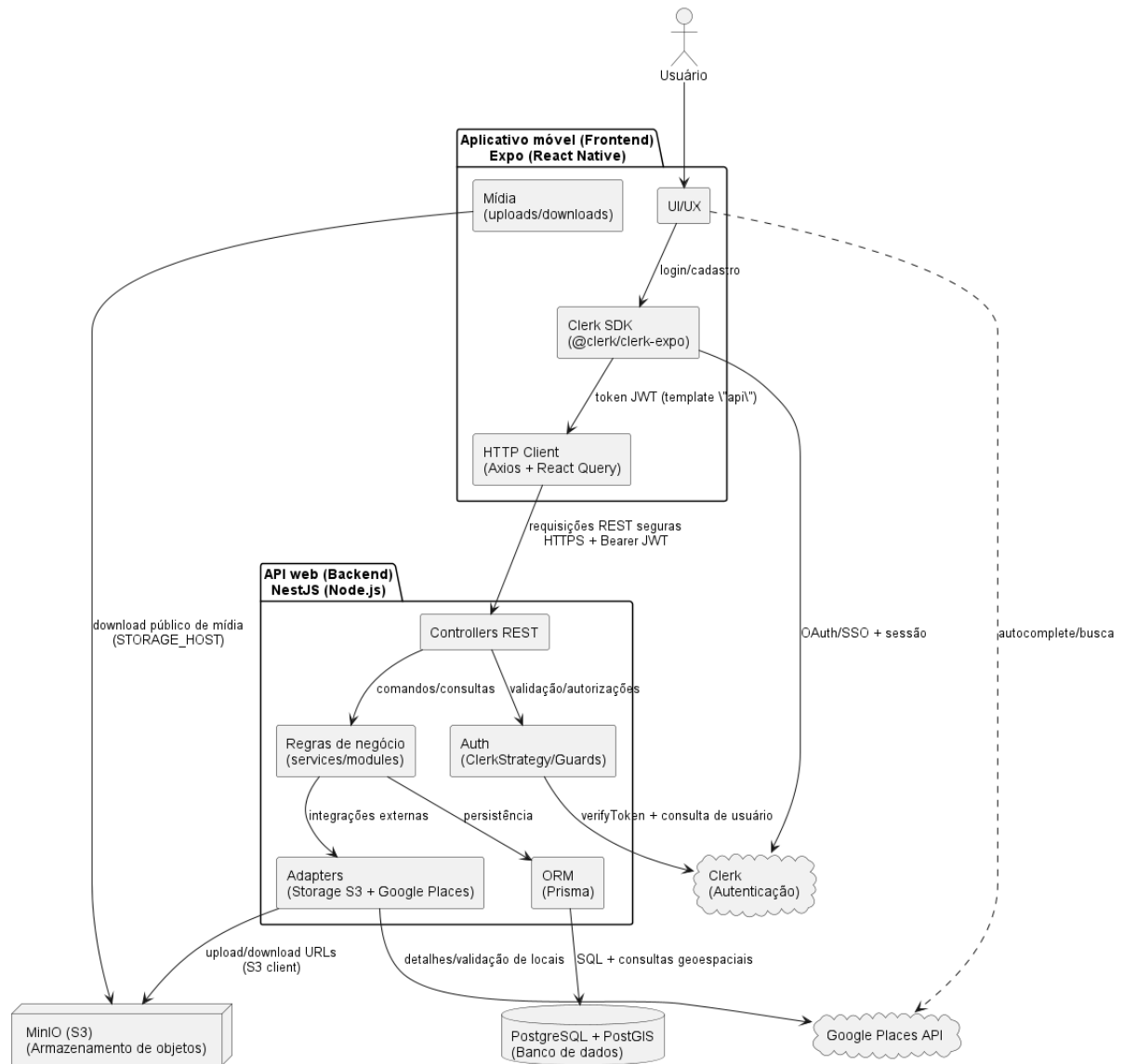


Figura 4.1: Visão geral da arquitetura da plataforma *Pró-Inclusão*. Fonte: autoria própria.

persistência, autenticação e integrações externas.

- **Banco de Dados:** PostgreSQL com extensão PostGIS para suporte a consultas geoespaciais.
- **Armazenamento de Mídias:** MinIO/S3 para fotos, vídeos e gravações de consentimento.
- **Aplicativo Móvel:** desenvolvido em Expo/React Native, com autenticação, mapa interativo e fluxos de cadastro.
- **Integrações Externas:** Google Places API para validação e enriquecimento de endereços.

4.2 Modelagem de Dados

O modelo de dados foi implementado com Prisma ORM, que descreve entidades como moradores, localizações, familiares, pedidos de ajuda e ofertas. As entidades principais incluem:

- **Homeless**: dados pessoais, histórico, foto, serviços sociais.
- **HomelessLocation**: armazena múltiplas localizações georreferenciadas (geom).
- **HomelessFamilyMember**: familiares vinculados ao morador.
- **HelpRequest**: pedidos de ajuda associados a um morador.
- **HelpOffer**: ofertas feitas pelos usuários.
- **MediaAsset**: chave e URL de mídias armazenadas.
- **GooglePlace**: *cache* de resultados do Google Places.
- **AppConfig**: parâmetros de configuração do app.

A extensão PostGIS foi utilizada para consultas de *viewport*, ordenação por distância e geração de índices geoespaciais.

A Figura 4.2 resume o modelo entidade-relacionamento, permitindo visualizar as associações centrais do domínio: um morador pode possuir múltiplas localizações e registros complementares (familiares, serviços sociais e mídias), enquanto pedidos de ajuda se relacionam ao morador e recebem ofertas de usuários. Esse diagrama orienta a leitura das seções de implementação ao conectar rotas/serviços do *back-end* às entidades persistidas no banco.

4.3 Back-end - Estrutura e Fluxos

O *back-end* foi organizado em módulos do NestJS, cada um isolando responsabilidades.

4.3.1 Autenticação e Autorização

A autenticação é realizada via Clerk, com *tokens* verificados pelo *back-end*. Há dois papéis principais expostos no aplicativo:

- **basic**: pode explorar pedidos e fazer ofertas.
- **volunteer**: pode cadastrar e editar moradores e criar pedidos de ajuda.

Guards específicos asseguram que cada *endpoint* seja acessado apenas por usuários autorizados. A mudança de perfil entre *basic* e *volunteer* é realizada no painel administrativo do próprio Clerk, não havendo uma interface de administração interna ao *Pró-Inclusão* para esse tipo de operação.

4.3.2 Módulo de Usuários

Responsável por sincronizar dados do usuário com o Clerk, exigir perfil completo e expor *endpoints* para consulta do próprio perfil.

4.3.3 Módulo de Moradores

Permite criar ou atualizar perfis completos de moradores, incluindo:

- dados pessoais;
- histórico e motivo da situação de rua;
- serviços sociais (CRAS, CREAS, albergues);
- familiares;
- foto de perfil;
- consentimento em áudio ou vídeo;

- localizações georreferenciadas.

Permite também a busca geoespacial via *viewport*.

4.3.4 Módulo de Pedidos de Ajuda

Voluntários criam pedidos contendo tipo, quantidade, prioridade, localização de entrega e prazo. A listagem geral retorna apenas pedidos abertos de terceiros.

4.3.5 Módulo de Ofertas de Ajuda

Usuários podem ofertar ajuda respeitando regras de duplicidade e status. O criador do pedido aceita ou rejeita ofertas.

4.3.6 Módulo de Listas Gerais

Endpoint genérico que preenche listas como tipos de ajuda, escolaridade, motivos, parentesco e estado civil.

4.4 Trechos de Código Seleccionados

Esta seção apresenta trechos de código representativos da lógica implementada no *back-end* e no aplicativo móvel. Os exemplos são extraídos e simplificados a partir da implementação real, com o objetivo de tornar explícitas decisões de engenharia relacionadas a geolocalização, regras de negócio e captura contínua de localização.

4.4.1 Consulta Geoespacial com Prisma e PostGIS

O Código 4.1 ilustra uma consulta que recupera moradores dentro da *viewport* atual do mapa, utilizando tipos espaciais e índices fornecidos pelo PostGIS.

```
1 SELECT hl."homelessId" FROM "HomelessLocation" hl
2 WHERE hl."isMainLocation" = true
3 AND ST_Within(
4     hl.geom,
```

```
5     ST_MakeEnvelope({west}, {south}, {east}, {north}, 4326)
6 );
```

Listing 4.1: Consulta geoespacial simplificada para carregar ids de moradores por *viewport*.

Na implementação com Prisma e NestJS, essa consulta é encapsulada em um serviço que recebe os limites da *viewport* (latitudes e longitudes mínimas e máximas) e retorna uma lista de moradores para exibição no mapa. O uso de funções espaciais como *ST_Within* em conjunto com envelopes retangulares permite aproveitar o índice espacial do PostGIS, reduzindo o custo de busca em grandes conjuntos de dados (PROJECT, 2024; LONGLEY et al., 2015).

4.4.2 Regra de Negócio para Bloqueio de Ofertas Duplicadas

O Código 4.2 representa, de forma simplificada, a lógica no serviço de ofertas que impede que um mesmo usuário crie duas ofertas ativas para o mesmo pedido de ajuda.

```
1  async create(
2    data: HelpOfferCreateInput,
3    currentUser: User,
4    location?: Point,
5  ) {
6    await this.userService.checkCompletedProfile(currentUser.id);
7
8    const helpRequest = await
9      this.prisma.helpRequest.findUnique({
10     where: { id: data.requestId },
11     select: { id: true, createdById: true, status: true },
12   });
13
14   const openOffer = await this.prisma.helpOffer.count({
15     where: {
16       requestId: data.requestId,
17       userId: currentUser.id,
```

```
17     status: {
18         in: [HelpOfferStatus.PLEDGED,
19             HelpOfferStatus.ACCEPTED],
20     },
21 };
22 if (openOffer > 0)
23     throw new ConflictException(
24         'Você já possui uma oferta de ajuda aberta para esta
25         solicitação.',
26     );
27 if (!helpRequest)
28     throw new NotFoundException('Solicitação de ajuda não
29     encontrada.');
```

```
30 if (helpRequest.createdById === currentUser.id)
31     throw new BadRequestException(
32         'Não é possível oferecer ajuda para sua própria
33         solicitação.',
34     );
35 if (helpRequest.status !== HelpRequestStatus.OPEN)
36     throw new BadRequestException(
37         'Solicitação de ajuda não está aberta para ofertas.',
38     );
39
40 return this.prisma.helpOffer.create({
41     data: {
42         requestId: data.requestId,
43         userId: currentUser.id,
44         quantityOffered: data.quantityOffered ?? null,
```

```
45     message: data.message ?? null,
46     createdLatitude: location?.latitude ?? null,
47     createdLongitude: location?.longitude ?? null,
48   },
49 );
50 }
```

Listing 4.2: Trecho simplificado de serviço NestJS para bloquear ofertas duplicadas.

Esse trecho evidencia a separação entre a camada de serviço e a camada de acesso a dados, bem como a aplicação explícita da regra de negócio que restringe ofertas duplicadas (RF13). Ao consultar ofertas com status *PLEDGED* ou *ACCEPTED* antes da criação, o sistema garante integridade do fluxo de ajuda e evita inconsistências na interface do usuário. Na interface móvel, a mensagem configurada na exceção de conflito é apresentada ao usuário quando ele tenta oferecer ajuda mais de uma vez para a mesma solicitação, conectando diretamente a validação de *back-end* ao fluxo de erro nas telas de pedidos de ajuda.

4.4.3 Captura Contínua de Localização no Aplicativo Móvel

No aplicativo móvel, a captura contínua da localização é utilizada para ordenar pedidos de ajuda por proximidade e atualizar o mapa conforme o usuário se desloca. O Código 4.3 apresenta um *hook* simplificado em React Native/Expo que solicita permissão, obtém a posição atual e registra atualizações periódicas.

```
1 export const useWatchPosition = () => {
2   const { coords, setCoords } = usePositionStore()
3
4   const header = coords ?
5     ‘${coords.latitude},${coords.longitude}‘ : undefined
6   useEffect(() => {
7     if (!header) return
8     const interceptorId = http.interceptors.request.use(config
9       => {
```

```
8     config.headers['x-user-location'] = header
9     return config
10  })
11
12  return () => {
13    http.interceptors.request.eject(interceptorId)
14  }
15  }, [header])
16
17  useEffect(() => {
18    let subscription: Location.LocationSubscription
19    Location.requestForegroundPermissionsAsync().then(async ({
20      status }) => {
21      if (status !== 'granted') return
22
23      subscription = await Location.watchPositionAsync(
24        {
25          accuracy: Location.Accuracy.High,
26          timeInterval: 2000,
27          distanceInterval: 1
28        },
29        pos => setCoords(pos.coords)
30      )
31    })
32
33    return () => subscription?.remove()
34  }, [])
35 }
```

Listing 4.3: Hook simplificado para monitoramento contínuo de localização no aplicativo.

Esse código ilustra a preocupação em solicitar permissões explicitamente, lidar com possíveis recusas de acesso e limitar a frequência de atualizações por meio de um

distanceInterval, o que é coerente com recomendações de usabilidade e consumo consciente de recursos em dispositivos móveis (NIELSEN; BUDIU, 2012).

4.5 Aplicativo Móvel - Estrutura e Funcionalidades

O aplicativo móvel utiliza Expo/React Native, com navegação em abas, mapas e formulários distribuídos em múltiplos passos.

Para tornar explícitos os caminhos de navegação e os pontos de decisão, foram elaborados diagramas de *screenflow* (PlantUML) que descrevem as telas e transições principais do aplicativo. Esses diagramas ajudam a explicar onde o sistema precisa manter consistência de nomenclatura, oferecer feedback de estado e prevenir erros (por exemplo, ao finalizar cadastro ou ao alternar entre listas e detalhes).

A Figura 4.3 descreve o fluxo de autenticação e o encaminhamento para a tela principal, incluindo a etapa de finalização de cadastro.

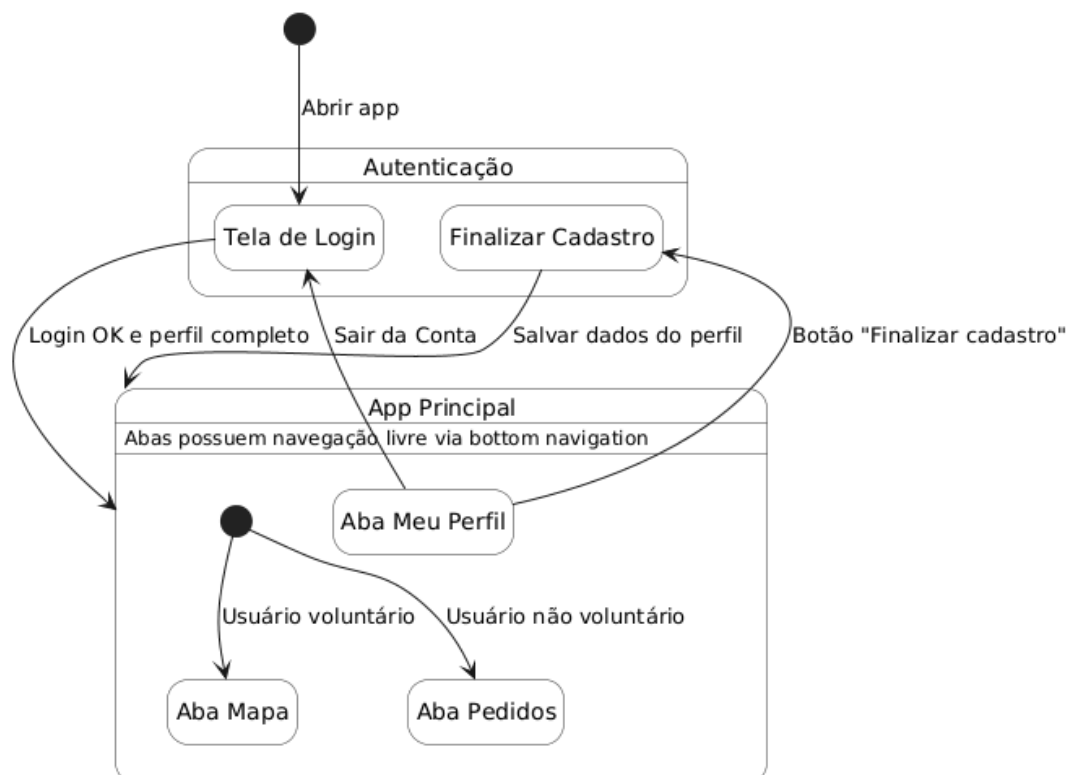


Figura 4.3: Fluxo de telas de autenticação e encaminhamento para a aplicação. Fonte: autoria própria.

A Figura 4.4 detalha o fluxo de criação de um morador, evidenciando a divisão em etapas e os retornos possíveis (cancelamento/voltar), que devem preservar dados e

reduzir retrabalho do voluntário.

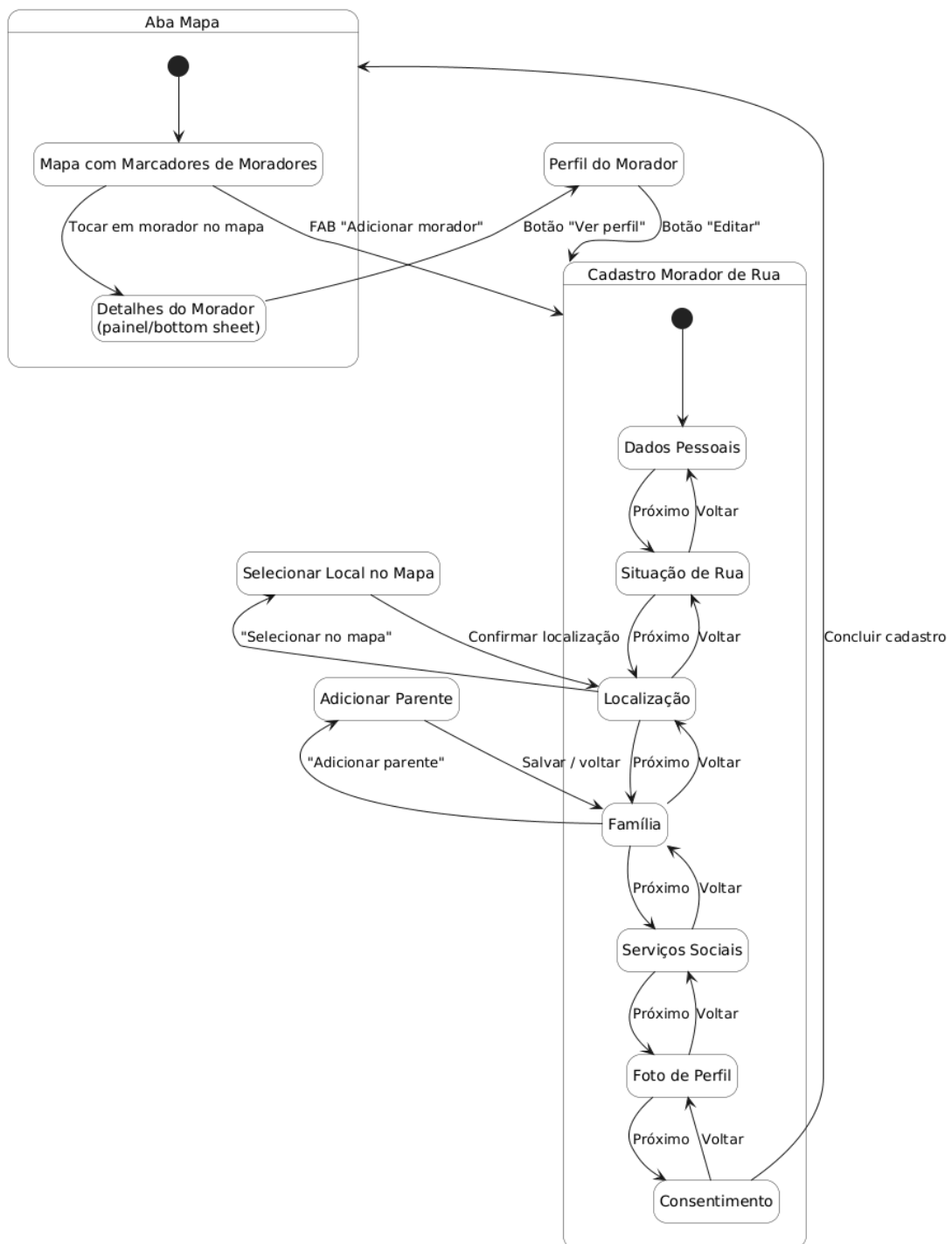


Figura 4.4: Fluxo de telas para criação e cadastro de morador. Fonte: autoria própria.

A Figura 4.5 apresenta o fluxo do perfil voluntário, conectando a criação de pedidos ao acompanhamento de ofertas e deixando explícitas as telas de gestão que suportam a continuidade do atendimento.

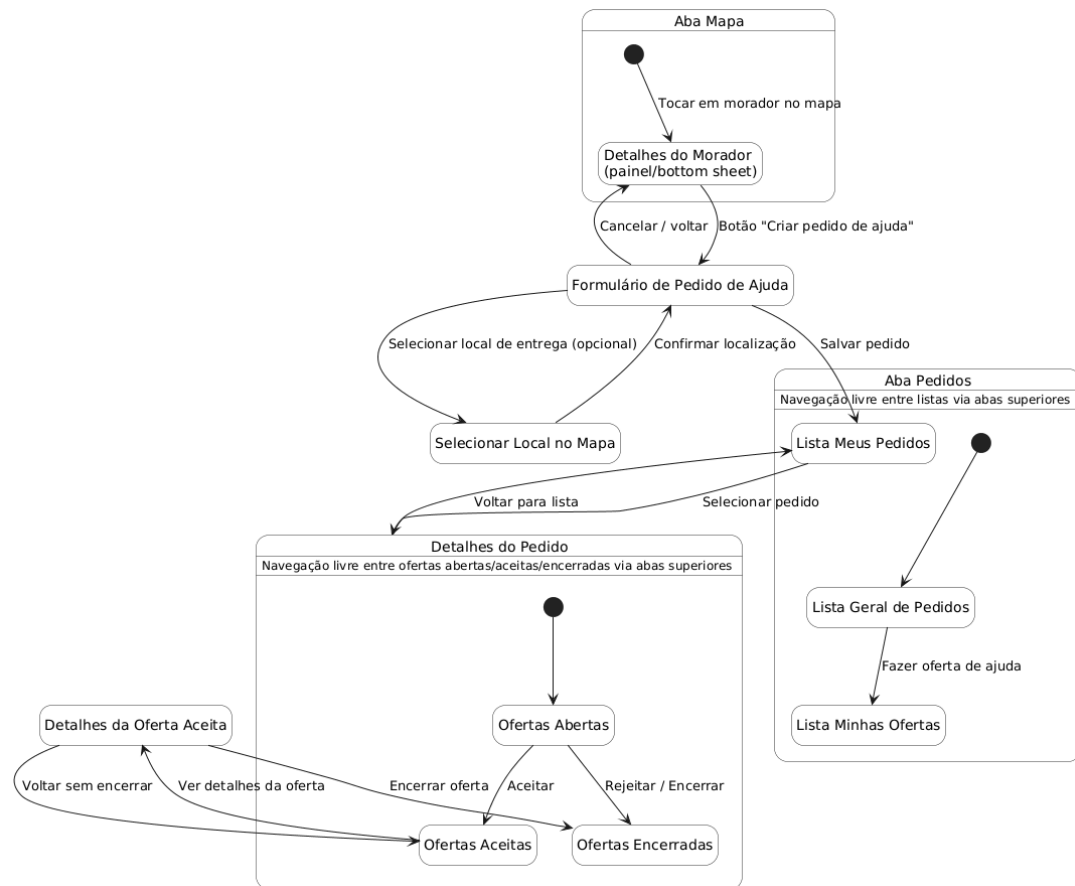


Figura 4.5: Fluxo de telas para solicitar e ofertar ajuda (perfil voluntário). Fonte: autoria própria.

A Figura 4.6 descreve o fluxo do perfil básico, no qual o usuário explora pedidos abertos, abre detalhes e registra uma oferta de ajuda, com restrições para evitar ofertas duplicadas.

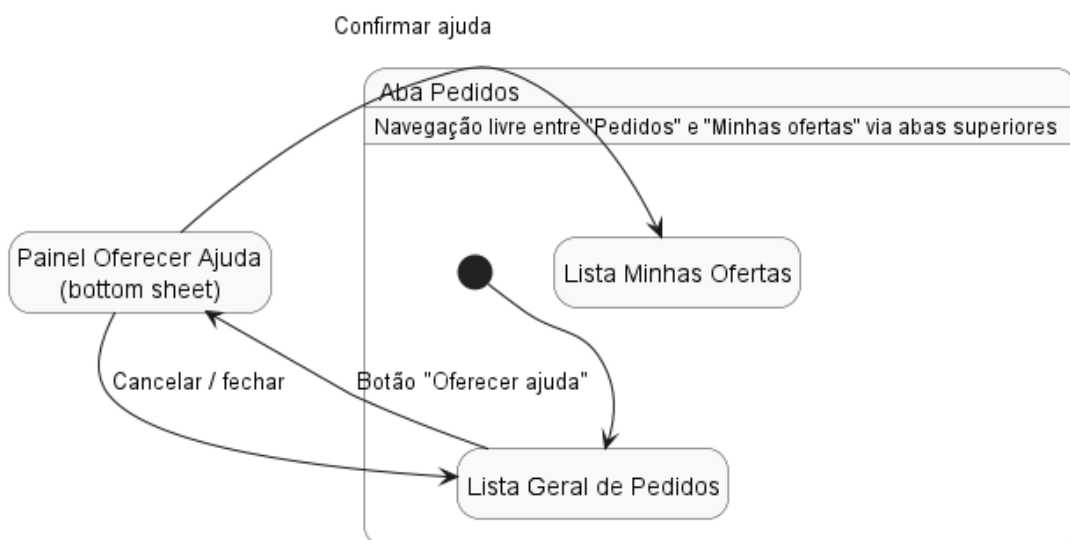


Figura 4.6: Fluxo de telas para ofertar ajuda (perfil básico). Fonte: autoria própria.

4.5.1 Autenticação

O fluxo de autenticação implementado (ilustrado no *screenflow* da Figura 4.3) inicia na tela de *login* e encaminha o usuário para o aplicativo principal somente após validação de sessão e verificação de perfil completo.

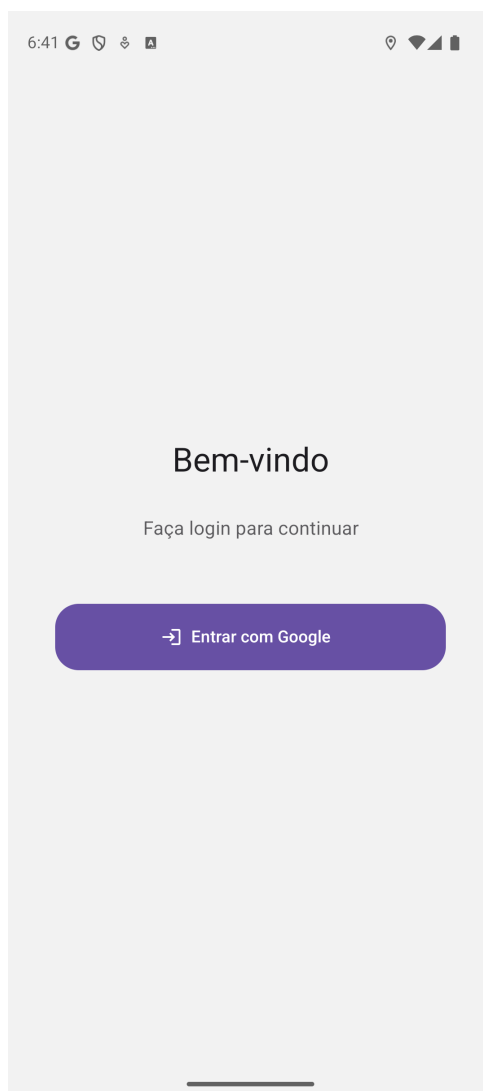


Figura 4.7: Tela de *login* do usuário. Fonte: autoria própria.

4.5.2 Cadastro de Morador

O cadastro é dividido em sete etapas.

Informações Pessoais

Nesta etapa, o voluntário insere dados básicos e seleciona atributos em listas controladas para reduzir variação e aumentar consistência dos registros.

The screenshot shows a mobile application interface titled "Adicionar Morador de Rua". At the top, there is a back arrow and the title. Below the title is a section header "Informações Pessoais" with a purple underline. The form contains several input fields: "Apelido" (empty), "Nome *" (filled with "John"), "Sobrenome *" (filled with "Doe"), "Data de Nascimento *" (filled with "04/10/2001" and a calendar icon), and three radio buttons for "Masculino" (selected), "Feminino", and "Outro". Below these are three dropdown menus: "Estado Civil" (filled with "Solteiro"), "Religião" (filled with "Católica"), and "Escolaridade" (filled with "Ensino Médio Completo"). At the bottom right, there is a purple button labeled "Próximo →".

Figura 4.8: Interface implementada para cadastro de informações pessoais do morador.
Fonte: autoria própria.

Situação de Rua

Nesta etapa, o sistema registra informações de contexto (tempo e motivo), que são utilizadas para histórico e acompanhamento do atendimento.

The screenshot displays a mobile application interface titled "Adicionar Morador de Rua". At the top, there is a back arrow and the title. Below the title, the form is organized into sections:

- Situação de Rua**: A section header with a purple underline.
- Tempo em Situação de Rua ***: A text input field containing the number "6". Below it are three buttons: "Dias", "Meses" (which is highlighted in purple), and "Anos".
- Motivo da Situação de Rua ***: A dropdown menu showing "Desemprego" with a downward arrow.
- Descrição do Motivo**: A text input field for providing details about the reason.

At the bottom of the screen, there are two navigation buttons: "Voltar" (with a left arrow) and "Próximo" (with a right arrow).

Figura 4.9: Interface implementada para registrar situação de rua e contexto. Fonte: autoria própria.

Seleção da Localização

Nesta etapa, o voluntário seleciona um ponto no mapa que representa um local relevante (por exemplo, local principal ou região frequentada), permitindo que consultas por *viewport* e ordenação por distância funcionem de forma consistente.

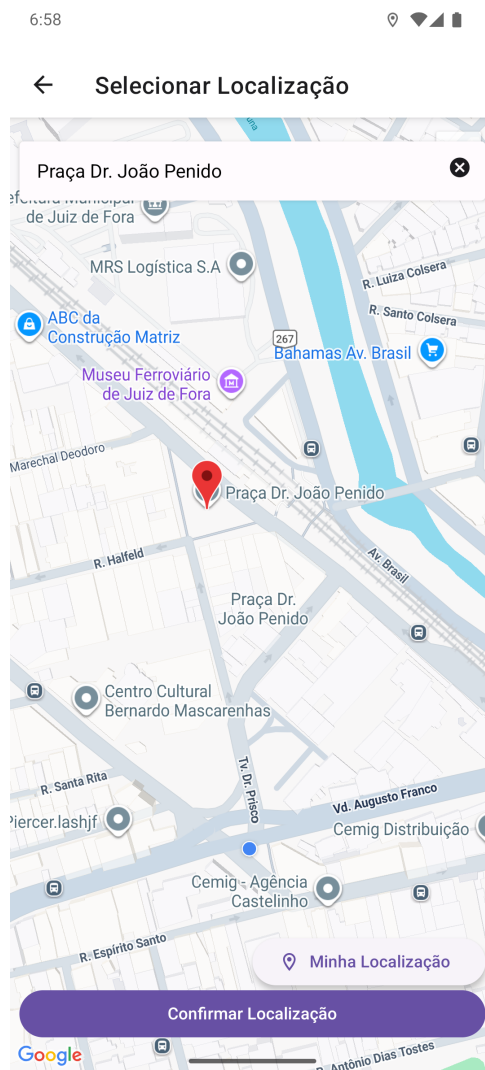


Figura 4.10: Interface implementada para seleção de localização no mapa. Fonte: autoria própria.

Localizações do Morador

Após inserir uma ou mais localizações, o aplicativo apresenta uma lista para revisão e edição, reduzindo erros de registro e apoiando atualizações futuras.

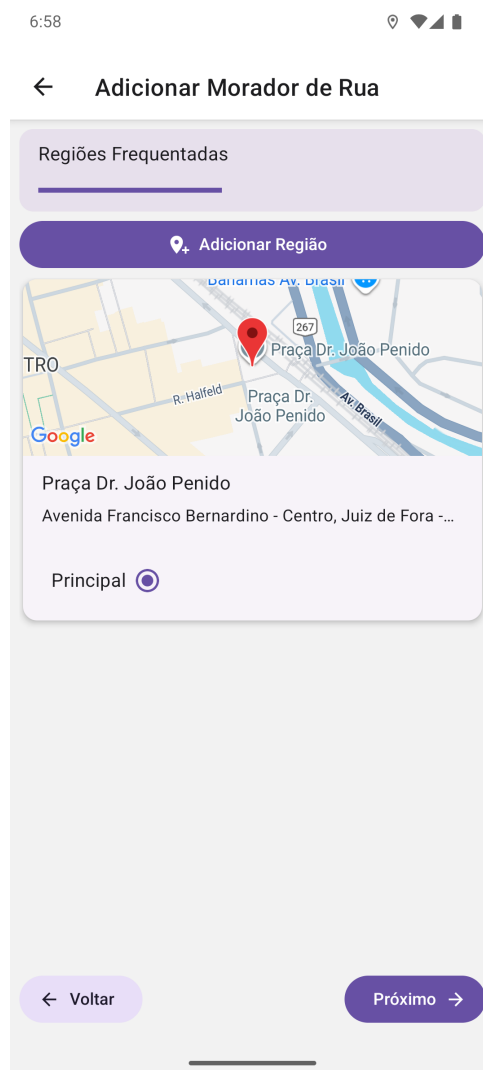
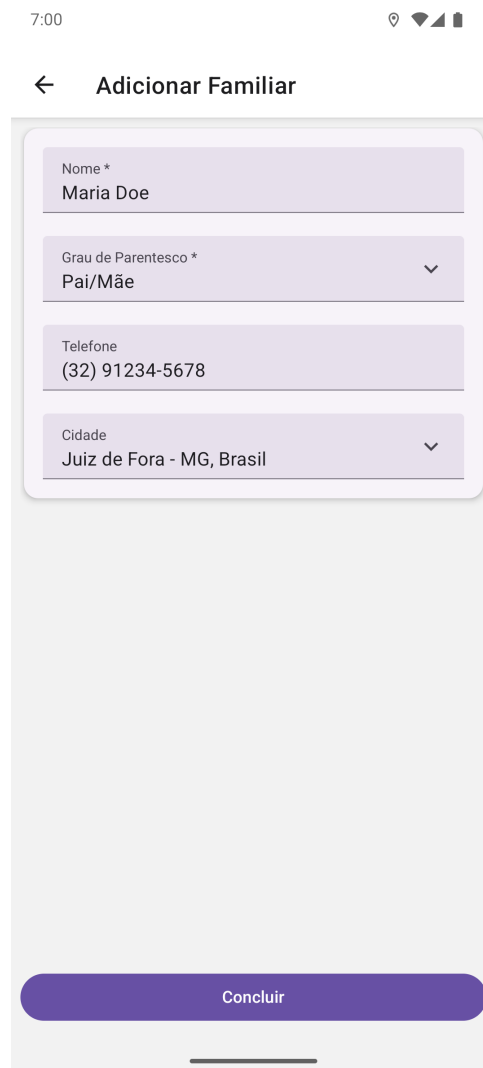


Figura 4.11: Interface implementada listando localizações do morador. Fonte: autoria própria.

Cadastro de Familiar

O registro de familiares permite documentar rede de apoio e possíveis contatos, quando disponíveis, mantendo o histórico vinculado ao perfil do morador.



The screenshot shows a mobile application interface for adding a family member. At the top, the status bar displays the time 7:00 and icons for location, Wi-Fi, and battery. Below the status bar, there is a back arrow and the title "Adicionar Familiar". The main content area contains a form with four input fields: "Nome*" with the value "Maria Doe", "Grau de Parentesco*" with a dropdown menu showing "Pai/Mãe", "Telefone" with the value "(32) 91234-5678", and "Cidade" with a dropdown menu showing "Juiz de Fora - MG, Brasil". At the bottom of the form, there is a purple button labeled "Concluir".

Figura 4.12: Interface implementada para cadastro de familiar. Fonte: autoria própria.

Lista de Familiares

A lista de familiares consolida os registros cadastrados, permitindo conferência rápida e evitando duplicidade ao longo de cadastros sucessivos.

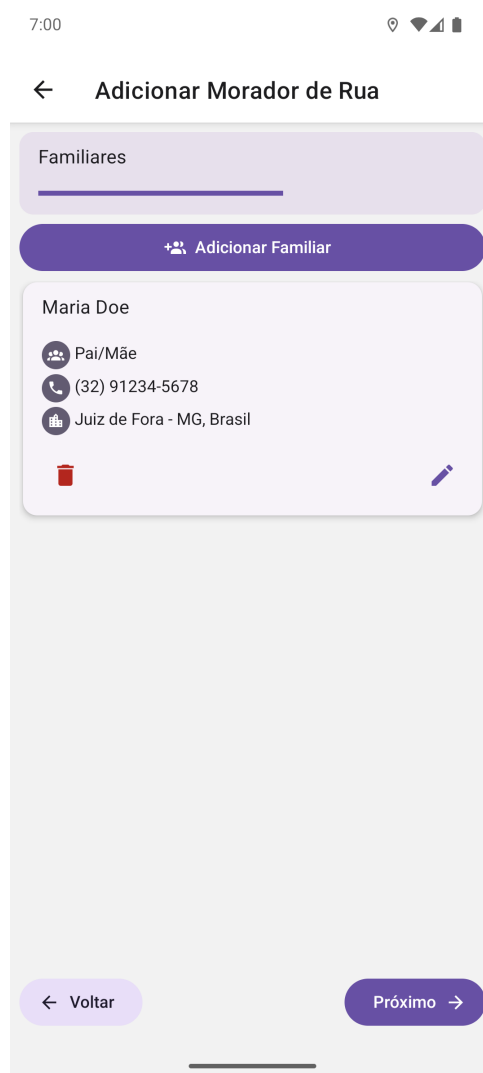


Figura 4.13: Interface implementada com lista de familiares cadastrados. Fonte: autoria própria.

Serviços Sociais

Nesta etapa, são registrados serviços e instituições relacionadas ao atendimento (por exemplo, CRAS e CREAS), apoiando acompanhamento e encaminhamentos.

The screenshot shows a mobile application interface titled "Adicionar Morador de Rua" (Add Street Resident). The interface is displayed on a smartphone screen with a status bar at the top showing the time 7:00, signal strength, Wi-Fi, and battery icons. The main content area is titled "Serviços Sociais" (Social Services) and contains three sections of questions, each with radio button options:

- Possui cadastro no CRAS (Centro de Referência de Assistência Social)**: Options are "Sim" (Yes), "Não sei" (I don't know), and "Não" (No).
- Possui cadastro no CREAS (Centro de Referência Especializado de Assistência Social)**: Options are "Sim" (Yes), "Não sei" (I don't know), and "Não" (No).
- Conhece a existência do Serviço de Albergue de Juiz de Fora?**: Options are "Sim" (Yes) and "Não" (No).

At the bottom of the screen, there are two navigation buttons: "Voltar" (Back) with a left arrow and "Próximo" (Next) with a right arrow.

Figura 4.14: Interface implementada para registrar serviços sociais. Fonte: autoria própria.

Foto de Perfil

A foto auxilia na identificação do morador e reduz ambiguidades em regiões com múltiplos registros próximos.

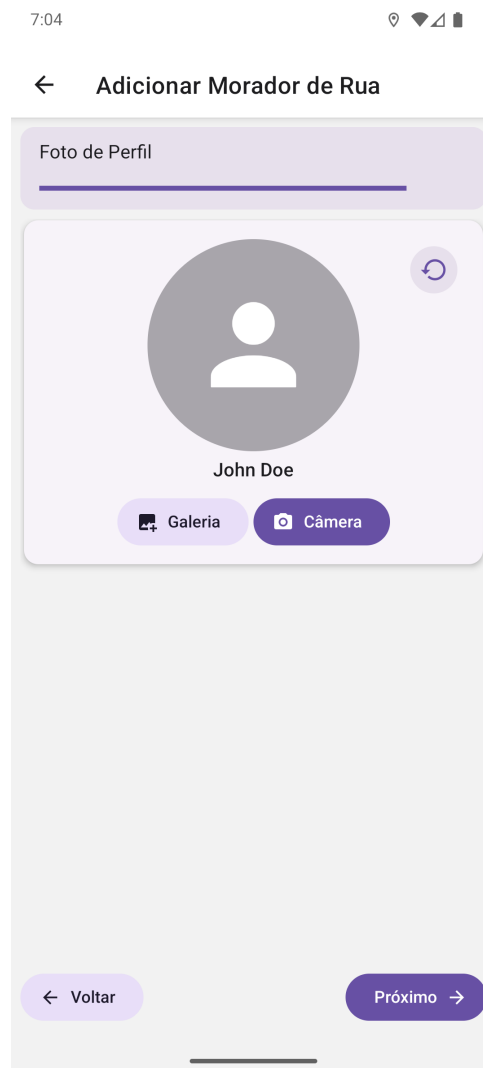


Figura 4.15: Interface implementada para captura da foto de perfil. Fonte: autoria própria.

Consentimento

O registro de consentimento (áudio/vídeo) documenta a autorização para armazenamento de informações e mídias, reforçando o cuidado com privacidade e dados sensíveis.

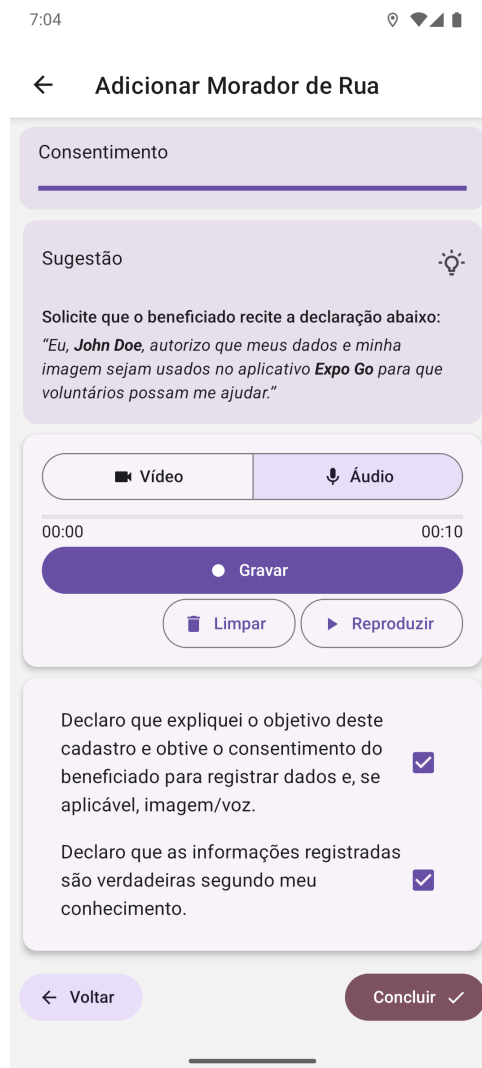


Figura 4.16: Interface implementada para registro de consentimento em áudio ou vídeo. Fonte: autoria própria.

4.5.3 Mapa e Visualização de Moradores

A Figura 4.17 ilustra a visualização cartográfica com carregamento por *viewport*, enquanto a Figura 4.18 mostra o painel de detalhes/ações ao selecionar um marcador, consolidando o vínculo entre consulta geoespacial e navegação de detalhes.

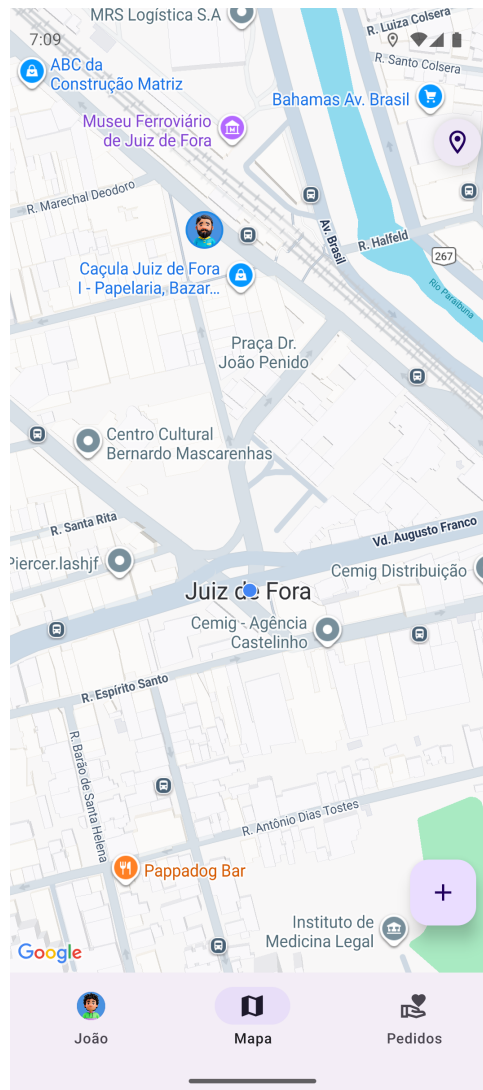


Figura 4.17: Mapa com moradores carregados por *viewport*. Fonte: autoria própria.

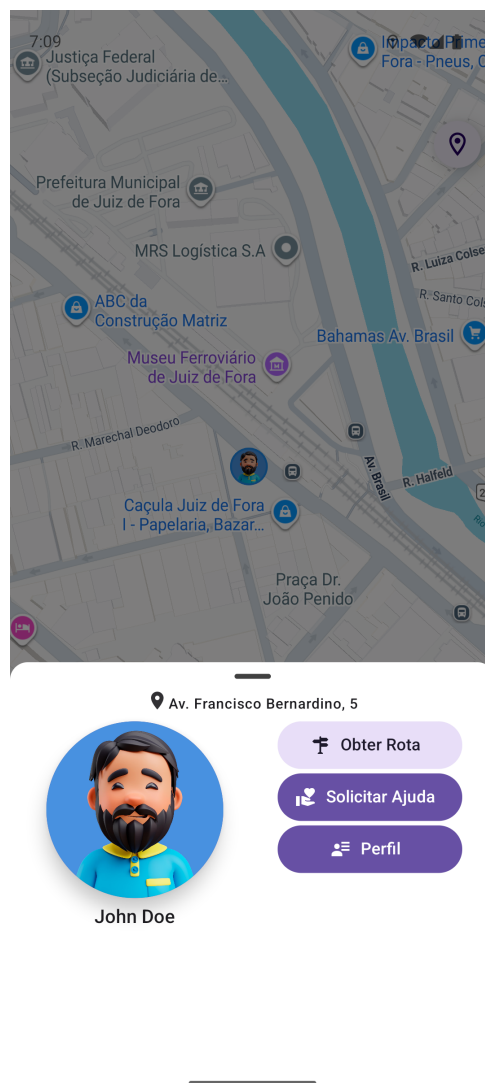


Figura 4.18: Detalhes e ações disponíveis ao selecionar um morador no mapa. Fonte: autoria própria.

4.5.4 Perfil do Morador

O perfil consolidado do morador (Figura 4.19) reúne histórico, localizações e ações relacionadas, reduzindo a necessidade de navegação entre múltiplas telas para obter contexto.

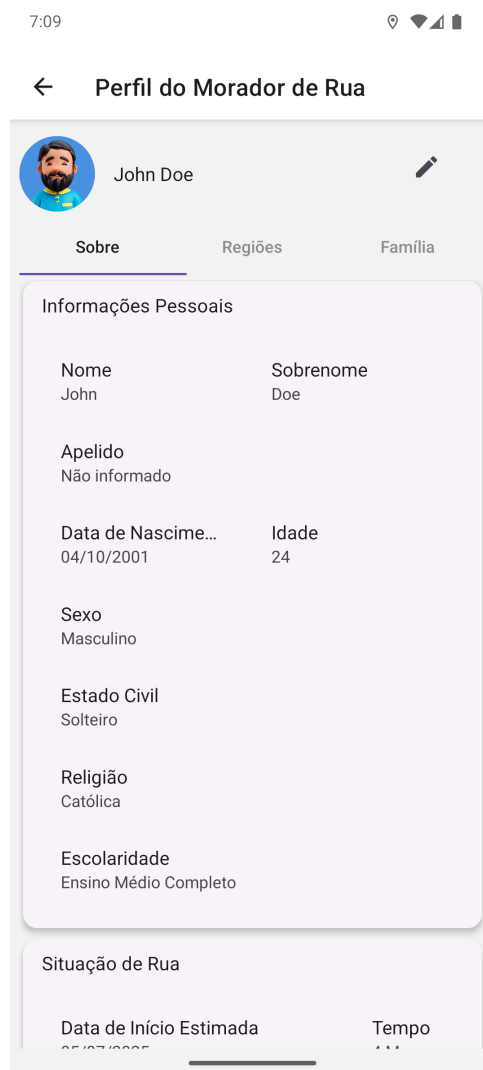


Figura 4.19: Perfil completo do morador. Fonte: autoria própria.

4.5.5 Pedidos de Ajuda

Criar Pedido de Ajuda

O formulário de criação (Figura 4.20) captura tipo, descrição, prioridade e prazo, além de permitir definir/confirmar localização de entrega, refletindo as decisões destacadas na modelagem de tarefas.

The screenshot shows a mobile application interface for creating a request for help. The title is "Nova Solicitação de Ajuda". The form includes the following fields and controls:

- Tipo de Ajuda ***: Alimento não perecível (dropdown menu)
- Título ***: Cesta básica 7 dias
- Descrição**: Alimentos não perecíveis e frutas
- Prioridade**: A slider control set to 9, with a scale from 0 (baixa) to 10 (alta prioridade).
- Quantidade**: 2
- Unidade**: Item (dropdown menu)
- Data Limite**: 05/12/2025 (calendar icon)
- Selecionar Local de Entrega**: A button with a location pin icon.
- Mapa**: A map showing the delivery location at Praça Dr. João Penido, with nearby streets R. Halfeld and Av. Brasil.
- Botão**: Criar Solicitação

Figura 4.20: Interface implementada para criação de pedido de ajuda. Fonte: autoria própria.

Meus Pedidos

Após a criação, o usuário acompanha os pedidos publicados (Figura 4.21), visualizando status e acessando ofertas recebidas para decisão.

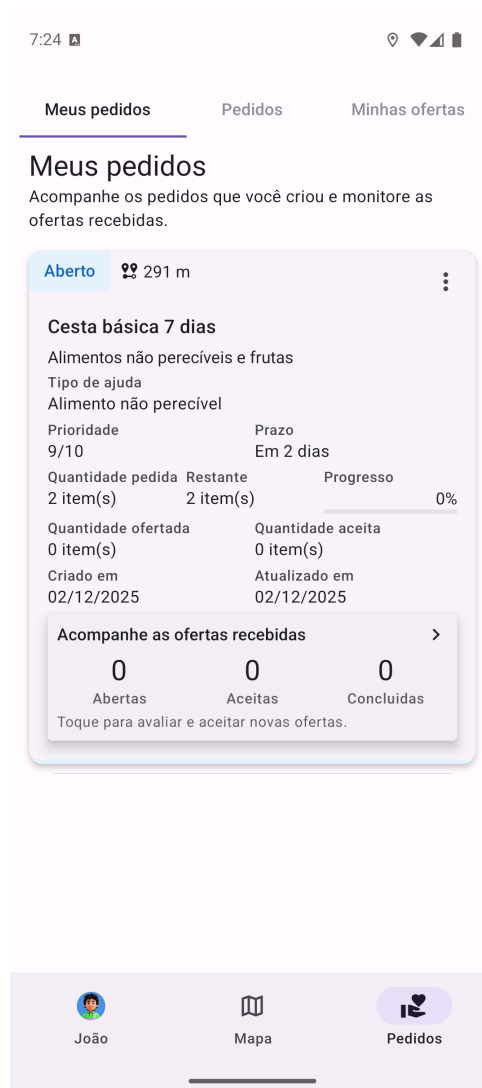


Figura 4.21: Interface implementada para gerenciamento de pedidos do usuário. Fonte: autoria própria.

Pedidos Disponíveis

Usuários do perfil básico exploram pedidos abertos de terceiros na listagem (Figura 4.22), normalmente ordenada por distância e filtrada por contexto.

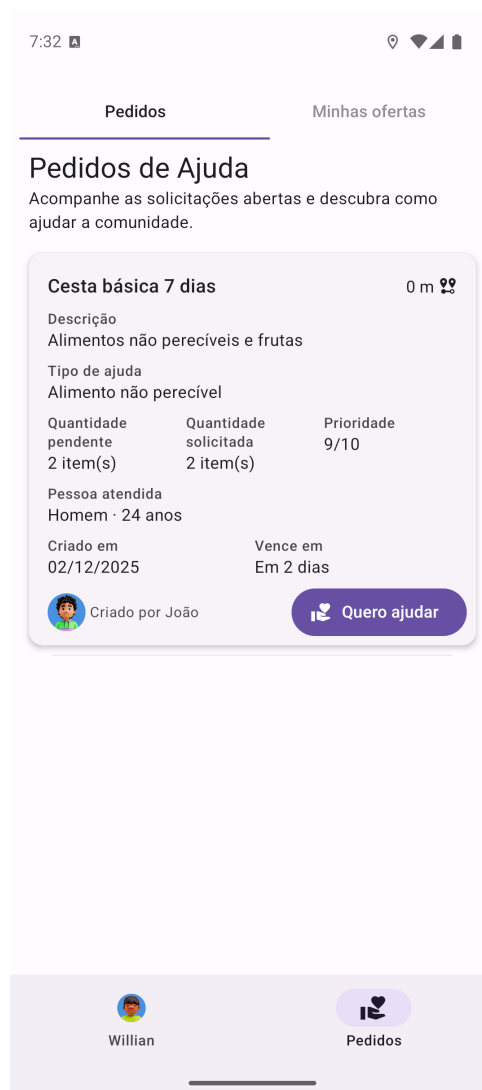


Figura 4.22: Interface implementada com pedidos de ajuda disponíveis. Fonte: autoria própria.

Ofertas para um Pedido

Ao abrir um pedido próprio, o voluntário visualiza e gerencia ofertas recebidas (Figura 4.23), apoiando decisões de aceitar, rejeitar e acompanhar conclusão.



Figura 4.23: Interface implementada para acompanhar ofertas recebidas em um pedido. Fonte: autoria própria.

4.5.6 Detalhes de Oferta Aceita

Quando uma oferta é aceita, a tela de detalhes (Figura 4.24) fornece informações para coordenação (contato/logística), reduzindo ambiguidades no acompanhamento do atendimento.

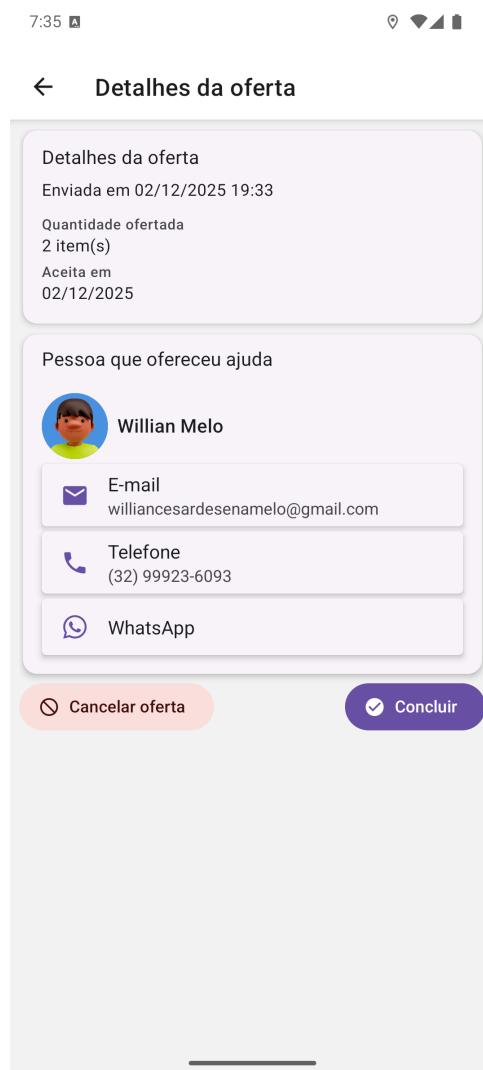


Figura 4.24: Detalhes e contato do voluntário na oferta aceita. Fonte: autoria própria.

4.5.7 Ofertas Concluídas

O histórico de ofertas concluídas (Figura 4.25) permite consultar atendimentos finalizados e apoiar prestação de contas ou continuidade de ações.

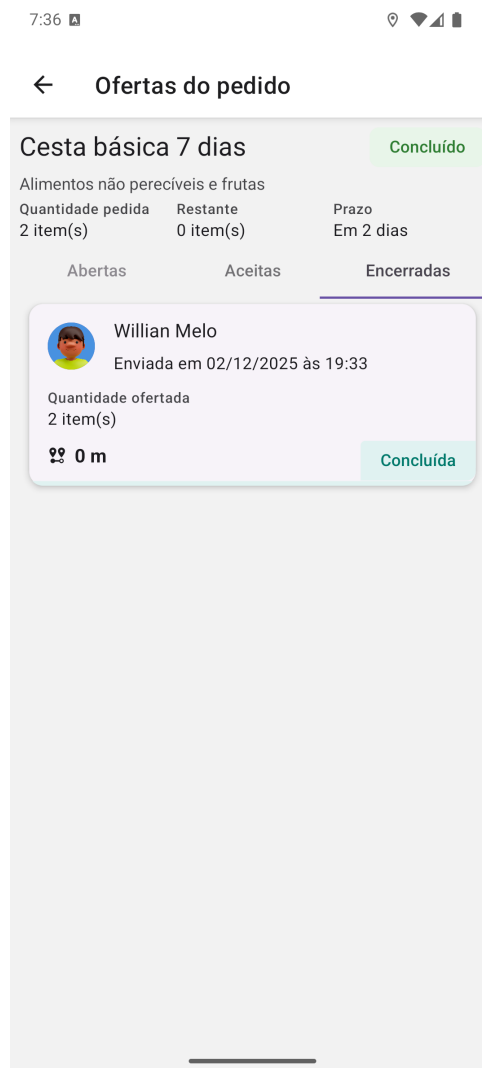


Figura 4.25: Interface implementada com histórico de ofertas concluídas. Fonte: autoria própria.

4.5.8 Minhas Ofertas

A listagem de ofertas realizadas (Figura 4.26) permite que o usuário acompanhe o status de suas contribuições (pendente, aceita, rejeitada ou concluída).

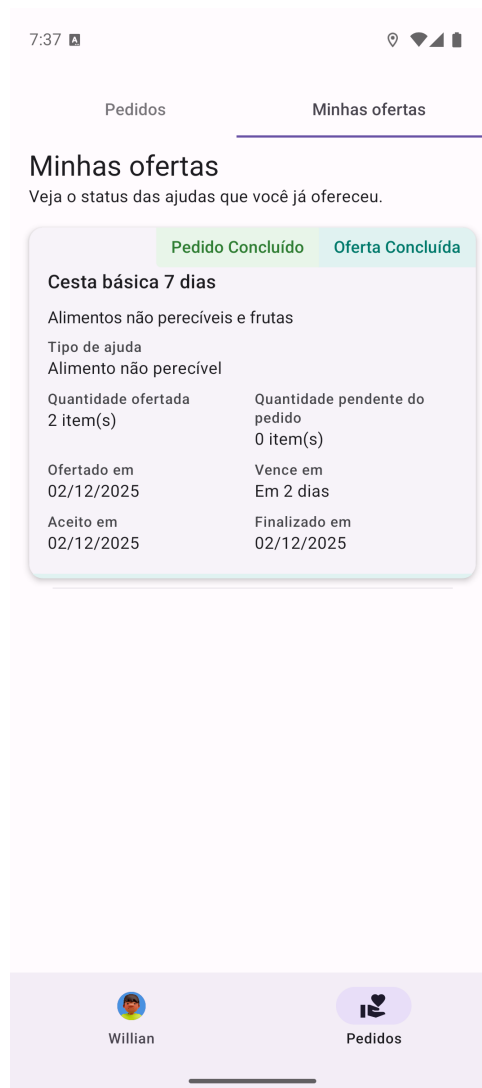


Figura 4.26: Interface implementada com ofertas realizadas pelo usuário. Fonte: autoria própria.

4.5.9 Finalização do Cadastro

O fluxo de finalização do cadastro (Figura 4.27) coleta informações mínimas de perfil para permitir ofertas e comunicação, alinhado ao *screenflow* de autenticação.



The screenshot shows a mobile application interface titled "Finalizar Cadastro" (Finalize Registration). At the top, there is a back arrow and the title. Below the title, a message reads: "Finalize o seu cadastro fornecendo as informações. Algumas funcionalidades não serão acessíveis até que a finalização seja concluída." (Finalize your registration by providing the information. Some features will not be accessible until the finalization is complete.)

The interface is divided into two main sections:

- Foto de perfil (Profile Photo):** This section contains a circular profile picture of a cartoon character with curly hair and a green shirt. To the right of the photo is a close button (X). Below the photo are two buttons: "Galeria" (Gallery) and "Câmera" (Camera).
- Dados pessoais (Personal Data):** This section contains three text input fields:
 - "Primeiro nome *" (First name) with the value "João".
 - "Sobrenome *" (Last name) with the value "Sena".
 - "Telefone *" (Phone number) which is currently empty.Below these fields are two buttons: "Sim" (Yes) and "Não" (No). A small text label below the buttons reads: "Informe se seu número possui WhatsApp." (Indicate if your number has WhatsApp.)

At the bottom of the form is a large purple button with a checkmark and the text "Salvar" (Save).

Figura 4.27: Interface implementada para finalização do cadastro do usuário. Fonte: autoria própria.

4.5.10 Perfil do Usuário

O perfil do usuário (Figura 4.28) concentra informações pessoais e ações como atualização de dados e saída da conta.

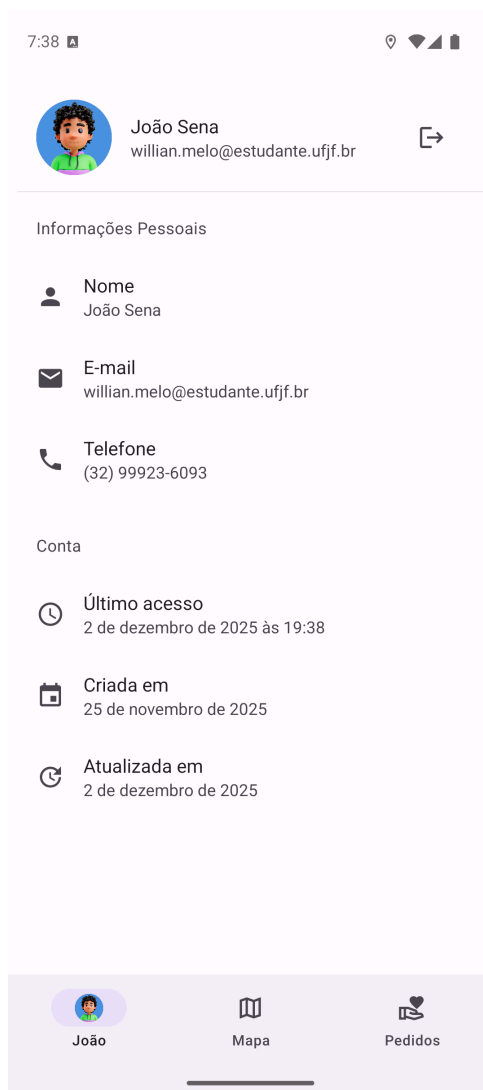


Figura 4.28: Interface implementada do perfil do usuário. Fonte: autoria própria.

4.6 Considerações Finais

Este capítulo apresentou a arquitetura, os componentes e as telas que compõem o sistema *Pró-Inclusão*. A estrutura integrada entre *back-end*, aplicativo móvel e serviços externos permite um fluxo contínuo de informações e suporte a funcionalidades essenciais como autenticação, geolocalização, registro de moradores e gestão de pedidos de ajuda.

5 Avaliação do Sistema

Neste Capítulo é apresentada a avaliação do sistema *Pró-Inclusão*, conduzida sob duas abordagens complementares: testes funcionais e avaliação heurística. Na Seção 5.1 é descrita a Metodologia de Avaliação adotada. Na Seção 5.2 são apresentados os Testes Funcionais realizados, com roteiros e resultados observados para os principais fluxos do sistema. Na Seção 5.3 é apresentada a Avaliação Heurística da interface do aplicativo, com os problemas identificados e sua classificação por severidade. Na Seção 5.4 são apresentadas as Recomendações de Melhoria derivadas dos resultados da avaliação. Finalmente, na Seção 5.5 são apresentadas as considerações finais deste Capítulo. No Capítulo 6 são apresentadas as conclusões, as contribuições e as perspectivas de trabalhos futuros do aplicativo *Pró-Inclusão*.

5.1 Metodologia de Avaliação

A avaliação do sistema *Pró-Inclusão* foi conduzida em duas frentes complementares: testes funcionais e avaliação heurística da interface. A combinação desses métodos é amplamente recomendada na literatura de engenharia de software e Interação Humano-Computador (IHC), pois permite identificar tanto falhas de implementação quanto problemas de uso que afetam a experiência do usuário (PRESSMAN; MAXIM, 2019; SOMMERVILLE, 2011; PREECE; ROGERS; SHARP, 2015; NIELSEN, 1993).

Os testes funcionais tiveram como objetivo verificar se as principais funcionalidades do sistema operam conforme o esperado, considerando o fluxo completo de interação e os diferentes perfis de usuários previstos (básico e voluntário). Foram definidos cenários de uso baseados nos requisitos levantados nos capítulos anteriores, contemplando tarefas como autenticação, cadastro de moradores em situação de rua, criação e acompanhamento de pedidos de ajuda, gerenciamento de ofertas e interação com o mapa. Para cada cenário, o comportamento observado foi comparado com o comportamento esperado, buscando evidências de falhas, inconsistências de estado, problemas de validação e situações de

erro não tratadas (PRESSMAN; MAXIM, 2019; SOMMERVILLE, 2011). A natureza exploratória dos testes permitiu exercitar trajetórias típicas de uso e caminhos alternativos, aproximando-se do contexto real de utilização do aplicativo.

Em paralelo, foi realizada uma avaliação heurística da interface do aplicativo móvel, adotando-se as heurísticas de usabilidade propostas por Nielsen (NIELSEN, 1993) e consolidadas em trabalhos posteriores (NIELSEN; MOLICH, 1990; PREECE; ROGERS; SHARP, 2015; SHNEIDERMAN et al., 2016). A avaliação heurística é um método de inspeção no qual avaliadores analisam as telas e fluxos de interação à luz de um conjunto de princípios gerais de usabilidade, identificando violações recorrentes e oportunidades de melhoria. Esse método é particularmente adequado em contextos de recursos limitados, por exigir menos infraestrutura do que testes com usuários e, ao mesmo tempo, ser capaz de revelar um número significativo de problemas de usabilidade (NIELSEN, 1993; NIELSEN; MOLICH, 1990).

No caso do *Pró-Inclusão*, a avaliação concentrou-se nas telas e fluxos considerados mais críticos para o uso cotidiano do sistema: autenticação, cadastro de moradores, visualização e filtragem no mapa, criação e acompanhamento de pedidos de ajuda, além do painel de ofertas. Para cada fluxo, foram percorridas sequências completas de interação, registrando-se problemas observados, a heurística violada e exemplos de situações em que o problema se manifesta. Em seguida, atribuiu-se um grau de severidade qualitativo (baixa, média ou alta) a cada problema, considerando critérios de impacto na realização da tarefa, frequência de ocorrência e dificuldade de correção, conforme recomenda Nielsen (NIELSEN, 1993).

Por fim, os resultados das duas frentes de avaliação foram consolidados. Os achados dos testes funcionais permitiram verificar a robustez das funcionalidades essenciais e a adequação do comportamento do sistema aos requisitos definidos. Os resultados da avaliação heurística, por sua vez, foram sintetizados em uma tabela de problemas, organizada por heurística violada e severidade, com o objetivo de apoiar a priorização de ajustes na interface em versões futuras do *Pró-Inclusão*.

5.2 Testes Funcionais

Os testes funcionais foram conduzidos de forma exploratória, simulando o uso real do sistema pelos dois perfis de usuários: básico e voluntário. As principais funcionalidades avaliadas foram estruturadas em casos de teste, que descrevem ações, resultados esperados e resultados observados, alinhando-se a recomendações de engenharia de software para validação de requisitos (PRESSMAN; MAXIM, 2019; SOMMERVILLE, 2011).

5.2.1 Roteiros de Teste

A Tabela 5.1 sintetiza casos de teste representativos dos fluxos avaliados. Os dados registrados referem-se a comportamentos observados durante a execução dos testes, sem uso de métricas quantitativas simuladas.

Tabela 5.1: Roteiros de teste funcional para fluxos principais do sistema.

Caso de Teste	Ação / Cenário	Resultado Esperado	Resultado Observado
CT01 - Autenticação	Usuário acessa a tela inicial, seleciona <i>login</i> com Google e conclui o fluxo de autenticação.	Criação ou recuperação de perfil e acesso à área principal após autenticação bem-sucedida.	Autenticação concluída sem erros, com <i>tokens</i> anexados às requisições subsequentes e redirecionamento para a área principal.
CT02 - Cadastro de morador	Voluntário cadastra um novo morador preenchendo as sete etapas do formulário, incluindo localização no mapa e consentimento em mídia.	Validação de campos obrigatórios, criação de registro de morador e associação de localizações e mídias.	Validações aplicadas por etapa, localizações exibidas no mapa e mídias registradas corretamente para o morador.

CT03 - Criação de pedido de ajuda	Voluntário, a partir do perfil de um morador, cria um pedido de ajuda com tipo, descrição, prazo e localização de entrega.	Pedido registrado em estado “aberto” e vinculado ao morador, disponível em listagens relevantes.	Pedido criado e exibido em “Meus pedidos”, aparecendo nas listagens com ordenação correta por prioridade, prazo e distância.
CT04 - Oferta de ajuda	Usuário básico acessa a lista de pedidos disponíveis, seleciona um pedido de terceiro e realiza uma oferta de ajuda.	Oferta registrada em estado “pendente” e visível para o criador do pedido para avaliação.	Oferta criada com sucesso, aparecendo para o criador do pedido e impedindo nova oferta ativa do mesmo usuário para o mesmo pedido.
CT05 - Bloqueio de duplicidade de oferta	Usuário com oferta “pendente” tenta criar nova oferta para o mesmo pedido.	Bloqueio da ação com mensagem de erro informando a existência de oferta ativa.	Sistema impede a criação da oferta, exibindo mensagem informativa sobre a oferta já existente.
CT06 - Mapa e geolocalização	Usuário movimenta o mapa e altera o <i>zoom</i> , observando a atualização de moradores e pedidos na região.	Carregamento dinâmico de elementos com base na <i>viewport</i> e na localização do usuário, com marcadores atualizados.	Marcadores de moradores e pedidos atualizados sem erros aparentes, com boa responsividade entre o mapa e as telas de detalhes.

Durante os testes, não foram observadas falhas críticas que inviabilizassem o uso das funcionalidades principais. Entretanto, alguns comportamentos identificados

nos cenários de teste (como navegação com campos obrigatórios em branco em etapas específicas) foram retomados na avaliação heurística e nas recomendações de melhoria, por estarem diretamente relacionados à prevenção de erros e feedback ao usuário.

5.3 Avaliação Heurística

A avaliação heurística foi realizada com base em um conjunto de dez heurísticas amplamente utilizadas em IHC (NIELSEN, 1993), visando avaliar a interface do aplicativo móvel. Foram analisadas telas de autenticação, cadastro de moradores, mapa, pedidos de ajuda e painel de ofertas.

5.3.1 Heurísticas Aplicadas

As heurísticas consideradas incluem:

1. Visibilidade do status do sistema.
2. Correspondência com o mundo real.
3. Controle e liberdade do usuário.
4. Consistência e padrões.
5. Prevenção de erros.
6. Reconhecimento em vez de memorização.
7. Flexibilidade e eficiência de uso.
8. Design estético e minimalista.
9. Recuperação de erros.
10. Ajuda e documentação.

5.3.2 Problemas Identificados

A Tabela abaixo sumariza os principais problemas observados durante a avaliação. A severidade foi classificada em três níveis (baixa, média e alta), considerando impacto na realização das tarefas, frequência potencial de ocorrência e esforço estimado para correção, em linha com recomendações de Nielsen (NIELSEN, 1993).

Tabela 5.3: Problemas encontrados durante a avaliação heurística.

Heurística	Problema Observado	Severidade
Visibilidade do status	Ausência de feedback claro ao salvar alterações no perfil.	Média
Consistência	Nomeclaturas ligeiramente diferentes entre telas de pedidos.	Baixa
Prevenção de erros	Usuário consegue navegar sem preencher campos obrigatórios em alguns passos.	Alta
Design minimalista	Alguns rótulos longos ocupam espaço excessivo em telas pequenas.	Baixa
Flexibilidade	Falta opção de repetir pedidos comuns a partir de um modelo.	Média

5.4 Recomendações de Melhoria

Com base nos problemas identificados, foram propostas as seguintes melhorias:

- adicionar indicações visuais e mensagens de sucesso após salvar edições;
- padronizar títulos, botões e nomenclaturas;
- reforçar validações nos formulários mais longos;
- otimizar textos e rótulos para telas menores;
- permitir duplicação ou clonagem de pedidos recorrentes.

Essas recomendações visam melhorar a fluidez da interação, reduzir erros e aumentar a satisfação geral dos usuários.

5.5 Considerações Finais

A avaliação demonstrou que o sistema *Pró-Inclusão* apresenta boa aderência aos princípios de usabilidade e atende às principais necessidades dos usuários. Os testes funcionais confirmaram a robustez das funcionalidades essenciais, enquanto a avaliação heurística identificou oportunidades pontuais de melhoria, que poderão ser implementadas em versões futuras.

6 Conclusão

Neste Capítulo são apresentadas as conclusões deste trabalho de conclusão de curso. Na Seção 6.1 é apresentada a Síntese dos Resultados obtidos com o desenvolvimento e a avaliação da plataforma *Pró-Inclusão*. Na Seção 6.2 são apresentadas as Contribuições do Trabalho ao campo da Interação Humano-Computador e ao domínio das ações sociais urbanas. Na Seção 6.3 são discutidas as Limitações identificadas no sistema e no processo de avaliação. Na Seção 6.4 são apresentadas as perspectivas de Trabalhos Futuros para a evolução da plataforma. Finalmente, na Seção 6.5 são apresentadas as considerações finais deste Capítulo e deste trabalho de conclusão de curso.

6.1 Síntese dos Resultados

O sistema criado atende ao objetivo geral proposto: fornecer uma plataforma integrada, composta por aplicativo móvel e API *web*, capaz de facilitar o registro estruturado de moradores em situação de rua, organizar pedidos de ajuda e aproximar voluntários de necessidades próximas por meio de geolocalização.

Os principais resultados obtidos incluem:

- implementação de um *back-end* robusto, com autenticação moderna, regras de negócio bem definidas e suporte a dados geoespaciais;
- desenvolvimento de um aplicativo móvel multiplataforma com fluxo completo de cadastro, mapa interativo, gerenciamento de pedidos e ofertas;
- aplicação consistente de princípios de usabilidade e IHC no *design* das interfaces;
- integração estável com Google Places e S3/MinIO para enriquecimento de dados e armazenamento de mídias;
- validação por meio de testes funcionais e avaliação heurística.

Tais resultados indicam que a solução proposta é tecnicamente viável e, à luz da avaliação heurística realizada, apresenta boa aderência aos princípios de usabilidade para o contexto de assistência social.

6.2 Contribuições do Trabalho

As principais contribuições deste trabalho são:

- a definição de um modelo de sistema voltado ao apoio de pessoas em situação de rua, integrando tecnologias modernas e princípios sólidos de IHC;
- a elaboração de fluxos de interação completos, derivados de perfis de usuário e cenários de uso representativos levantados na análise do domínio;
- a implementação de um conjunto de recursos que combinam localização, registro multimídia, listas de apoio e comunicação entre usuários;
- a disponibilização de uma solução extensível, capaz de ser adaptada a diferentes contextos de assistência urbana.

Além disso, o trabalho reforça a importância de projetar sistemas sensíveis à complexidade humana e social das interações, respeitando privacidade, consentimento e responsabilidade no tratamento de dados.

6.3 Limitações

Embora o sistema tenha alcançado seus objetivos, algumas limitações foram identificadas:

- a avaliação envolveu testes heurísticos e funcionais, mas não contou com usuários em larga escala;
- a dependência de serviços externos como Google Places pode gerar custos e limitações de uso em cenários reais;
- o fluxo de atualização contínua da localização depende da qualidade da rede móvel e das permissões concedidas;

- funcionalidades institucionais avançadas (por exemplo, acompanhamento por equipes do CRAS e do CREAS) não foram incluídas no escopo.

Essas limitações apontam para aspectos a serem endereçados em trabalhos futuros, sendo a ausência de validação com usuários reais a mais relevante para uma avaliação conclusiva da efetividade do sistema em contexto operacional.

6.4 Trabalhos Futuros

Diversas melhorias e extensões podem ser exploradas a partir do sistema atual:

- realização de testes com usuários reais em campo para validação aprofundada de usabilidade;
- expansão do aplicativo para permitir integração com equipes de assistência social e órgãos municipais;
- implantação de um painel *web* administrativo com visualização geográfica avançada;
- inclusão de métricas de impacto social e indicadores de atendimento;
- uso de notificações push para agilizar interações entre usuários e voluntários;
- criação de funcionalidade para acompanhar histórico de atendimentos ao longo do tempo.

Essas possibilidades reforçam o caráter evolutivo do sistema e seu potencial de impacto social.

6.5 Considerações Finais

O *Pró-Inclusão* sugere que tecnologias móveis, quando combinadas a práticas de *design* centrado no usuário e avaliação de usabilidade, têm potencial para apoiar iniciativas humanitárias. O trabalho representa uma proposta computacional consistente para o domínio de ações sociais voltadas à população em situação de rua, cujos benefícios concretos em uso real dependem de validação empírica em campo.

Encerradas as fases de especificação, implementação e avaliação, o sistema encontra-se em estágio apto para uso piloto, etapa que permitirá, em trabalhos futuros, verificar empiricamente sua efetividade e identificar novos requisitos emergentes da prática.

Bibliografia

- AGNI, E. *A Colmeia da Experiência do Usuário*. 2012. <<https://www.mergo.com.br/blog/user-experience/colmeia-da-experiencia-do-usuario/>>. Acesso em: 27 janeiro 2026.
- BARBOSA, S. D. J.; SILVA, B. S. *Interação Humano-Computador*. Rio de Janeiro: Elsevier, 2010.
- DIX, A. et al. *Human-Computer Interaction*. 3. ed. Harlow: Pearson Education, 2004.
- FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000.
- International Organization for Standardization. *ISO 9241-11: Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts*. 2018. Norma técnica. Acesso: 07 dezembro 2025.
- IPEA. *Estimativa da População em Situação de Rua no Brasil*. Brasília, 2023.
- KRUG, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. 3. ed. Berkeley: New Riders, 2014.
- LONGLEY, P. A. et al. *Geographic Information Systems and Science*. 4. ed. Chichester: John Wiley and Sons, 2015.
- NIELSEN, J. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1993.
- NIELSEN, J.; BUDI, R. *Mobile Usability*. Berkeley: New Riders, 2012.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Seattle: ACM, 1990.
- Nielsen Norman Group. *10 Usability Heuristics for User Interface Design*. 1994. <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Atualizado em: 20 fevereiro 2024. Acesso em: 27 janeiro 2026.
- NORMAN, D. A. *The Design of Everyday Things*. Revised and expanded. New York: Basic Books, 2013.
- PREECE, J.; ROGERS, Y.; SHARP, H. *Interaction Design: Beyond Human-Computer Interaction*. 4. ed. Chichester: John Wiley and Sons, 2015.
- PRESSMAN, R. S.; MAXIM, B. *Software Engineering: A Practitioner's Approach*. 9. ed. New York: McGraw-Hill, 2019.
- PROJECT, P. *PostGIS Documentation*. 2024. <<https://postgis.net/documentation>>. Acesso em: 07 dezembro 2025.
- SHNEIDERMAN, B. et al. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 6. ed. Boston: Pearson, 2016.
- SOMMERVILLE, I. *Software Engineering*. 9. ed. Boston: Addison-Wesley, 2011.

SOUZA, C. S. de. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, MA: MIT Press, 2005.

W3C. *Web Content Accessibility Guidelines (WCAG) 2.2*. 2023. <<https://www.w3.org/TR/WCAG22/>>. Acesso em: 07 dezembro 2025.