

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE MINAS GERAIS (IFMG)
CAMPUS BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Henrique Araújo Miranda

**DESENVOLVIMENTO DE UM APLICATIVO PARA LOCAÇÃO DIGITAL DE
IMÓVEIS VOLTADO PARA USO DE ESTUDANTES EM PEQUENAS CIDADES**

BambuÍ – MG
2026

HENRIQUE ARAÚJO MIRANDA

**DESENVOLVIMENTO DE UM APLICATIVO PARA LOCAÇÃO DIGITAL DE
IMÓVEIS VOLTADO PARA USO DE ESTUDANTES EM PEQUENAS CIDADES**

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Marcos Roberto Ribeiro

Catálogo na Fonte Biblioteca IFMG - Campus Bambuí

M672d Miranda, Henrique Araújo.
Desenvolvimento de um aplicativo para locação digital de imóveis voltado para uso de estudantes em pequenas cidades. / Henrique Araújo Miranda. – 2026.
66 f.; il.: color.4

Orientador: Dr. Marcos Roberto Ribeiro.
Trabalho de Conclusão de Curso (graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG, Curso Bacharelado em Engenharia de Computação, 2026.

1. Aplicativo para dispositivos móveis. 2. Locação de imóveis. 3. Estudantes. I. Ribeiro, Marcos Roberto. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG. III. Título.

CDD 342.1454

Henrique Araújo Miranda

DESENVOLVIMENTO DE UM APLICATIVO PARA LOCAÇÃO DIGITAL DE IMÓVEIS VOLTADO PARA USO DE ESTUDANTES EM PEQUENAS CIDADES

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus Bambuí* para obtenção do grau de Bacharel em Engenharia de Computação.

Aprovado em 18 de Maio de 2026 pela banca examinadora:

Prof. Dr. Marcos Roberto Ribeiro – IFMG – *Campus Bambuí* – (Orientador)

Prof.^a Ma. Natalia Camillo do Carmo – IFMG – *Campus Bambuí*

Prof. Dr. Itagildo Edmar Garbazza – IFMG – *Campus Bambuí*



Documento assinado eletronicamente por **Marcos Roberto Ribeiro, Professor**, em 18/05/2026, às 16:42, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Natalia Camillo do Carmo, Professora Substituta**, em 18/05/2026, às 16:42, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Itagildo Edmar Garbazza, Professor**, em 18/05/2026, às 16:42, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2718506** e o código CRC **9A3AF135**.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Gaspar e Nilma, o amor, o apoio incondicional e os ensinamentos que sempre me guiaram, sendo fundamentais para a realização deste sonho. À minha irmã, Nicole, pelo carinho, incentivo e por estar presente durante toda essa caminhada. Ao meu orientador, Dr. Marcos Roberto Ribeiro, pela orientação, dedicação e valiosas contribuições ao longo do desenvolvimento deste trabalho, sendo essencial para a sua concretização. Quero estender meus agradecimentos a todos os professores do IFMG - Campus Bambuí. Por fim, agradeço a todos que, de alguma forma, fizeram parte desta conquista.

“Aqueles que conseguem imaginar qualquer coisa, conseguem criar o impossível.”
Alan Turing

RESUMO

Houve um crescimento do acesso à internet no Brasil e um aumento da demanda por imóveis alugados, especialmente entre jovens em mobilidade acadêmica, em 2023. Entretanto, apesar da existência de soluções digitais voltadas a esse público, foi identificada uma limitação na presença dessas ferramentas em pequenas cidades. Diante desse contexto, este trabalho teve como objetivo digitalizar o mercado imobiliário de aluguel em cidades de menor porte, com foco em estudantes em mobilidade, por meio do desenvolvimento de um aplicativo móvel. Efetuou-se uma análise dos aplicativos mais relevantes do mercado, bem como de trabalhos acadêmicos relacionados, a partir da qual foram definidos os requisitos do sistema. Para a construção da interface e da lógica do aplicativo, foi adotado o *framework* Flutter. Para o gerenciamento de dados, utilizou-se o Firebase, possibilitando a implementação de autenticação de usuários e o armazenamento das informações. Em seguida, efetuaram-se a modelagem do banco de dados distribuído, a prototipação e o desenvolvimento do aplicativo para dispositivos móveis. Ao final, foram realizados testes de mesa para validação das funcionalidades implementadas. A utilização do Firebase mostrou-se adequada para aplicações de pequeno e médio porte, oferecendo suporte eficiente para armazenamento e autenticação com baixo custo inicial. Espera-se que o aplicativo possa contribuir para a inclusão digital no setor imobiliário local, tornando o processo de aluguel mais eficiente e acessível, além de apresentar viabilidade econômica para sua implementação como produto.

Palavras-chave: Aplicativo para Dispositivos Móveis. Locação de Imóveis. Estudantes

ABSTRACT

There has been an increase in internet access in Brazil and a growing demand for rental properties, especially among students in academic mobility in 2023. However, despite the existence of digital solutions aimed at this audience, a limitation was identified in the availability of these tools in small cities. In this context, this work aimed to digitalize the rental real estate market in smaller cities, focusing on students in mobility, through the development of a mobile application. An analysis of the most relevant applications in the market, as well as related academic works, was conducted, from which the system requirements were defined. For the construction of the application's interface and logic, the Flutter framework was adopted. For data management, Firebase was used, enabling the implementation of user authentication and information storage. Subsequently, the distributed database was modeled, and the application was prototyped and developed for mobile devices. Finally, desk testing was carried out to validate the implemented functionalities. The use of Firebase proved to be suitable for small and medium-sized applications, offering efficient support for storage and authentication with low initial cost. It is expected that the application may contribute to digital inclusion in the local real estate sector, making the rental process more efficient and accessible, as well as demonstrating economic feasibility for its implementation as a product.

Keywords: Mobile Application. Property Rental. Students

LISTA DE FIGURAS

Figura 1 - Caso de uso para o usuário	24
Figura 2 - Caso de uso para o locador	25
Figura 3 - Quadro Kanban	29
Figura 4 - Modelo MVC	30
Figura 5 - Modelagem da coleção <i>users</i>	32
Figura 6 - Modelagem da coleção <i>properties</i>	33
Figura 7 - Modelagem da coleção <i>groups</i>	34
Figura 8 - Organização de pastas do projeto	36
Figura 9 - Inicialização Firebase	37
Figura 10 -Gerenciador de rotas para navegação entre telas	38
Figura 11 -Código para login com e-mail e senha	38
Figura 12 -Código para criação de usuário com e-mail e senha	39
Figura 13 -Implementação do <i>widget</i> Google Maps	40
Figura 14 -Diagrama da navegação entre telas	41
Figura 15 -Mostrar imóveis recomendados	42
Figura 16 -Detalhes do imóvel	43
Figura 17 -Exibir favoritos	44
Figura 18 -Login e cadastro	45
Figura 19 -Perfil do usuário	46
Figura 20 -Função de moradia compartilhada	47
Figura 21 -Função de grupos	48
Figura 22 -Exibir notificações	49

LISTA DE QUADROS

Quadro 1 - Funcionalidades comparadas	19
Quadro 2 - Comparativo das funcionalidades entre os trabalhos relacionados . .	21
Quadro 3 - Materiais utilizados no projeto	26

SUMÁRIO

1	Introdução	12
1.1	Objetivos	13
1.2	Justificativa	14
1.3	Resultados esperados	14
1.4	Estrutura do documento	15
2	Fundamentação Teórica	16
2.1	Mercado imobiliário	16
2.2	Desenvolvimento para dispositivos móveis	17
2.3	Metodologia ágil	18
2.4	Comparativo de aplicativos existentes	19
2.5	Estado da arte	19
3	Metodologia	23
3.1	Classificação metodológica	23
3.2	Solução	23
3.3	Materiais e ferramentas	25
3.3.1	<i>Visual studio code</i>	26
3.3.2	<i>Figma</i>	26
3.3.3	<i>Firebase</i>	27
3.3.4	<i>GitHub</i>	27
3.4	Métodos e procedimentos	28
3.4.1	<i>Metodologia de desenvolvimento do trabalho</i>	28
3.4.2	<i>Modelagem do sistema</i>	29

3.4.3	<i>Etapas de desenvolvimento</i>	30
4	Desenvolvimento	31
4.1	Prototipagem	31
4.2	Modelagem do banco de dados	31
4.3	Detalhes do processo de implementação	33
4.3.1	<i>Preparação do ambiente de programação</i>	34
4.3.2	<i>Organização do projeto</i>	35
4.3.3	<i>Widgets</i>	35
4.3.4	<i>Conexão com banco de dados</i>	36
4.3.5	<i>Gerenciador de rotas</i>	37
4.3.6	<i>Função de autenticação</i>	38
4.3.7	<i>Biblioteca Google Maps</i>	39
4.4	Desenvolvimento do aplicativo	40
4.4.1	<i>Navegação entre telas</i>	41
4.4.2	<i>Mostrar imóveis</i>	41
4.4.3	<i>Detalhes dos imóveis</i>	43
4.4.4	<i>Favoritar imóvel</i>	44
4.4.5	<i>Login e cadastro</i>	44
4.4.6	<i>Perfil do usuário</i>	45
4.4.7	<i>Compartilhamento de moradia</i>	46
4.4.8	<i>Notificações</i>	47
4.5	Validação	48
4.6	Implementação para negócio	49
5	Considerações finais	51

5.1	Trabalhos futuros	52
	REFERÊNCIAS	54
	APÊNDICES	59
	APÊNDICE A - Prototipagem no Figma	60
	APÊNDICE B - Dependências do projeto	61
	APÊNDICE C - Implementação <i>Widget</i>	62
	APÊNDICE D - Mapeamento de rotas	63
	APÊNDICE E - Cadastro e login Google	64
E.1	Login usando Google	64
E.2	Criação de usuário com Google	65
	APÊNDICE F - Geolocalização Google Maps	66

1 INTRODUÇÃO

O mundo atual se apresenta cada vez mais digital. Diversas tecnologias têm como objetivo facilitar o dia a dia das pessoas, contribuindo para melhorar sua qualidade de vida e tornar suas atividades rotineiras menos exaustivas. Dados da Pesquisa Nacional por Amostra de Domicílios Contínua, no módulo de Tecnologia da Informação e Comunicação, realizada em 2023 pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostram que 92,5% dos domicílios brasileiros tinham acesso à internet, o que representa um total de 72,5 milhões de residências. Nas áreas urbanas, esse percentual passou de 93,5% para 94,1%, enquanto, nas áreas rurais, aumentou de 78,1% para 81,0%. Ainda segundo a pesquisa, a proporção de domicílios com telefone celular aumentou desde 2016, passando de 93,1% para 96,7% em 2023 (IBGE, 2023).

Paralelamente, o setor imobiliário brasileiro, especialmente na área de aluguel de imóveis, está em crescimento. Em 2000, 12,3% da população viviam em imóveis alugados. Esse número subiu para 16,4% em 2010 e atingiu 20,9% em 2022, demonstrando uma demanda crescente por locações e um mercado com potencial a ser explorado (SIQUEIRA; BRITTO, 2024).

No contexto educacional, o Brasil contava, em 2021, com cerca de 8,9 milhões de estudantes matriculados em cursos de graduação e sequenciais, segundo o Censo da Educação Superior, realizado pelo Inep. O estudo também aponta que 11% dos estudantes ingressantes em cursos da rede federal se mudaram de cidade para estudar, o que evidencia a necessidade de soluções de moradia para esse público específico (DINO, 2023).

Apesar da existência de plataformas digitais voltadas à locação de imóveis, consolidadas no mercado, como QuintoAndar¹, Zap Imóveis² e Airbnb³, essas soluções concentram-se majoritariamente em grandes centros urbanos, deixando uma lacuna importante em cidades de menor porte. Nessas localidades, é comum que o processo de aluguel ainda ocorra de maneira informal, com anúncios físicos e redes pessoais, o que pode dificultar o acesso de estudantes de outras regiões e limitar a transparência e segurança nas negociações (CASTRO; LESO; CORTIMIGLIA, 2020).

Além disso, dados da Fundação João Pinheiro (FJP) mostram que o déficit habitacional brasileiro atinge de forma significativa municípios pequenos e médios, onde a ausência de infraestrutura digital nos serviços de habitação ainda é uma realidade. Ao mesmo tempo, muitas dessas cidades abrigam instituições federais ou estaduais de ensino superior, como o Instituto Federal de Educação, Ciência e Tecno-

¹ <https://www.quintoandar.com.br/>

² <https://www.zapimoveis.com.br/>

³ <https://www.airbnb.com.br/>

logia de Minas Gerais (IFMG), Universidade Federal dos Vales Jequitinhonha e Mucuri (UFVJM), e Universidade Federal de Ouro Preto (UFOP), que atraem milhares de estudantes todos os anos. Esse público, geralmente jovem e habituado ao uso de dispositivos móveis, encontra dificuldades para buscar moradia de forma remota e segura nesses contextos menos digitalizados (FJP, 2024).

O conceito de “cidade pequena” não apresenta uma definição única e consensual na literatura, havendo variações significativas entre os autores quanto ao critério populacional. Pereira (2007) considera como pequenas as cidades com até 20 mil habitantes, enquanto Bernardelli (2004) amplia esse limite para 30 mil. Corrêa (1999), por sua vez, define como pequenas as cidades com até 50 mil habitantes. Já o IBGE adota uma classificação mais abrangente, considerando como pequenas todas as cidades com população inferior a 50 mil habitantes (VIEIRA; ROMA; MIYAZAKI, 2020). Para os fins deste trabalho, optou-se por adotar o critério adotado pelo IBGE, estabelecendo como limite máximo o número de 50 mil habitantes.

Diante desse cenário, o presente trabalho propôs o desenvolvimento de um aplicativo para dispositivos móveis voltado à locação de imóveis. A plataforma permitirá que os usuários visualizem imóveis disponíveis para aluguel e realizem todo o processo de locação de forma digital. Considerando que é comum estudantes buscarem novas moradias ao ingressarem no ensino superior, o aplicativo contará com uma funcionalidade voltada especificamente para esse público, a qual permite a busca de moradias compartilhadas por mais de uma pessoa, além de também permitir encontrar outros usuários para o compartilhamento de um imóvel.

1.1 Objetivos

O objetivo geral do presente trabalho foi digitalizar o mercado imobiliário para aluguel em cidades de menor porte, voltado para estudantes em mobilidade, por meio do desenvolvimento de um aplicativo móvel. Para alcançar o objetivo geral, foram necessários os seguintes objetivos específicos:

- analisar as principais funcionalidades dos aplicativos de destaque no mercado imobiliário no Brasil;
- levantar os requisitos do projeto;
- projetar o sistema;
- implementar o sistema.

1.2 Justificativa

De acordo com Freitas (2020), a globalização, os avanços nos meios de comunicação e a diversidade de recursos digitais têm impulsionado as empresas a buscarem, cada vez mais, a excelência em suas atividades, seja na entrega de produtos, na prestação de serviços ou no desenvolvimento de soluções inovadoras. Nesse contexto de constantes transformações, os mercados tornam-se progressivamente mais competitivos. O consumidor moderno, por sua vez, adota uma postura mais criteriosa e exigente no momento da decisão de compra. No setor imobiliário, essa realidade não é diferente. Empresas e profissionais que não acompanharem essa evolução correm o risco de perder relevância ou, até mesmo, desaparecer do mercado.

Segundo a pesquisa de Cordeiro, Baggio e França (2019) com consumidores, 58% dos participantes manifestaram a intenção de adquirir um imóvel nos próximos cinco anos. Por outro lado, 18% dos entrevistados disseram preferir viver em um imóvel alugado, independentemente de sua situação financeira. De acordo com o levantamento, a principal razão para essa preferência é a flexibilidade proporcionada pelo aluguel, que permite mudanças de residência com mais facilidade. Esses dados revelam que, além daqueles que optam pelo aluguel por necessidade, há uma parcela significativa da população que vê essa modalidade como uma escolha alinhada a um estilo de vida mais livre, voltado a viagens, empreendedorismo e novas experiências. Por meio da pesquisa, constatou-se, também, que a internet se configura como o canal prioritário para o início do processo de compra ou aluguel.

Diante desse contexto, torna-se perceptível a necessidade de adaptação do setor imobiliário às novas exigências e comportamentos dos consumidores. A digitalização dos processos de locação configura-se como uma resposta eficaz ao perfil contemporâneo de cliente, que prioriza agilidade e praticidade. Os estudantes, em especial, são significativamente beneficiados, uma vez que, na maioria das vezes, não possuem familiaridade com a nova cidade. Dessa forma, o uso de tecnologias digitais proporciona maior facilidade na busca por moradia adequada às suas necessidades.

A criação de soluções tecnológicas voltadas à intermediação de aluguéis, sobretudo em cidades de menor porte, que não possuem soluções bem difundidas, representaria uma otimização dos serviços imobiliários nestas cidades, agilizando todo o processo envolvido. Nesse sentido, este projeto propõe o desenvolvimento de um aplicativo móvel que ofereça uma experiência eficiente e acessível.

1.3 Resultados esperados

Espera-se que este trabalho resulte em uma digitalização do mercado imobiliário para aluguel em cidades de menor porte, em específico, a cidade de Bambuí,

situada em Minas Gerais, a qual foi escolhida devido à presença de um campus do IFMG, contendo, assim, uma demanda estudantil e também a baixa presença de soluções tecnológicas similares. Caso seja bem-sucedido, espera-se que a digitalização também possa ser adotada por outras cidades, por meio do uso do aplicativo móvel.

Para o cliente, espera-se uma significativa melhoria na experiência de busca e divulgação de imóveis para locação nas regiões aplicadas. Através do aplicativo, os locadores poderão expor seus imóveis de maneira eficiente, com fotos, descrições detalhadas, localização no mapa e filtros personalizados que aumentam a visibilidade. Por outro lado, os locatários, em especial, os estudantes, poderão ter acesso rápido e organizado às opções de moradia compatíveis com suas circunstâncias, compartilhando uma moradia.

1.4 Estrutura do documento

O restante do trabalho está estruturado como se segue. No Capítulo 2, são apresentados os conceitos relacionados ao mercado imobiliário, destacando sua relevância e a influência da tecnologia nesse contexto, além de abordar o desenvolvimento de aplicações móveis, metodologias ágeis e uma análise de soluções existentes alinhadas à proposta do trabalho.

No Capítulo 3, efetuam-se a caracterização da pesquisa quanto à sua classificação, bem como a descrição dos materiais, ferramentas e métodos empregados no desenvolvimento do estudo.

No Capítulo 4, são apresentados o desenvolvimento e os resultados do trabalho e também do aplicativo, a partir da aplicação dos métodos e ferramentas adotados. No capítulo 5, são expostas as conclusões do trabalho e os próximos passos a serem desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma revisão bibliográfica com o objetivo de expor o estado da arte relacionado ao tema em estudo. Inicialmente, na Seção 2.1, são abordados os conceitos referentes ao funcionamento e às particularidades do mercado imobiliário. Em seguida, a Seção 2.2 trata das especificidades do desenvolvimento de aplicativos móveis multiplataforma. Posteriormente, na Seção 2.3, discute-se a aplicação de metodologias ágeis no contexto de projetos de software.

Na sequência, a Seção 2.4 apresenta um comparativo das funcionalidades de aplicativos já existentes no mercado. Por fim, na Seção 2.5, é apresentado o estudo do estado da arte, consolidando as principais abordagens e contribuições relacionadas ao tema.

2.1 Mercado imobiliário

O mercado imobiliário representa um setor estratégico da economia, cuja estrutura é composta por um conjunto de mecanismos interdependentes voltados à criação, transferência, gestão e financiamento de bens imóveis (GEIPELE; KAUŠKALE, 2013; ROULAC, 1996).

O mercado de aluguel de imóveis tem se consolidado como um componente estratégico das dinâmicas urbanas contemporâneas, refletindo transformações nas formas de habitar, no acesso à moradia e nas políticas públicas habitacionais. Além de funcionar como alternativa à aquisição da casa própria, o aluguel tornou-se central na organização socioespacial das cidades, influenciando aspectos econômicos, sociais e territoriais (LIU *et al.*, 2020).

No Brasil, o mercado imobiliário de aluguel desempenha papel fundamental na dinâmica socioeconômica brasileira, especialmente diante dos desafios de acesso à moradia enfrentados por parcelas significativas da população. Segundo o Instituto de Pesquisa Econômica Aplicada, aproximadamente 18% dos domicílios urbanos são ocupados por inquilinos (IPEA, 2019). Além disso, o déficit habitacional, que ultrapassa 6,2 milhões de unidades, reforça a necessidade de alternativas flexíveis como o aluguel.

O mercado imobiliário de aluguel, no Brasil, também tem demonstrado um crescimento notável, com os preços de locação subindo 13,5% em 2024, superando a inflação oficial do período (BRASIL, 2024). Essa valorização reflete um setor dinâmico, onde a rentabilidade de imóveis residenciais alcançou 19,1% em 2024, segundo estudos recentes (CNN BRASIL, 2024). Além da dinâmica de mercado, a pesquisa de Carvalho e Ruas (2017) sobre o tema reforça a sua relevância econômica, mostrando que os gastos com aluguel têm um peso significativo no Índice Nacional de Preços ao

Consumidor Amplo. O crescimento do setor também se reflete no mercado financeiro, com o aumento substancial do volume de emissões de Fundos de Investimento Imobiliário nos últimos anos, que cresceu de R\$ 13,41 bilhões, em 2018, para R\$ 41,4 bilhões em 2019 (VARTANIAN *et al.*, 2022).

A tecnologia vem transformando o mercado imobiliário ao tornar o processo de compra e locação de imóveis mais ágil e eficiente. Por meio de recursos baseados em algoritmos, é possível automatizar etapas como a comunicação entre as partes, burocracias, precificação e atração de mais consumidores. Além disso, ferramentas digitais facilitam o acompanhamento do desempenho da imobiliária e o acesso a informações por meio de telas digitais e bancos de dados organizados (CASTRO; LESO; CORTIMIGLIA, 2020).

2.2 Desenvolvimento para dispositivos móveis

Os dispositivos móveis assumiram papel central no cotidiano contemporâneo, sobretudo em virtude de sua versatilidade e ampla gama de funcionalidades. Apesar da existência de diversos sistemas operacionais voltados a esses dispositivos, o Android e o iOS destacam-se como as plataformas predominantes. Conforme dados da IDC (2025), o sistema operacional Android está presente em 83,8% dos dispositivos móveis globalmente, enquanto o iOS detém uma participação de 19,5% no mercado. Diante desse cenário, para o desenvolvimento de aplicativos, é fundamental adotar abordagens que permitam a criação de soluções compatíveis com múltiplas plataformas, de modo a alcançar o maior número possível de usuários.

O desenvolvimento nativo de aplicações apresenta vantagens significativas no que se refere ao desempenho e à integração com funcionalidades específicas de cada sistema operacional, além de proporcionar uma experiência de usuário alinhada com os padrões da plataforma de destino. Entretanto, essa abordagem requer domínio de tecnologias distintas para cada sistema, o que resulta em um aumento na complexidade, no tempo e nos custos do processo de desenvolvimento. Dadas essas limitações, observa-se uma crescente adoção de *frameworks* multiplataformas, os quais possibilitam a construção de uma única base de código capaz de ser executada em diferentes sistemas operacionais, promovendo, assim, maior eficiência no desenvolvimento e maior abrangência ao público-alvo da aplicação (EL-KASSAS *et al.*, 2017).

O Flutter¹ é um kit de desenvolvimento *open-source* criado pelo Google que permite a construção de aplicações nativas para dispositivos móveis Android e iOS, além do desenvolvimento de softwares para Windows, Mac, Linux e *Web* a partir de

¹ <https://flutter.dev/>

uma única base de código. O Flutter utiliza a linguagem de programação Dart² e fornece um mecanismo de renderização próprio, o que permite controle total sobre cada pixel na tela, resultando em interfaces personalizáveis e com desempenho otimizado (FLUTTER, 2025).

A estrutura do Flutter fundamenta-se na utilização de *widgets*, termo derivado da junção das palavras *window* (janela) e *gadget* (dispositivo). Esses componentes representam elementos da interface gráfica, abrangendo desde textos e botões até a definição do *layout*. Assim, os *widgets* constituem a base da construção da interface do usuário, facilitando a composição, a reutilização e a customização de componentes visuais. Dessa forma, o Flutter propõe uma abordagem eficiente para o desenvolvimento multiplataforma, mantendo a consistência visual e o desempenho nativo das aplicações (MEILLER, 2022).

2.3 Metodologia ágil

Com o início da filosofia proposta pelo Manifesto Ágil, em 2001, as metodologias ágeis de desenvolvimento de software passaram a ser amplamente adotadas no setor de tecnologia, tanto por grandes quanto por pequenas empresas desenvolvedoras. Essa abordagem surgiu como uma alternativa às metodologias tradicionais, oferecendo uma resposta às limitações observadas nos modelos convencionais de desenvolvimento (ARAÚJO; GORGÔNIO; VALE, 2014).

A adoção dessa metodologia, bem como a entrega contínua de funcionalidades e a capacidade de responder rapidamente a mudanças, permitem que cada modificação no sistema seja orientada à maximização do valor entregue ao usuário final. Além disso, a interação frequente entre a equipe de desenvolvimento e os clientes contribui diretamente para o refinamento contínuo da solução. Essa colaboração estreita assegura que o produto final esteja alinhado, de forma precisa, às necessidades do projeto, promovendo maior eficiência às expectativas estabelecidas (BARBOZA *et al.*, 2016; TSUI; KARAM; BERNAL, 2013).

Estudos destacam os benefícios dessas metodologias, como o aumento da produtividade e a melhoria da qualidade do *software* LOPES *et al.* (2024), além de promoverem uma cultura colaborativa que otimiza a eficiência organizacional (SILVA; GRACIANO, 2022). A adoção de práticas ágeis, como o Scrum e o Kanban, por exemplo, tem se mostrado eficaz para gerenciar projetos complexos, lidando com a imprevisibilidade por meio de ciclos iterativos e entregas incrementais (OLIVEIRA, 2020).

² <https://dart.dev/>

2.4 Comparativo de aplicativos existentes

Para a obtenção dos requisitos do aplicativo, foram analisados os principais aplicativos na área abordada pelo trabalho, com base em popularidade e pelo número de *downloads* nas principais lojas digitais. Assim, foi realizada uma comparação de algumas das suas funcionalidades com a solução proposta, como pode ser observado no Quadro 1. Utilizando-se o quadro, foi possível identificar quais as funcionalidades mais relevantes e quais não são adotadas pelos aplicativos analisados, permitindo o desenvolvimento de um aplicativo que mantenha um padrão de qualidade e apresente diferenciais em relação às soluções existentes.

Quadro 1 – Funcionalidades comparadas

Aplicativos	Busca por filtros	Integração com mapas	Chat com locador	Voltado para locação mensal	Navegar sem cadastro	Quarto individual	Favoritar imóvel
Zap Imóveis	X	X	X	X	X		X
QuintoAndar	X	X	X	X	X		X
Airbnb	X	X	X		X		X
VivaReal	X	X	X		X		X
Imovelweb	X	X	X	X	X		X
HousingAnywhere	X	X	X		X	X	X

Fonte: Elaborado pelo autor, 2026.

As comparações foram realizadas manualmente, sob a perspectiva de um usuário comum, buscando identificar e registrar os principais aspectos que contribuem para uma experiência de uso satisfatória. Além disso, é importante ressaltar que a análise foi conduzida utilizando-se as versões disponíveis dos aplicativos apresentados no Quadro 1, permitindo uma avaliação prática de suas funcionalidades, interfaces e recursos oferecidos aos usuários.

2.5 Estado da arte

O artigo de Dhamodaran *et al.* (2023) apresenta o desenvolvimento de um sistema de aluguel de imóveis baseado em uma aplicação *web* e móvel. Possui fun-

cionalidades como pesquisa de imóveis, GPS integrado, mensagens pela plataforma, pagamentos integrados, gerenciamento de anúncios, mecanismos de autenticação e criptografia. Utiliza tecnologias como Flutter, Node.js, React, Firebase, MongoDB e AWS S3. A aplicação adota o modelo de desenvolvimento em cascata, com arquitetura voltada à usabilidade e segurança. O sistema é projetado para permitir aos usuários, mediante assinatura, o acesso a propriedades *premium*, favoritar anúncios e interagir em tempo real.

O estudo de Kunwar *et al.* (2023) descreve o desenvolvimento de um aplicativo Android voltado à locação de quartos e residências. A aplicação foi implementada utilizando-se o software Android Studio, assim como Java, XML e Firebase. Oferece funcionalidades como cadastro de usuários e proprietários, localização geográfica dos imóveis via mapa, gerenciamento de dados administrativos e visualização detalhada dos imóveis disponíveis pelos usuários e proprietários. A plataforma conta com três painéis: usuário, proprietário e administrador.

O trabalho de Voumick *et al.* (2021) demonstra o desenvolvimento de uma aplicação *web* inteligente para aluguel de residências, voltada à realidade urbana de Bangladesh. A solução oferece funcionalidades como autenticação de usuários, comunicação segura por *chat*, integração com o Google Maps para localização dos imóveis e um banco de dados estruturado para gerenciamento de perfis, anúncios e mensagens. A arquitetura do sistema segue o modelo de desenvolvimento em cascata e utiliza tecnologias como: HTML/CSS, PHP, Laravel, JavaScript, AJAX e MySQL. Além disso, apresenta viabilidade financeira com monetização via anúncios pagos.

Manalu *et al.* (2016) realizaram o desenvolvimento e a avaliação de um aplicativo móvel voltado à divulgação de informações sobre quartos para aluguel, incorporando os recursos de *chat* integrado e notificações por *push*. A arquitetura do sistema é baseada em cliente-servidor, com comunicação por REST *web services*, para dados gerais; XMPP, para o serviço de mensagens em tempo real; e *Google Cloud Messaging*, para o serviço de notificação *push*. As funcionalidades principais incluem listagem e visualização detalhada dos quartos, mapa com localização via Google Maps, envio de avaliações, *chat* direto com o administrador do imóvel e sistema de *broadcast* para notificar mudanças na disponibilidade dos quartos. A avaliação do sistema, por meio de questionários com usuários e entrevistas com gerentes de imóveis, demonstrou alta aceitação e usabilidade da aplicação.

O artigo de Milkovich *et al.* (2020) apresenta um aplicativo móvel voltado para estudantes universitários e locatários de baixa renda. Adota uma abordagem centrada no usuário, oferecendo uma interface de navegação por gestos e um sistema de recomendação personalizado. Esse sistema é desenvolvido com o uso do Flutter e MongoDB, além de utilizar técnicas de aprendizado profundo, incluindo redes neurais convolucionais para classificação de imagens de imóveis, e algoritmos de recomen-

dação baseados no histórico de visualizações dos usuário, proporcionando sugestões alinhadas às preferências individuais.

O projeto de Biswas (2024) propõe o desenvolvimento de um aplicativo móvel de aluguel voltado para imigrantes e estudantes internacionais no Canadá. A proposta é centrada em três pilares: simplicidade de *design*, usabilidade e impacto emocional. O estudo emprega metodologias de *design thinking*, prototipagem com Figma, *wireframing*, análise de concorrência e criação de personas, resultando em uma solução que inclui verificação de identidade, mensagens integradas, agendamento de visitas, sistema de reputação e validação obrigatória das informações dos anúncios. Além disso, o autor aplica os princípios de experiência de usuário e estratégias de *design* para estimular o engajamento e retenção de usuários.

Com base nos trabalhos analisados, observa-se um consenso quanto à relevância de soluções digitais voltadas à otimização do processo de locação de imóveis. Os estudos evidenciam diferentes abordagens e metodologias para suprir lacunas de usabilidade, segurança e acessibilidade em plataformas de aluguel. O Quadro 2 traz uma comparação das funcionalidades e tecnologias dos principais trabalhos relacionados identificados. Apesar da diversidade de públicos e tecnologias utilizadas, todos os projetos compartilham a premissa de que a digitalização do setor imobiliário pode melhorar significativamente a experiência de locadores e locatários.

Quadro 2 – Comparativo das funcionalidades entre os trabalhos relacionados

Autores	Busca por filtros	Integração com mapas	Chat com o locador	Pagamento	Cadastro de usuário	Sistema de reputação	Quarto individual	Dividir imóvel com usuários
(DHAMODARAN <i>et al.</i> , 2023)	X	X	X	X	X			
(KUNWAR <i>et al.</i> , 2023)	X	X			X		X	
(VOUMICK <i>et al.</i> , 2021)	X	X	X		X			
(MANALU <i>et al.</i> , 2016)	X	X	X		X	X	X	
(MILKOVICH <i>et al.</i> , 2020)	X				X			
(BISWAS, 2024)			X		X	X		
Trabalho desenvolvido	X	X			X	X	X	X

Fonte: Elaborado pelo autor, 2026.

O presente trabalho posiciona-se como uma extensão das iniciativas dos

trabalhos analisados, aplicando conceitos consolidados em um contexto ainda pouco explorado, com potencial de impacto social e econômico. Conforme apresentado no Quadro 2, observa-se que a funcionalidade de compartilhamento de imóveis entre usuários não é contemplada nos estudos analisados. Dessa forma, essa característica pode ser considerada um diferencial da aplicação desenvolvida neste trabalho, uma vez que busca atender a uma necessidade específica de estudantes que desejam dividir despesas de moradia.

Por outro lado, optou-se por não implementar funcionalidades como chat direto com o locador e processamento de pagamentos pelo aplicativo. Essa decisão foi tomada com o objetivo de reduzir a complexidade do projeto, concentrando os esforços no desenvolvimento e na validação das funcionalidades consideradas essenciais para os objetivos propostos.

3 METODOLOGIA

Este capítulo descreve como foram realizados os processos de desenvolvimento do presente trabalho. A Seção 3.1 apresenta a classificação da pesquisa. A Seção 3.2 explica a solução proposta para resolver o problema identificado. Já a Seção 3.3 aborda os materiais e as principais tecnologias utilizadas no desenvolvimento. Por fim, na Seção 3.4, são apresentados os métodos e os procedimentos adotados.

3.1 Classificação metodológica

A metodologia adotada para o desenvolvimento deste trabalho, de acordo com Wazlawick (2009), caracteriza-se como uma pesquisa aplicada, uma vez que busca solucionar um problema concreto: a ausência de soluções tecnológicas eficientes para locação de imóveis voltadas a estudantes em cidades de pequeno porte, ao propor a criação de um aplicativo móvel que visa gerar impacto positivo na realidade observada.

Quanto aos objetivos, segundo os critérios estabelecidos por Gehard e Silveira (2009), esta pesquisa é classificada como descritiva e exploratória. O caráter descritivo refere-se à análise das funcionalidades de aplicativos similares já disponíveis no mercado e das necessidades específicas do público-alvo. Por sua vez, a dimensão exploratória está presente na tentativa de compreender um cenário ainda pouco investigado no contexto da locação digital em regiões com menor acesso tecnológico, possibilitando a identificação de oportunidades de inovação.

Em relação aos procedimentos metodológicos, o presente trabalho caracteriza-se como uma pesquisa-ação, pois pressupõe uma participação planejada do pesquisador na situação problemática a ser investigada (GEHARD; SILVEIRA, 2009). Portanto, o objetivo foi transformar uma realidade observada, no caso, a falta de soluções tecnológicas, do mercado imobiliário em cidades de pequeno porte.

Por fim, de acordo com Gehard e Silveira (2009), a abordagem metodológica adotada é quali-quantitativa. A dimensão qualitativa representa a compreensão das experiências e expectativas dos usuários envolvidos no processo de locação. Já a dimensão quantitativa diz respeito à análise de dados numéricos obtidos por meio de testes do aplicativo, permitindo identificar padrões, frequências e correlações que complementam a interpretação dos resultados qualitativos.

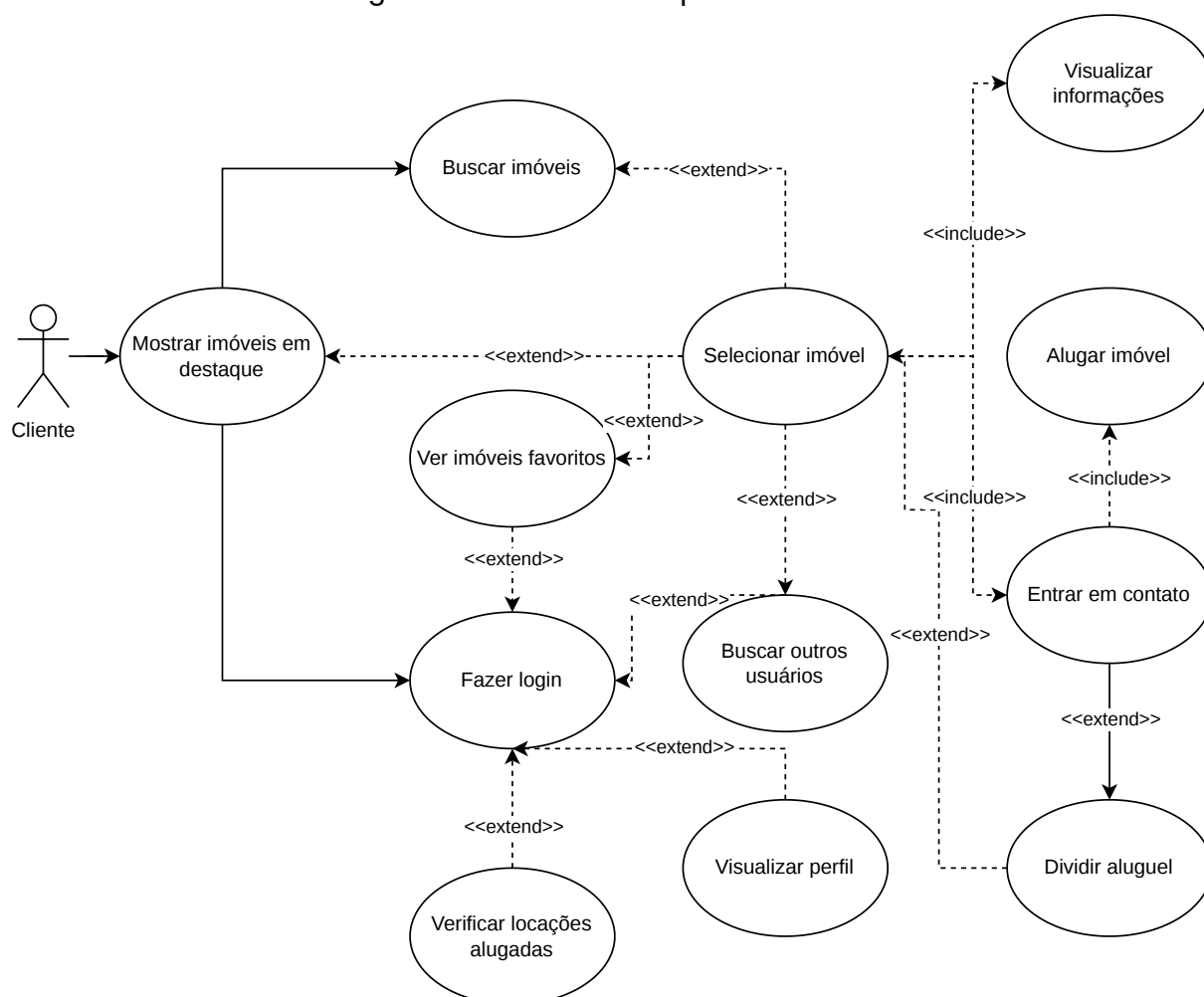
3.2 Solução

O diagrama da Figura 1 demonstra, por meio de um caso de uso, os passos da interação do usuário com a solução. Na tela inicial, apresentam-se imóveis

em destaque, com base no critério de popularidade e proximidade. Em seguida, é possível realizar buscas personalizadas por meio de filtros como localização, faixa de preço, tipo de imóvel e disponibilidade. Ao se selecionar um imóvel, tem-se a visualização detalhada, contendo galeria de fotos, descrição completa, localização via mapa e dados de contato do locador. Caso haja interesse, é possível entrar em contato diretamente com o anunciante por meio de aplicativo externo.

Para atender à demanda estudantil, o aplicativo também disponibiliza uma área específica voltada à locação conjunta, permitindo a busca por quartos individuais e perfis de outros estudantes interessados em dividir moradia, com possibilidade de interação direta. Para visualizar o perfil e acessar a área de locação conjunta, é preciso possuir uma conta registrada e realizar o login. As demais funções do aplicativo funcionam sem necessidade da ação de cadastro e login.

Figura 1 – Caso de uso para o usuário

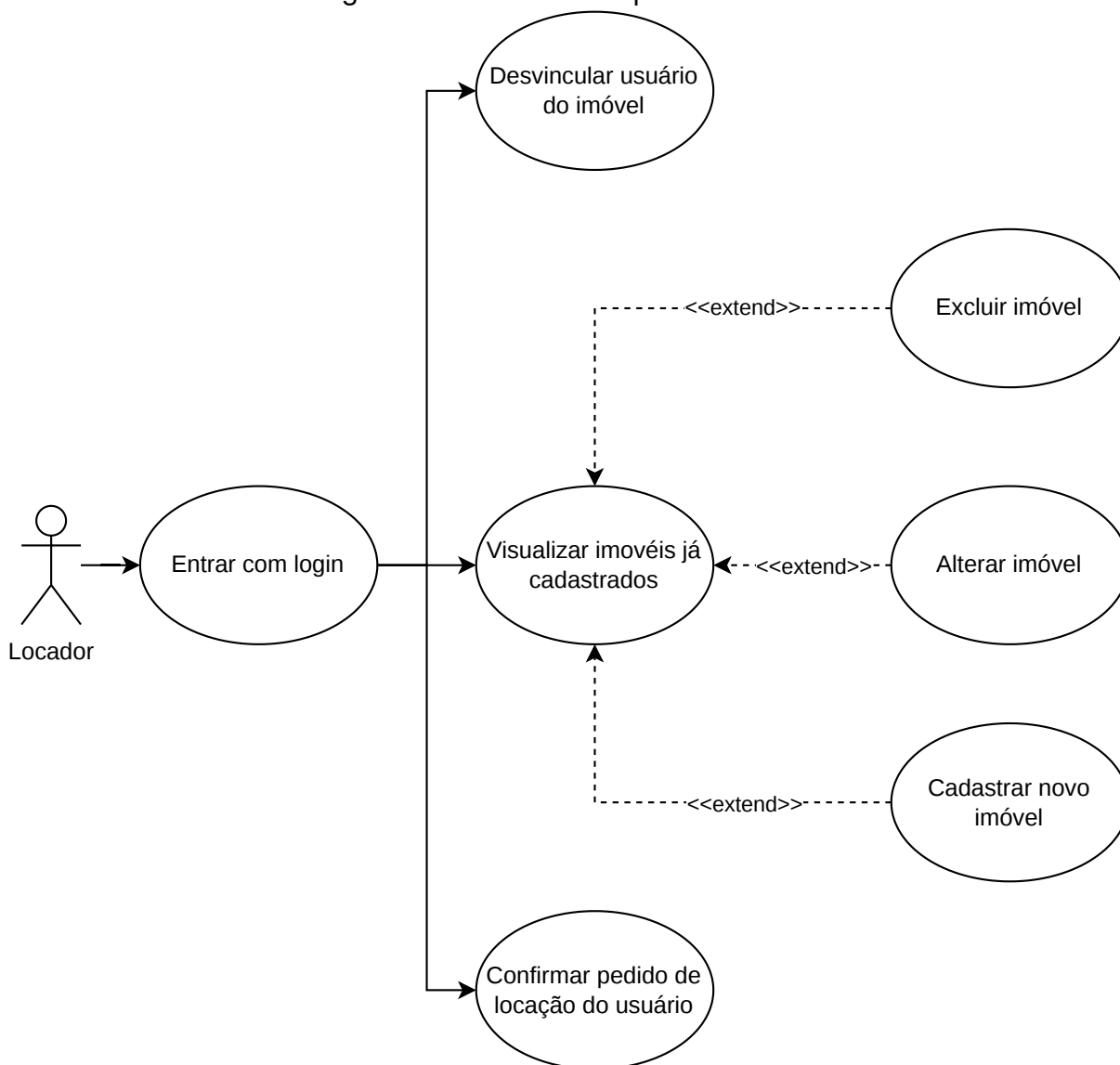


Fonte: Elaborado pelo autor, 2026.

Já o diagrama da Figura 2 demonstra os passos da interação do locador com a solução. O login é obrigatório para inicialização da aplicação, devido ao fato de os imóveis serem vinculados a cada locador. Na tela inicial, é possível visualizar imó-

veis cadastrados, assim como efetuar as ações de excluir, alterar e cadastrar imóveis. Por fim, é possível aceitar que um usuário alugue seu imóvel e também desvincular usuários de suas alocações.

Figura 2 – Caso de uso para o locador



Fonte: Elaborado pelo autor, 2026.

3.3 Materiais e ferramentas

Esta seção trata dos materiais e ferramentas usados no desenvolvimento do presente trabalho. O material utilizado para o desenvolvimento do aplicativo foi somente o computador pessoal, que possui as configurações expostas no Quadro 3. Para o desenvolvimento, utilizaram-se as seguintes ferramentas, bem como suas versões:

- Visual Studio Code 1.102.0 (<https://code.visualstudio.com>);

- Flutter 3.35.3 (<https://flutter.dev>);
- Dart 3.8 (<https://dart.dev>);
- Figma (<https://figma.com>);
- Github (<https://github.com>);
- Firebase (<https://firebase.google.com>).

Quadro 3 – Materiais utilizados no projeto

Processador	AMD Radeon Ryzen 5 5600GT
Processador Gráfico	AMD Radeon RX 7600
Memória RAM	16 GB, 3000MHz, DDR4
Unidade de Estado Sólido	1 TB
Sistema operacional	Windows 11 Pro 64 bits 24H2

Fonte: Elaborado pelo autor, 2026.

3.3.1 Visual studio code

O Visual Studio Code é um editor de código-fonte leve, multiplataforma e extensível. Desenvolvido pela Microsoft, oferece suporte nativo a diversas linguagens de programação, incluindo Dart, utilizada neste projeto. No contexto deste trabalho, foi empregado como ambiente de desenvolvimento principal, oferecendo praticidade na organização do código, facilidade na navegação entre arquivos do projeto e integração eficiente com ferramentas externas como Github e Firebase (CODE, 2025).

3.3.2 Figma

O Figma é uma ferramenta de *design* de interfaces gráficas e prototipação, amplamente utilizada no desenvolvimento de aplicações *web* e móveis. Além disso, a plataforma disponibiliza recursos avançados para criação de *wireframes*, protótipos interativos, componentes reutilizáveis e sistemas de *design* consistentes (FIGMA, 2025).

No contexto deste trabalho, o Figma foi utilizado para o desenvolvimento das interfaces das telas do aplicativo, permitindo a visualização e validação antecipadas da experiência do usuário antes da implementação prática, visando à elaboração de telas funcionais, acessíveis e alinhadas às necessidades do público-alvo.

3.3.3 Firebase

O Firebase é uma plataforma *Backend-as-a-Service* desenvolvida pelo Google que fornece uma série de serviços para desenvolvimento rápido de aplicativos móveis e *web*. Sua proposta visa eliminar a necessidade de configuração e manutenção de servidores próprios, permitindo que desenvolvedores foquem diretamente na lógica da aplicação (FIREBASE, 2025; MALLIK *et al.*, 2020).

Entre os principais serviços do Firebase utilizados neste projeto, destaca-se o *Cloud Firestore*, um banco de dados não relacional em tempo real. Esse serviço permite a leitura, escrita e sincronização de dados de forma imediata entre os dispositivos conectados, o que o torna especialmente adequado para aplicações que exigem atualização dinâmica de conteúdo. Além disso, o Firestore possibilita a definição de regras de segurança baseadas em autenticação e permissões específicas, garantindo controle de acesso e proteção às informações armazenadas.

Outro recurso adotado neste trabalho foi o *Firebase Authentication*, sistema de autenticação gerenciado pela própria plataforma Firebase. Esse serviço oferece suporte tanto para métodos de login simples, como autenticação por e-mail e senha, quanto para soluções mais avançadas, incluindo login por contas Google. Além de facilitar a integração da autenticação ao aplicativo, o serviço garante o armazenamento seguro das credenciais dos usuários, reduzindo a complexidade e aumentando a confiabilidade do processo de login. Por fim, o *Storage* é responsável pelo armazenamento seguro de imagens para uso em aplicações.

3.3.4 GitHub

O GitHub é uma plataforma de hospedagem de código-fonte baseada no sistema de controle de versão Git. Ele permite o versionamento distribuído de projetos, colaborando com a rastreabilidade, a colaboração remota e a organização de projetos de software.

O principal benefício proporcionado por esta ferramenta, no contexto deste trabalho, é a possibilidade de acompanhar a evolução do código ao longo do tempo, por meio do controle de versões. Essa característica não apenas contribui para a organização, mas também facilita eventuais manutenções e extensões do sistema em trabalhos futuros. Considerando que o desenvolvimento da aplicação será realizado por um único autor, a adoção de múltiplos ramos não se mostrou necessária neste caso. Dessa forma, optou-se pela utilização do modelo de desenvolvimento denominado *Trunk Based Development*, que consiste na construção contínua diretamente na principal linha de código.

3.4 Métodos e procedimentos

A presente seção explica o desenvolvimento do trabalho, descrevendo os passos seguidos, bem como o detalhamento de como foram feitos. A Seção 3.4.1 demonstra qual a principal metodologia utilizada. A Seção 3.4.2 explica a escolha da arquitetura do processo de desenvolvimento e como foi implementada. Por fim, a Seção 3.4.3 mostra as etapas seguidas no desenvolvimento.

3.4.1 Metodologia de desenvolvimento do trabalho

Para o desenvolvimento deste trabalho, adotou-se a metodologia Kanban, que é um método de gerenciamento visual de tarefas que teve origem no sistema de produção da Toyota, sendo posteriormente adaptado para a área de desenvolvimento de software. No contexto de metodologias ágeis, o Kanban se destaca por oferecer uma visão clara e instantânea do fluxo de trabalho por meio de quadros com colunas, representando os diferentes estágios de progresso de uma tarefa (ZAYAT; SENVAR, 2020). A configuração padrão de um quadro Kanban apresenta uma divisão em três colunas, sendo elas:

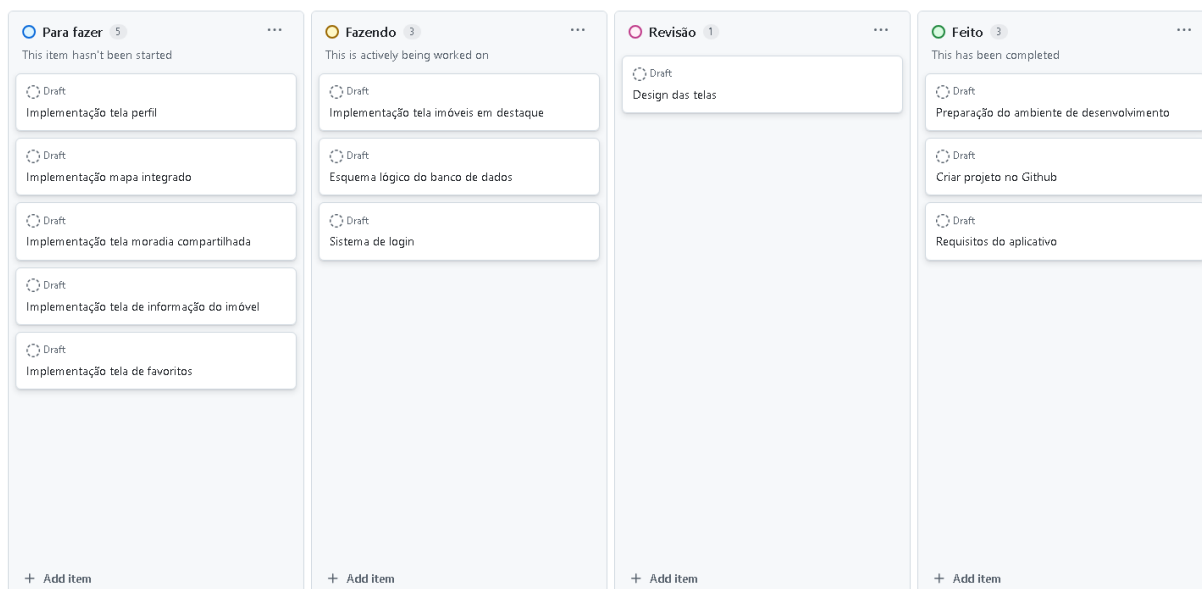
- *To do* (Para fazer);
- *Doing* (Fazendo);
- *Done* (Feito).

O quadro Kanban pode ser personalizado em diversas colunas, caso o projeto tenha tal necessidade. Sendo assim, é extremamente flexível e facilmente aplicável em diversos contextos e projetos.

No desenvolvimento da aplicação e na elaboração deste trabalho, a metodologia Kanban foi empregada para a gestão visual do fluxo de tarefas. Esta abordagem permitiu verificar, de forma objetiva, o progresso das funcionalidades, bem como otimizar a ordem de execução das tarefas propostas.

Para sua implementação, foi utilizado o GitHub, que, além das funções de versionamento e hospedagem de repositórios, possui um quadro Kanban. A Figura 3 apresenta o quadro adotado para o desenvolvimento do trabalho, com as colunas "Para fazer", "Fazendo", "Revisão" e "Feito". Cada tarefa pode ser movida a qualquer momento para outras colunas, alterando seu estado. Quando todas as tarefas chegarem à coluna "Feito", o trabalho estará concluído.

Figura 3 – Quadro Kanban



Fonte: Elaborado pelo autor, 2026.

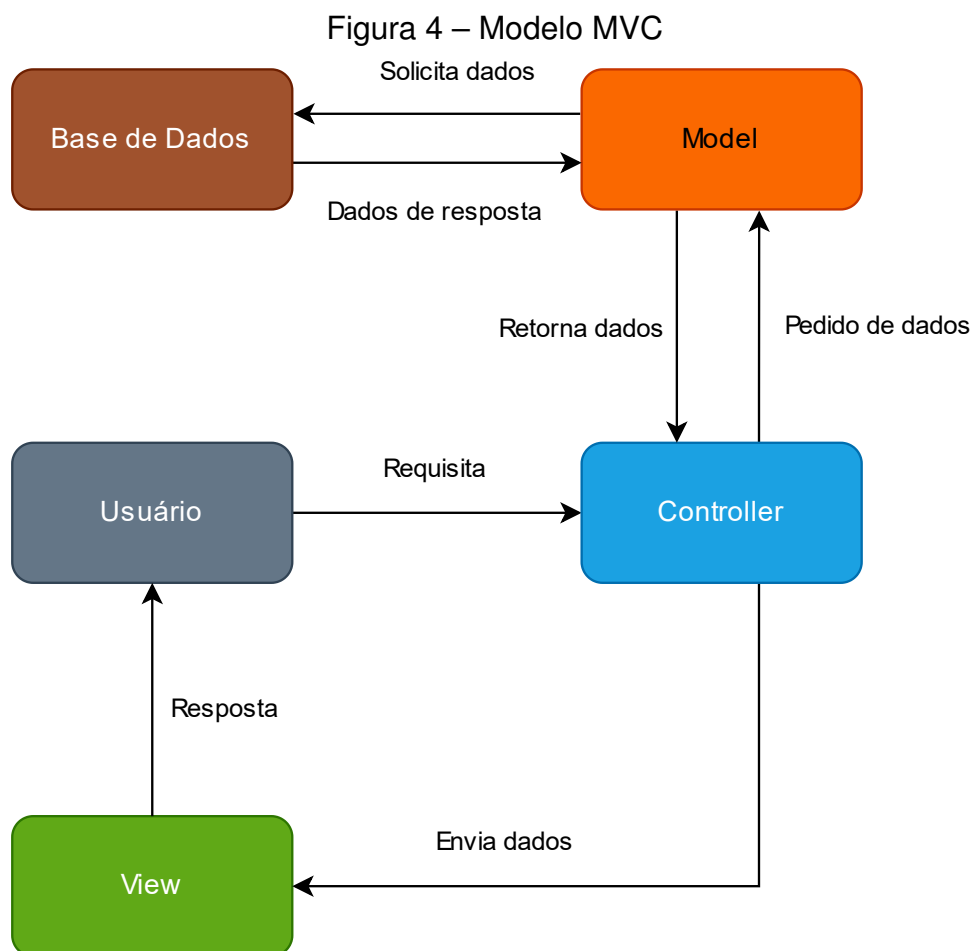
3.4.2 Modelagem do sistema

Para a organização do código da aplicação móvel, foi adotada a biblioteca Flutter Modular, que tem como objetivo integrar ao Flutter uma abordagem de injeção de dependência modularizada, gerenciamento de rotas e estruturação por escopos. Essa arquitetura permite que cada funcionalidade da aplicação possua sua própria implementação do padrão *Model*, *View* e *Controller* (MVC), ou, modelo, visão e controle. Assim, favorece a escalabilidade do projeto e facilita sua manutenção (HUSSAIN *et al.*, 2020; MODULAR, 2025; HAMDALAH *et al.*, 2025).

A Figura 4 demonstra a arquitetura. A camada *Model* se comunica diretamente com o Firebase, sendo os serviços do *Cloud Firestore* e *Firebase Authentication*, realizando, assim, operações de leitura, escrita e atualização dos dados armazenados de forma reativa.

A camada *View* corresponde à apresentação da interface gráfica, ou seja, aquilo que o usuário visualiza e interage no aplicativo. Esta camada é construída com *widgets* do Flutter e exibe os dados fornecidos pelo *Model*, reagindo a alterações de estado.

Por fim, o *Controller* atua como intermediador entre o *Model* e *View*, sendo responsável por processar as ações do usuário, atualizar os dados do *Model* e redirecionar os resultados de volta à *View*.



Fonte: Adaptada de Hussain *et al.* (2020).

3.4.3 Etapas de desenvolvimento

O processo de desenvolvimento foi dividido em três etapas principais: levantamento de requisitos, prototipagem das interfaces e implementação do sistema.

Inicialmente, foi feito o levantamento de requisitos, ao serem identificadas as principais demandas do público-alvo, por meio da revisão bibliográfica efetuada, análise de plataformas existentes e observações do contexto local em cidades pequenas.

Com base nos requisitos levantados, foi elaborado o projeto de interfaces contendo as telas da solução, a partir do Figma. A interface gerada guiou a produção do aplicativo, empregando-se a metodologia ágil. Essa etapa possibilitou ajustes antes do início da implementação.

Foi realizada a implementação do sistema utilizando-se o *framework* Flutter, com a linguagem Dart. Para o banco de dados, efetuou-se a integração do Firebase com o Flutter.

4 DESENVOLVIMENTO

Este capítulo descreve como foi o desenvolvimento e resultados do presente trabalho. A Seção 4.1 aborda a etapa de prototipação. A Seção 4.2 explica a modelagem do banco de dados. A Seção 4.3 mostra as etapas de implementação da solução. A Seção 4.4 discute os detalhes de implementação e telas do aplicativo. A Seção 4.5 apresenta o processo de validação da aplicação. Por fim, a Seção 4.6 descreve os aspectos relacionados à implementação para negócio, incluindo custos e estratégias de monetização.

4.1 Prototipagem

A etapa de projeto teve como foco inicial a definição da arquitetura e a organização das interfaces gráficas. O Apêndice A mostra a prototipagem que foi desenvolvida no Figma, seguindo-se os princípios de *design* centrado no usuário (MALLADI; KUMAR, 2025). As principais telas projetadas incluem:

- tela com inicial com imóveis em destaque;
- tela de login e cadastro de usuários;
- tela de busca de imóveis com filtros por localização, preço e tipo de imóvel;
- visualização detalhada do imóvel com fotos, descrição, valor, localização no mapa e dados de contato;
- tela para usuários que desejam compartilhar o imóvel, com busca por quartos separados e possibilidade de encontrar outros usuários.

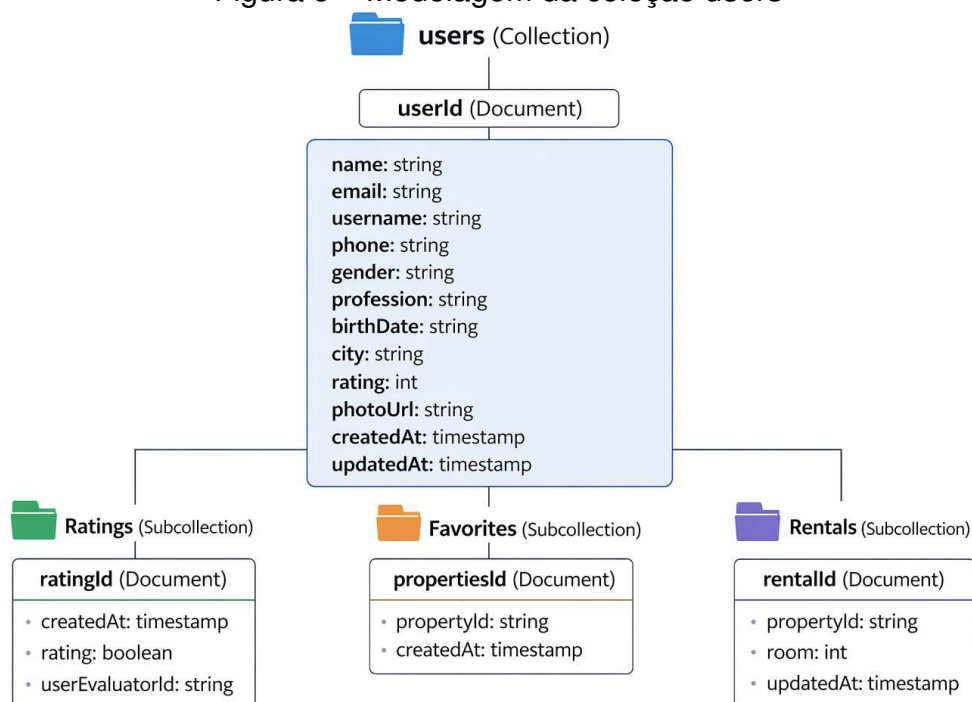
4.2 Modelagem do banco de dados

A Figura 5 apresenta uma parte do esquema lógico do banco de dados desenvolvido para o aplicativo, que foi implementado utilizando-se o Firebase, por meio do *Cloud Firestore*. Diferentemente de abordagens tradicionais baseadas em bancos relacionais, a modelagem adotada segue o paradigma *NoSQL*, no qual os dados são organizados em coleções, documentos e subcoleções.

O banco de dados foi estruturado para atender às funcionalidades principais da aplicação, incluindo cadastro de usuários, gerenciamento de imóveis, controle de locações e sistema de notificações. Conforme ilustrado na Figura 5, a coleção *users* é responsável pelo armazenamento dos dados dos usuários da aplicação. Cada documento, identificado por *userId*, representa um usuário único e contém informações

cadastrais, de perfil e de controle temporal.

Figura 5 – Modelagem da coleção *users*



Fonte: Elaborado pelo autor, 2026.

Os atributos como e-mail e nome de usuário permitem a identificação e autenticação do usuário, enquanto campos como profissão, telefone e cidade contribuem para a personalização da experiência dentro da plataforma. O campo *photoUrl* armazena a referência da imagem de perfil. Além disso, os campos *createdAt* e *updatedAt* permitem o controle e o rastreamento de alterações ao longo do tempo.

A subcoleção *ratings* armazena as avaliações recebidas pelo usuário. Cada documento representa uma avaliação individual contendo o valor, a data de criação e o identificador do avaliador. Essa abordagem permite registrar múltiplas avaliações sem impactar o tamanho do documento principal. Além disso, o campo *rating*, presente no documento principal do usuário, representa um valor agregado, evitando a necessidade de cálculos em tempo de consulta.

A subcoleção *favorites* armazena os imóveis favoritados pelo usuário. Cada documento contém o identificador do imóvel e a data em que foi adicionado aos favoritos. A utilização de uma subcoleção, em vez de uma lista no documento principal, foi adotada para garantir maior escalabilidade, uma vez que o número de favoritos pode crescer indefinidamente, sem comprometer o limite de tamanho do documento.

A subcoleção *rentals* registra os imóveis alugados pelo usuário. Cada documento contém informações como o identificador do imóvel, a posição do quarto na lista e a data de atualização.

Essa estrutura permite manter um histórico de locações, possibilitando a

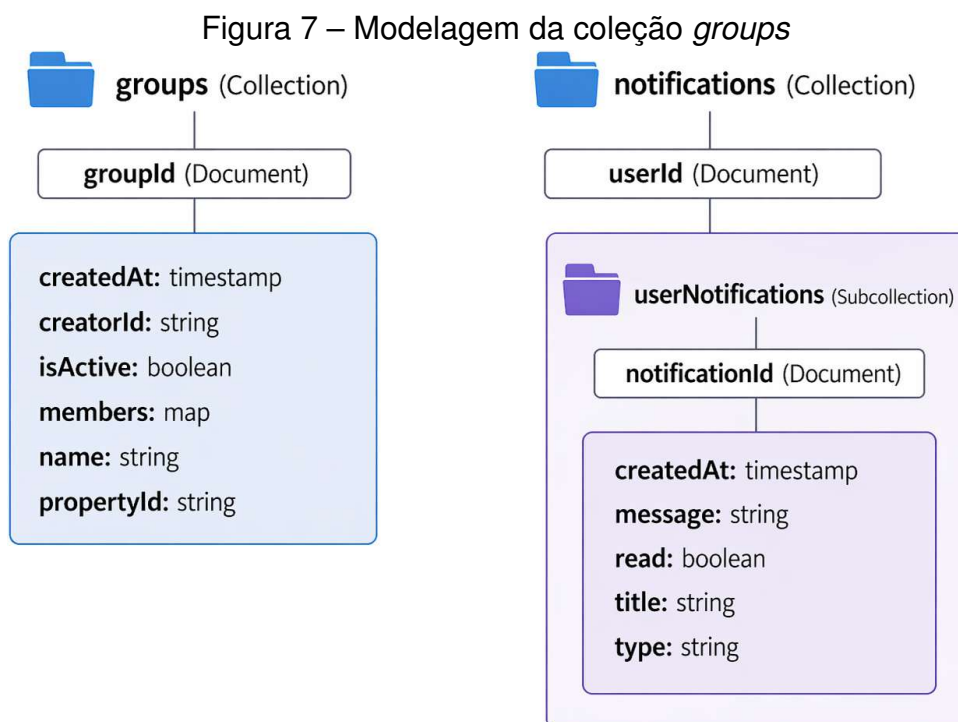
Figura 6 – Modelagem da coleção *properties*

Fonte: Elaborado pelo autor, 2026.

implementação de funcionalidades futuras, como a análise de comportamento dos usuários e o controle de transações realizadas. Para o desenvolvimento da aplicação, também foram estruturadas outras coleções. A coleção *properties*, ilustrada na Figura 6, é responsável por armazenar os imóveis disponíveis na plataforma. Além disso, as coleções *notifications* e *groups*, apresentadas na Figura 7, são responsáveis, respectivamente, pela gestão das notificações associadas aos usuários e dos grupos criados na aplicação.

4.3 Detalhes do processo de implementação

Nesta seção, são abordados os passos mais relevantes no processo de implementação. Os detalhes de como o ambiente de programação foi preparado encontram-se na Seção 4.3.1. O projeto foi organizado segundo o padrão da Se-



Fonte: Elaborado pelo autor, 2026.

ção 4.3.2. A Seção 4.3.3 mostra o uso de *widgets* para construção do aplicativo. A integração do banco de dados com o Flutter está descrita na Seção 4.3.4. O modo como foi feita a navegação entre as telas encontra-se na Seção 4.3.5. A Seção 4.3.6 apresenta o processo de autenticação de usuários. Por fim, a Seção 4.3.7 descreve a integração com o Google Maps.

4.3.1 Preparação do ambiente de programação

Inicialmente, foi baixada e realizada a instalação do *framework* Flutter por meio do site oficial, na versão 3.35.3. Após a instalação, procedeu-se com a configuração das variáveis de ambiente do sistema *Path*, permitindo o acesso global aos comandos do Flutter no sistema operacional.

Em seguida, foi utilizado o comando *flutter create* para a criação da estrutura inicial do projeto. Esse comando gera automaticamente a organização básica de diretórios, incluindo pastas como *lib*, na qual se encontra o código-fonte principal da aplicação, e *pubspec.yaml*, responsável pelo gerenciamento das dependências do projeto.

A inclusão de bibliotecas externas foi realizada por meio do arquivo *pubspec.yaml*, no qual são especificadas as dependências necessárias para o funcionamento da aplicação. Esse arquivo encontra-se no Apêndice B. Após a adição das bibliotecas, executou-se o comando *flutter pub get*, responsável por baixar e integrar automaticamente os pacotes ao projeto.

Como se trata de um aplicativo para o sistema Android, foi necessário baixar o *Android Studio*, que possui a ferramenta *Virtual Device Manager*, capaz de simular diferentes dispositivos móveis com variadas configurações de *hardware* e versões do sistema operacional Android. O telefone Pixel 8 foi o dispositivo escolhido para emulação, pelo fato de ter uma proporção de tela comumente usada no mercado.

Por fim, a integração entre o Flutter e o emulador foi realizada por meio do comando *flutter run*, que compila e executa a aplicação diretamente no dispositivo virtual selecionado. Essa etapa permite a visualização, em tempo real, das alterações realizadas no código.

4.3.2 Organização do projeto

A criação de um projeto em Flutter gera uma estrutura que segue um padrão organizado, contendo pastas essenciais como *lib*, onde fica o código principal, além de diretórios como *assets*, *android* e *ios*, responsáveis pelas configurações específicas de cada plataforma.

A Figura 8 apresenta a organização das pastas da aplicação. A pasta *models* reúne os arquivos responsáveis por definir a estrutura dos dados, determinando como as informações são organizadas e manipuladas no projeto. A pasta *controllers* contém os arquivos encarregados de realizar a comunicação com o banco de dados, além de controlar a lógica de negócio da aplicação.

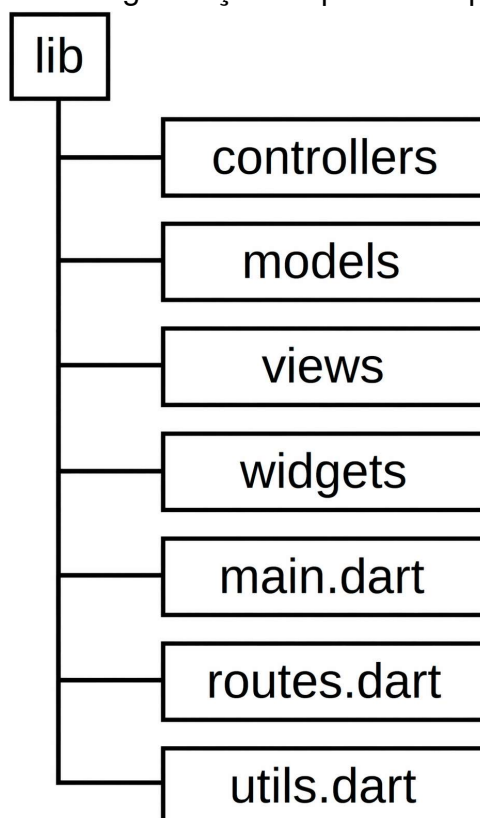
A pasta *views* engloba os arquivos responsáveis pela interface do usuário, exibindo as informações na tela e acionando outras camadas do sistema conforme necessário. O arquivo *main.dart* representa o ponto de entrada da aplicação, sendo responsável por inicializar o sistema; o *routes.dart* gerencia a navegação entre as telas; e o *utils.dart* centraliza definições reutilizáveis, como cores e estilos de texto utilizados ao longo das páginas.

4.3.3 Widgets

No *framework* Flutter, os *widgets* constituem os blocos fundamentais para a construção da interface gráfica de um aplicativo, sendo responsáveis por definir tanto a estrutura quanto o comportamento dos elementos visuais. Cada widget possui uma funcionalidade específica e pode conter um ou mais *widgets* filhos, que são organizados de forma hierárquica e aninhada para compor telas completas.

No Apêndice C, é possível ver um exemplo de utilização de *widgets*, sendo alguns deles: *SizedBox*, *Column*, *Row*, *Text* e *Padding*. O *SizedBox* é utilizado para definição de dimensões. Já os *widgets* *Column* e *Row* permitem a organização de múltiplos elementos nos eixos vertical e horizontal, respectivamente, ao passo que

Figura 8 – Organização de pastas do projeto



Fonte: Elaborado pelo autor, 2026.

Text exibe informações textuais, e *Padding* controla o espaçamento interno.

Além disso, cada *widget* possui propriedades configuráveis, como o *crossAxisAlignment* no *Column*, que permite definir o espaçamento entre os elementos filhos. Em aplicações, é comum a criação de *widgets* personalizados, reutilizáveis em diferentes telas, contribuindo para a modularização e manutenibilidade do código. Além disto, é possível a utilização de controladores em *widgets* de entrada, como o *TextField*, que permite o monitoramento, em tempo real, das informações inseridas pelo usuário.

Outro *widget* importante é o *ListView*, que possibilita a renderização de listas de elementos com suporte à rolagem, sendo, por exemplo, usado para exibir as listas de imóveis no aplicativo. Por fim, *widgets* como o *GestureDetector* possibilitam a captura de interações, como toques na tela, acionando funções que podem, por exemplo, atualizar dados em serviços externos, como o *Cloud Firestore*, e realizar a navegação entre telas.

4.3.4 Conexão com banco de dados

Para integrar o *Cloud Firestore* ao projeto Flutter, é necessário inicialmente criar um projeto no console do Firebase. Durante esse processo, é gerado automatica-

mente um arquivo de configuração denominado *google-services.json*, o qual contém todas as informações necessárias para autenticação e comunicação entre a aplicação e os serviços do Firebase.

Após a correta configuração do arquivo, torna-se necessário inicializar o Firebase no ponto de entrada da aplicação, localizado no arquivo *main.dart*. A Figura 9 mostra a chamada responsável pela inicialização, localizada nas linhas 2 e 3 do código.

Após a inicialização do *Cloud Firestore*, é possível instanciar o contexto do banco de dados e utilizar os métodos disponibilizados pelo *Firestore* apenas importando o pacote desejado.

Figura 9 – Inicialização Firebase

```
1 void main() async {  
2   WidgetsFlutterBinding.ensureInitialized();  
3   await Firebase.initializeApp(options:  
4     ↪ DefaultFirebaseOptions.currentPlatform);  
5   runApp(const MyApp());  
6 }
```

Fonte: Elaborado pelo autor, 2026.

4.3.5 Gerenciador de rotas

O principal recurso utilizado para a navegação é o objeto *Navigator*, que possibilita a transição entre telas por meio de métodos como *push* e *pop*. O método *push* é responsável por inserir uma nova rota na pilha, enquanto o método *pop* remove a rota atual, retornando à tela anterior.

O código na Figura 10 é responsável por interceptar as requisições de navegação e gerar dinamicamente as rotas da aplicação. Nesse trecho, o parâmetro *settings* contém informações sobre a rota solicitada, como seu nome e possíveis argumentos enviados durante a navegação.

Desta forma, o código realiza uma busca em uma estrutura de dados denominada *routes*, presente no Apêndice D, que consiste em um mapeamento entre os nomes das rotas e suas respectivas funções construtoras.

Caso a rota solicitada seja encontrada, o método retorna um objeto do tipo *MaterialPageRoute*, responsável por construir a nova tela. A função *builder* é, então, executada, recebendo o contexto da aplicação e os argumentos associados à rota, permitindo a passagem de dados entre telas. Por outro lado, caso a rota não seja encontrada no mapeamento, o método retorna uma rota padrão que exibe uma mensagem de erro ao usuário.

Figura 10 – Gerenciador de rotas para navegação entre telas

```

1 static Route<dynamic>? onGenerateRoute(RouteSettings settings) {
2     final builder = routes[settings.name];
3
4     if (builder != null) {
5         return MaterialPageRoute(
6             builder: (context) => builder(context, settings.arguments),
7         );
8     }
9
10    return MaterialPageRoute(
11        builder: (_) =>
12            const Scaffold(body: Center(child: Text('Rota não encontrada'))),
13    );
14 }

```

Fonte: Elaborado pelo autor, 2026.

4.3.6 Função de autenticação

O Firebase *Authentication* foi utilizado como solução de autenticação da aplicação, permitindo o gerenciamento de usuários por meio de diferentes métodos de login. Sua integração com o Flutter foi realizada através das bibliotecas oficiais. A lógica de autenticação utiliza o *FirebaseAuth* para login; o *Cloud Firestore*, para armazenamento de dados dos usuários; e o *Firebase Storage*, para imagens de perfil.

A configuração do serviço envolveu a criação de um projeto no Firebase Console, a vinculação da aplicação via arquivo de configuração e a ativação dos provedores de autenticação por e-mail/senha e Google. Para autenticação Google, foi utilizada a biblioteca Google *Sign-In*, responsável pela obtenção dos *tokens necessários* ao processo de login, disponível no Apêndice E.

A Figura 11 demonstra o método *loginWithEmailAndPassword*, que efetua a autenticação tradicional utilizando credenciais de e-mail e senha fornecidas pelo usuário. Esse processo ocorre por meio da função *signInWithEmailAndPassword*, que valida os dados junto ao *Firebase Authentication*.

Figura 11 – Código para login com e-mail e senha

```

1 Future<UserCredential> loginWithEmailAndPassword(
2     String email,
3     String password,
4 ) async {
5     return await _auth.signInWithEmailAndPassword(
6         email: email,
7         password: password,
8     );
9 }

```

Fonte: Elaborado pelo autor, 2026.

O método *loginWithGoogle*, presente no Apêndice E, implementa a autenticação utilizando contas Google. Inicialmente, o usuário é direcionado ao fluxo de login do Google por meio da biblioteca *Google Sign-In*. Após a autenticação, são obtidos os tokens de acesso *accessToken* e *idToken*, que são utilizados para gerar uma credencial compatível com o Firebase. Essa credencial é, então, enviada ao *Firebase Authentication* por meio do método *signInWithCredential*, permitindo a autenticação do usuário na aplicação.

O método presente na Figura 12 é responsável pelo cadastro de novos usuários. Inicialmente, é criada uma conta no *Firebase Authentication* utilizando e-mail e senha. Em seguida, os dados complementares do usuário (nome, telefone, cidade) são armazenados no *Cloud Firestore*. Caso o usuário forneça uma imagem de perfil, esta é armazenada no *Firebase Storage*.

Figura 12 – Código para criação de usuário com e-mail e senha

```

1 User? user = _auth.currentUser;
2   if (user == null && password.isNotEmpty) {
3     UserCredential userCredential = await
4       ↪ _auth.createUserWithEmailAndPassword(
5         email: email,
6         password: password,
7       );
8     user = userCredential.user;
  }

```

Fonte: Elaborado pelo autor, 2026.

A funcionalidade de login utilizando o Facebook seria implementada, conforme previsto no protótipo. No entanto, devido a erros e à complexidade envolvida no processo de configuração, optou-se por adiar sua implementação, a fim de viabilizar a entrega do presente trabalho.

4.3.7 Biblioteca Google Maps

Para a implementação da funcionalidade de geolocalização dos imóveis, foi utilizada a biblioteca *google_maps_flutter*, um *plugin* oficial que permite a integração do serviço Google Maps em aplicações desenvolvidas com o *framework* Flutter. Essa biblioteca possibilita a exibição de mapas interativos diretamente na interface do aplicativo, oferecendo recursos como posicionamento geográfico, marcadores e controle de câmera.

No sistema desenvolvido, a biblioteca foi empregada para apresentar a localização exata dos imóveis cadastrados, utilizando coordenadas geográficas (latitude e longitude) armazenadas diretamente no banco de dados. A partir dessas informações, é criado um objeto do tipo *LatLng*, responsável por representar a posição do

imóvel no mapa. O código está disponível no Apêndice F.

Figura 13 – Implementação do *widget* Google Maps

```

1 GoogleMap(
2   initialCameraPosition: CameraPosition(
3     target: _propertyPosition!,
4     zoom: 15,
5   ),
6   markers: {
7     Marker(
8       markerId: const MarkerId("property"),
9       position: _propertyPosition!,
10      infoWindow: InfoWindow(
11        title: title,
12        snippet: location,
13      ),
14    ),
15  },
16  zoomControlsEnabled: false,
17 ),

```

Fonte: Elaborado pelo autor, 2026.

O trecho de código que gera a visualização do mapa é mostrado na Figura 13, por meio do *widget* *GoogleMap*, no qual são definidos parâmetros como a posição inicial da câmera, com *CameraPosition*, e também o nível de *zoom*, proporcionando ao usuário uma visualização adequada da região. Além disso, foi utilizado o recurso de marcadores *Marker*, permitindo destacar visualmente a localização do imóvel, acompanhado de uma janela informativa através do uso do *InfoWindow*, contendo dados como título e endereço.

Outro aspecto é o controle da exibição do mapa, que ocorre apenas quando as coordenadas estão disponíveis. Caso contrário, o sistema apresenta uma mensagem informando a indisponibilidade da localização.

4.4 Desenvolvimento do aplicativo

Nesta seção, são apresentadas as interfaces desenvolvidas, bem como o detalhamento do funcionamento geral do aplicativo. Além disso, a Seção 4.4.1 descreve o fluxo de navegação entre as telas. Inicialmente, na Seção 4.4.2, são descritas as funcionalidades relacionadas à visualização dos imóveis disponíveis para o usuário. Em seguida, aborda-se, na Seção 4.4.3, a tela de detalhes dos imóveis. Posteriormente, é apresentada a funcionalidade de favoritar imóveis, na Seção 4.4.4. Na sequência, a Seção 4.4.5 discute sobre os processos de login e cadastro, e a Seção 4.4.6 contempla a descrição da tela de perfil do usuário. Além disso, a Seção 4.4.7 detalha a funcionalidade de compartilhamento de moradia. Por fim, na Seção 4.4.8,

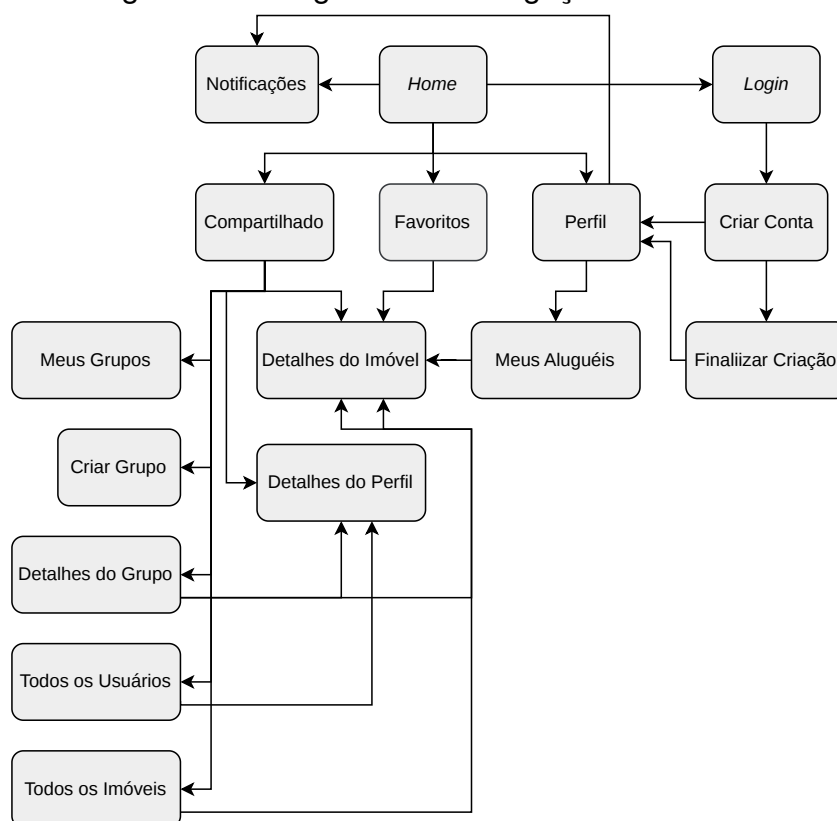
apresentam-se as notificações do sistema.

4.4.1 Navegação entre telas

Conforme apresentado no diagrama de navegação entre telas, ilustrado na Figura 14, o ponto inicial da aplicação é a tela principal (*Home*). A partir dessa interface, é possível acessar os diferentes módulos do sistema por meio do menu inferior.

A aplicação foi desenvolvida de forma a permitir o acesso sem a necessidade de autenticação, possibilitando a visualização da tela *Home* e dos detalhes dos imóveis. No entanto, para acessar funcionalidades que envolvem interação com o usuário, é obrigatório realizar o login. Dessa forma, ao tentar acessar essas funcionalidades, o usuário é automaticamente redirecionado para a tela de autenticação.

Figura 14 – Diagrama da navegação entre telas



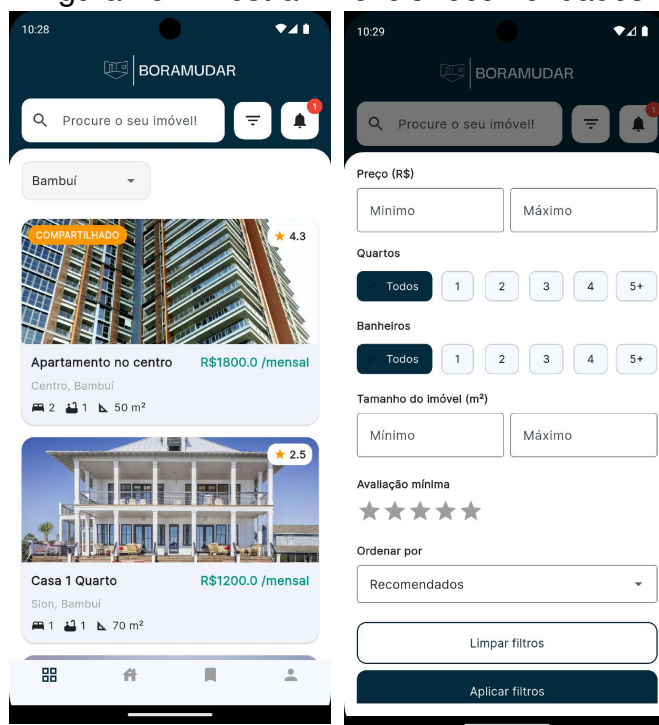
Fonte: Elaborado pelo autor, 2026.

4.4.2 Mostrar imóveis

A Figura 15(a) demonstra a tela inicial do aplicativo desenvolvida. Como elemento principal, o sistema entrega uma recomendação automática de imóvel, baseado na nota que o imóvel possui e também na quantidade de avaliações feitas por

usuários. A parte superior da página contém uma caixa de texto que possibilita pesquisar pelo nome do anúncio do imóvel. Ao lado, há, também, um botão que redireciona o usuário para uma página contendo todas as suas notificações, além de exibir, com um ícone vermelho, se existe alguma não lida. Possui, ainda, uma barra de navegação na parte inferior, que leva para as seguintes telas: início, favoritos, moradia conjunta e perfil.

Figura 15 – Mostrar imóveis recomendados



(a) – Página inicial

(b) – Filtros dos imóveis

Fonte: Elaborado pelo autor, 2026.

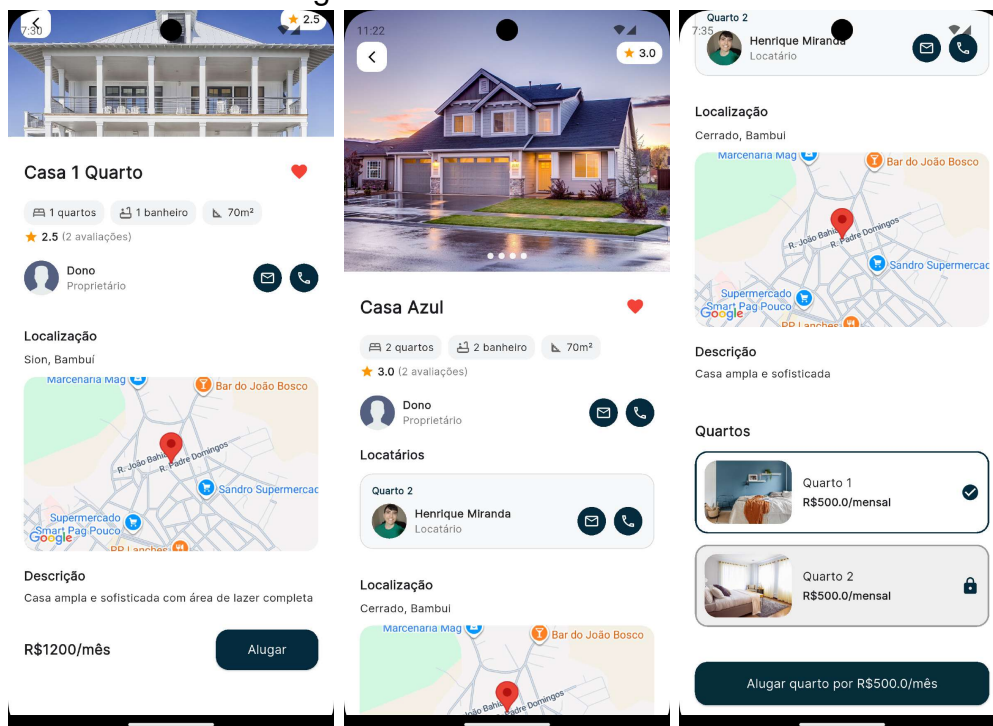
Existem dois tipos de imóveis na aplicação. O primeiro corresponde ao modelo mais comum, semelhante ao encontrado em outros aplicativos do mesmo segmento, não apresentando funcionalidades diferenciadas relevantes. O segundo tipo, por sua vez, possui funcionalidades que permitem o compartilhamento do imóvel entre usuários, apresentando, portanto, comportamentos distintos. Dessa forma, esse tipo é identificado por meio de uma *tag* específica, com o objetivo de melhorar a usabilidade da aplicação.

A página inicial também possui um botão com funcionalidade de aplicar filtros para os imóveis, visando personalizar resultados de busca. Ao pressionar o botão, é mostrado para o usuário o campo presente na Figura 15(b), contendo ajustes de preferência para encontrar imóveis, sendo possível definir o preço desejado, a quantidade de quartos e banheiros, o tamanho do imóvel e também a avaliação mínima.

4.4.3 Detalhes dos imóveis

Na tela inicial, ao clicar em um imóvel, o aplicativo redireciona o usuário para a tela que contém todos os dados do imóvel desejado, conforme exposto na Figura 16(a). Nesta tela, é possível ver uma descrição do imóvel, bem como sua localização, com representação no Google Maps.

Figura 16 – Detalhes do imóvel



(a) – Detalhes do imóvel (b) – Imóvel compartilhado (c) – Alugar quarto

Fonte: Elaborado pelo autor, 2026.

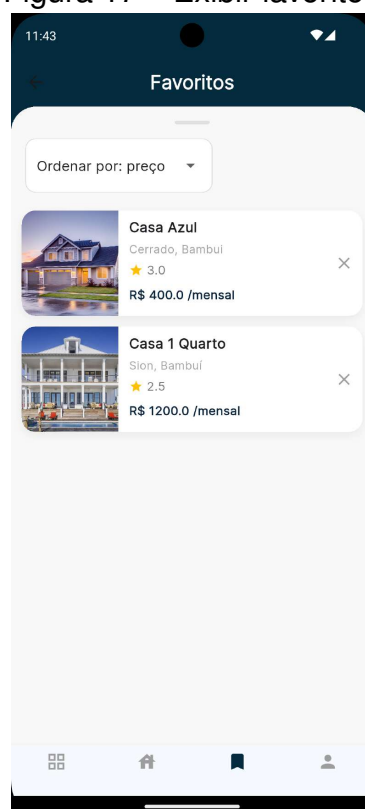
Conforme mencionado anteriormente, existem dois tipos de imóveis na aplicação. Nesse contexto, a Figura 16(b) apresenta um imóvel configurado para compartilhamento. A principal diferença desse tipo consiste na possibilidade de visualizar os locatários associados, caso existam. Além disso, é exibida uma área contendo os quartos individualizados, conforme ilustrado na Figura 16(c), na qual são apresentadas imagens e valores específicos para cada unidade. Ressalta-se que esses quartos podem ser bloqueados quando já estiverem alugados.

Consegue-se entrar em contato com o dono da propriedade ou locatários, caso haja o desejo de alugar o imóvel, por meio de botões que redirecionam para o Whatsapp ou aplicativo de e-mail previamente instalado. A conclusão da etapa de locação, entre o proprietário e o usuário, deve ser feita utilizando-se os meios mencionados anteriormente. Sendo assim, a plataforma serve apenas para encurtar a relação entre os meios, sendo efetuada uma locação apenas representativa.

4.4.4 Favoritar imóvel

Por meio da barra de navegação, é possível acessar a área de favoritos. Favoritar tem a funcionalidade de armazenar aquele anúncio do imóvel para ser rapidamente acessado. Sempre que se abrir os detalhes de um imóvel, é possível favoritá-lo clicando no ícone de coração, no canto superior direito. A Figura 17 demonstra esta funcionalidade.

Figura 17 – Exibir favoritos



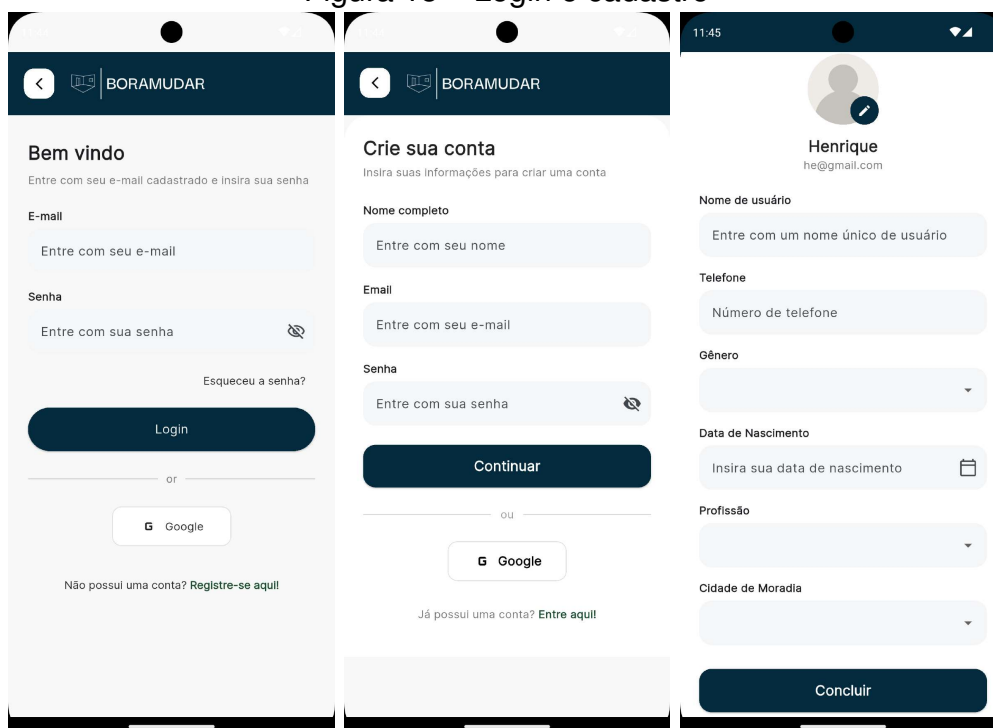
Fonte: Elaborado pelo autor, 2026.

4.4.5 Login e cadastro

A Figura 18(a), inicialmente, mostra componentes nos quais é possível o usuário fazer login utilizando e-mail e senha, ou usar uma conta cadastrada Google. Para criar uma conta, deve-se clicar no texto “Registre-se aqui!”.

A Figura 18(b) possui campos necessários para o usuário preencher e cadastrar uma nova conta. A Figura 18(c) contém os campos restantes para realizar esta ação. Para o cadastro, todas as informações são obrigatórias, e, além disso, os campos nome de usuário e e-mail precisam ser únicos. Também é possível realizar o cadastro por meio do serviço de conta Google, sendo necessário ter uma previamente.

Figura 18 – Login e cadastro



(a) – Tela de login (b) – Tela de cadastro (c) – Concluir cadastro

Fonte: Elaborado pelo autor, 2026.

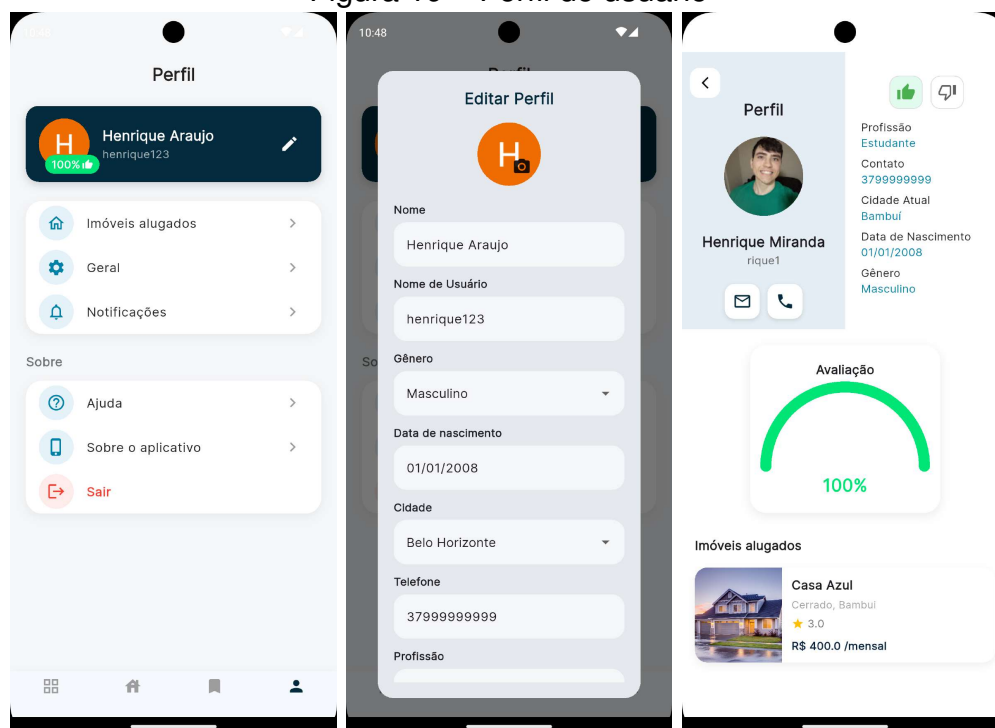
4.4.6 Perfil do usuário

Na Figura 19(a), é possível visualizar o perfil, com as informações do usuário. Por meio dos botões, é possível ver todos os imóveis que o usuário está alugando, alterar as configurações de privacidade e personalização do aplicativo, visualizar as notificações e também sair da conta atual. Por fim, é possível editar as informações inseridas na criação da conta, como exibido na Figura 19(b).

Ao acessar o perfil de outro usuário, este é apresentado conforme ilustrado na Figura 19(c). Nessa tela, é possível visualizar informações detalhadas do usuário, incluindo dados relevantes para a negociação, bem como acessar os meios de comunicação disponibilizados para contato. Além disso, também é possível consultar os imóveis vinculados ao usuário, facilitando a navegação entre seu perfil e os anúncios de locação publicados.

A plataforma também incorpora um mecanismo de avaliação que permite aos usuários registrar avaliações positivas ou negativas com base em suas experiências de interação. Essas avaliações contribuem para a formação de uma reputação digital, auxiliando na identificação de usuários mais confiáveis e promovendo maior transparência e segurança nas negociações. Adicionalmente, é possível consultar a avaliação geral atribuída ao usuário, permitindo uma análise rápida de sua credibilidade perante a comunidade da plataforma.

Figura 19 – Perfil do usuário



(a) – Visualizar perfil (b) – Editar perfil (c) – Detalhes do perfil

Fonte: Elaborado pelo autor, 2026.

4.4.7 Compartilhamento de moradia

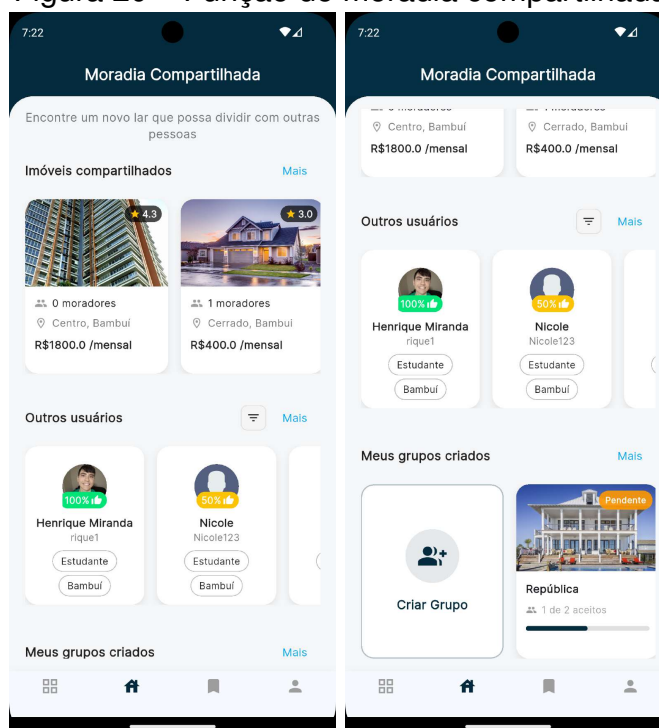
A Figura 20(a) apresenta o funcionamento da área que permite aos usuários compartilharem uma mesma moradia. Nessa seção, os imóveis são exibidos com base em sua avaliação, considerando aqueles que possibilitam a locação de quartos individuais, sendo necessário que ao menos um quarto esteja disponível para que o imóvel seja listado. Além disso, o botão “mais” redireciona o usuário para uma página dedicada na qual é possível visualizar uma quantidade maior de imóveis.

A Figura 20(b) apresenta outros perfis de usuários do aplicativo, permitindo o estabelecimento de contato com pessoas interessadas em compartilhar uma moradia. Há, também, a presença de um botão de filtragem que possibilita refinar a busca com base em *tags*, auxiliando na identificação de perfis compatíveis. Por fim, o botão “mais” permite a exibição de uma quantidade ampliada de usuários.

Ainda na Figura 20(b), observa-se a área responsável pela criação e visualização de grupos, os quais têm como finalidade permitir que usuários se organizem para solicitar a um proprietário a utilização de um imóvel de forma compartilhada, possibilitando sua divisão em quartos individuais.

A Figura 21(a) apresenta a tela exibida ao ser acionado o botão de criação de grupos, sendo que, para a criação de um grupo, é necessário definir um nome para identificação, selecionar um imóvel e escolher pelo menos outro usuário. Caso seja

Figura 20 – Função de moradia compartilhada

(a) – Imóvel comparti- (b) – Usuários e grupos
lhado

Fonte: Elaborado pelo autor, 2026.

selecionada uma quantidade de usuários superior ao número de quartos disponíveis no imóvel, é exibida uma mensagem de alerta.

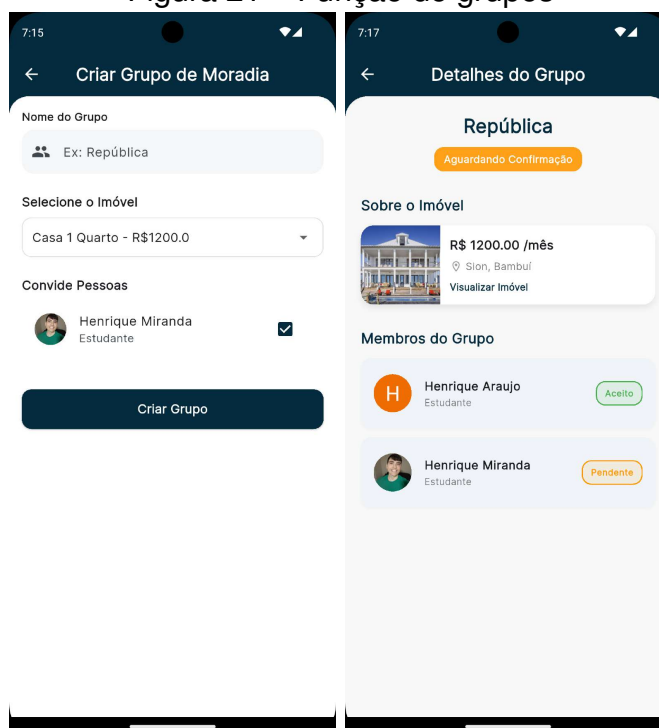
Ressalta-se que os imóveis listados nessa tela são apenas aqueles previamente favoritados, assim como os usuários exibidos são aqueles que receberam avaliações positivas. Tal abordagem foi adotada com o objetivo de reduzir o tempo de resposta em consultas ao banco de dados. Após a criação do grupo, todos os envolvidos, usuários e proprietário do imóvel devem aprovar a solicitação por meio de notificações recebidas. Somente após essa aprovação, o imóvel é efetivamente alugado e passa a ser classificado como do tipo compartilhado.

A Figura 21(b) ilustra a tela acessada ao se selecionar um grupo específico, permitindo a visualização detalhada de suas informações. Nessa interface, são apresentados o imóvel associado e os membros que compõem o grupo. Além disso, é possível acompanhar o andamento das solicitações por meio de *tags* indicativas de status.

4.4.8 Notificações

A Figura 22 ilustra o funcionamento do sistema de notificações do usuário, permitindo mantê-lo informado sobre diversas ações realizadas ao longo da utiliza-

Figura 21 – Função de grupos



(a) – Criar grupos

(b) – Visualizar grupo

Fonte: Elaborado pelo autor, 2026.

ção do aplicativo. As notificações ocorrem apenas dentro da aplicação e apresentam diferenças de exibição conforme seu estado de visualização, ou seja, se já foram visualizadas ou não. Além disso, é possível excluir qualquer notificação por meio de um gesto de deslizamento para a esquerda sobre o item desejado.

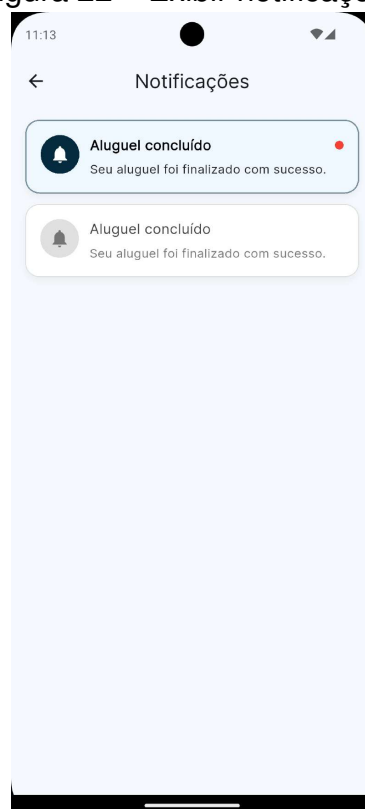
4.5 Validação

As validações de funcionamento do aplicativo foram realizadas em dispositivos Android, tanto em ambiente virtual quanto físico. Para o ambiente virtual, utilizou-se um emulador do modelo Pixel 8, com Android versão 11. Já no ambiente físico, as validações foram conduzidas em um dispositivo pessoal modelo POCO X7, com Android versão 16.

Durante a execução das validações, observou-se que a maioria dos erros identificados estava relacionada à conexão com o banco de dados, aos problemas de configuração no uso de bibliotecas externas e a inconsistências em elementos visuais que não apresentavam o comportamento esperado.

Adicionalmente, efetuaram-se validações de usabilidade com um grupo reduzido de usuários, com o intuito de identificar possíveis falhas na interação com o sistema. Foram encontrados e corrigidos problemas estruturais na arquitetura da aplicação, bem como inconsistências no layout, organização e fluxo de navegação. Essas

Figura 22 – Exibir notificações



Fonte: Elaborado pelo autor, 2026.

validações tiveram como objetivo avaliar a facilidade de uso do aplicativo, especialmente no que se refere à execução de tarefas essenciais e à localização de informações.

4.6 Implementação para negócio

Nesta seção, são abordados os aspectos relacionados à viabilidade econômica da aplicação, considerando tanto os custos operacionais quanto as possíveis estratégias de monetização.

No estágio inicial do projeto, é possível utilizar o plano gratuito oferecido pela plataforma, o qual atende a aplicações com baixa demanda de usuários e volume reduzido de dados. Para um estágio mais avançado, estimou-se uma base de aproximadamente 1000 usuários ativos mensais. Dessa forma, projetou-se um custo de até US\$ 20 por mês, com base nos valores praticados pela plataforma Firebase, variando principalmente em função da quantidade de leituras e escritas realizadas no banco de dados, bem como do volume de dados armazenados.

Os principais fatores que influenciam esses custos incluem o número de operações no banco de dados, o armazenamento de imagens dos imóveis e o uso de autenticação. Além disso, a integração com serviços externos, como o Google

Maps, pode gerar custos adicionais, uma vez que sua utilização também é baseada no volume de requisições efetuadas.

No que se refere à monetização, diferentes estratégias podem ser adotadas para garantir a sustentabilidade financeira da aplicação. Uma das abordagens consiste na cobrança de comissão sobre contratos de locação efetivados por intermédio da plataforma.

Para a implementação desse modelo, seria necessária a formalização de mecanismos operacionais e jurídicos que permitam identificar e validar as negociações concluídas por meio do sistema, bem como definir, em termos de uso e contratos específicos, as responsabilidades e obrigações das partes envolvidas. Inicialmente, prevê-se a cobrança de uma comissão equivalente a 1% do valor do aluguel, percentual que poderá ser ampliado para até 5% à medida que a plataforma alcance uma base consolidada de usuários e demonstre maior capacidade de geração de valor para locadores e locatários.

Outra estratégia consiste na oferta de planos premium para anunciantes, permitindo que proprietários de imóveis obtenham maior visibilidade para seus anúncios por meio de recursos como destaque em listagens, prioridade nos resultados de busca e divulgação ampliada. Adicionalmente, a inserção de publicidade no aplicativo representa uma alternativa viável de geração de receita, podendo ser implementada por meio de plataformas como o Google AdMob¹, que possibilitam a exibição de anúncios segmentados conforme o perfil e os interesses dos usuários.

Dessa forma, observa-se que a aplicação apresenta potencial de sustentabilidade econômica, uma vez que os custos operacionais estimados são relativamente baixos quando comparados às possibilidades de monetização disponíveis, especialmente após a consolidação da base de usuários e a formalização dos mecanismos necessários para a intermediação das locações.

¹ <https://admob.google.com/>

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo principal contribuir para a digitalização do mercado imobiliário voltado à locação de imóveis em cidades de menor porte, com foco no atendimento às necessidades de estudantes em mobilidade acadêmica. Para isso, foi desenvolvido um aplicativo móvel multiplataforma capaz de facilitar o processo de busca, visualização e contato entre locadores e locatários, promovendo maior acessibilidade, praticidade e eficiência durante as negociações de aluguel.

O desenvolvimento da solução foi conduzido por meio de três etapas principais: levantamento de requisitos, prototipação das interfaces e implementação do sistema. Para sua execução, foram empregadas tecnologias amplamente utilizadas no desenvolvimento de aplicações modernas, como o framework Flutter, a linguagem Dart e o Firebase, como banco de dados NoSQL. Além disso, adotou-se a metodologia ágil Kanban para organização das atividades, e o padrão arquitetural MVC para estruturar o sistema de forma modular e escalável.

Os resultados obtidos demonstram que o aplicativo desenvolvido atende satisfatoriamente aos requisitos funcionais definidos durante a etapa de levantamento de requisitos. As funcionalidades implementadas contemplam recursos essenciais para o processo de locação, incluindo visualização de imóveis em destaque, sistema de busca com filtros, exibição detalhada dos anúncios com múltiplas imagens, integração com mapas para localização geográfica e acesso direto às informações de contato dos locadores. Dessa forma, foi possível validar que a solução desenvolvida é capaz de atender aos objetivos inicialmente propostos pelo trabalho.

Entre os diferenciais implementados, destaca-se a funcionalidade de locação compartilhada, que permite aos estudantes encontrar quartos individuais e identificar usuários interessados em dividir moradia. Tal recurso atende a uma demanda específica do público-alvo e representa uma característica pouco explorada nas soluções analisadas durante o levantamento de requisitos. Além disso, a estruturação dos dados no Cloud Firestore possibilitou a organização adequada das informações relacionadas a usuários, imóveis, avaliações, favoritos, notificações e grupos, garantindo a persistência e o gerenciamento eficiente dos dados da aplicação.

A relevância deste trabalho pode ser observada sob diferentes perspectivas. Do ponto de vista social, a solução contribui para a modernização dos processos de locação em cidades onde a divulgação de imóveis ainda ocorre predominantemente por meios informais. Sob a perspectiva dos estudantes, o aplicativo oferece uma alternativa centralizada para a busca por moradia, reduzindo dificuldades comuns enfrentadas durante processos de mudança para localidades desconhecidas. Além disso, o trabalho demonstra a viabilidade da aplicação de tecnologias modernas de desenvolvimento móvel na resolução de problemas reais presentes em contextos

regionais.

Apesar dos resultados positivos alcançados, algumas limitações devem ser consideradas. A validação da aplicação foi realizada principalmente por meio de testes de mesa, não envolvendo uma avaliação sistemática com usuários reais. Dessa forma, aspectos relacionados à usabilidade, experiência do usuário e aceitação da plataforma ainda podem ser explorados em trabalhos futuros. Adicionalmente, a evolução da solução para um cenário de produção demandará a implementação de mecanismos complementares, como verificação de identidade dos usuários, integração com documentação jurídica, estratégias de monetização e aprimoramentos voltados à segurança e ao desempenho da aplicação.

Conclui-se, portanto, que os objetivos propostos foram alcançados, uma vez que foi desenvolvida uma aplicação funcional capaz de atender aos requisitos estabelecidos e às necessidades identificadas para estudantes em mobilidade acadêmica e proprietários de imóveis. As funcionalidades implementadas demonstraram a viabilidade técnica da proposta e evidenciaram o potencial da plataforma para contribuir com a digitalização do mercado imobiliário em cidades de menor porte. Assim, o sistema desenvolvido constitui uma base sólida para futuras evoluções, podendo ser expandido e aprimorado para atender a novos cenários e demandas do setor.

5.1 Trabalhos futuros

Como possíveis trabalhos futuros, será adicionada a possibilidade de realizar o cadastro com as contas do Facebook, além da criação de uma nova tela para usuários que esqueceram a senha. Destaca-se, também, a necessidade de adaptar e realizar testes com o sistema operacional iOS. A expansão para a plataforma iOS permitirá maior abrangência de público, promovendo maior escalabilidade e potencial de uso do sistema em diferentes contextos.

Outro ponto refere-se à ampliação das funcionalidades do aplicativo, especialmente no que diz respeito ao cadastro de imóveis. Pretende-se implementar um aplicativo separado que permita a criação de anúncios de imóveis diretamente pelos proprietários. Além disso, almeja-se viabilizar a efetivação do processo de locação entre usuários, incluindo etapas de aprovação para as solicitações de aluguel.

Pretende-se, ainda, desenvolver e implementar a documentação jurídica necessária para regulamentar o uso da plataforma, incluindo termos de uso, políticas de privacidade e demais instrumentos legais pertinentes. Essa documentação tem como objetivo estabelecer, de forma clara, os direitos, deveres e responsabilidades dos usuários, proporcionando maior segurança jurídica à plataforma e contribuindo para o cumprimento das obrigações assumidas por todas as partes envolvidas.

Sobre a interface e experiência do usuário, observa-se a necessidade de

implementação de funcionalidades ainda não desenvolvidas, como os botões “Geral”, “Ajuda” e “Sobre o aplicativo”. A inclusão dessas áreas visa melhorar a usabilidade, oferecendo suporte informativo ao usuário e maior transparência sobre o funcionamento do sistema. Ademais, pretende-se incorporar opções de personalização relacionadas à privacidade e segurança, permitindo que o usuário tenha controle sobre dados sensíveis que deseja disponibilizar na plataforma.

A partir dos testes de usabilidade realizados, foi possível identificar melhorias importantes na interface. Uma das principais contribuições foi a adição de títulos para identificação dos cômodos dos imóveis cadastrados, o que facilitou a compreensão das informações apresentadas e tornou a navegação mais intuitiva.

Por fim, ressalta-se a possibilidade de evolução do sistema para um modelo de implementação comercial. Nesse contexto, seria necessário incorporar funcionalidades adicionais, como mecanismos de verificação de usuários e políticas de uso, além de estratégias de monetização. Tal expansão exigiria, também, maior robustez em termos de segurança e desempenho.

REFERÊNCIAS

- ARAÚJO, N.; GORGÔNIO, F.; VALE, K. Combinando Metodologias Ágeis para Execução de Projetos de Software Acadêmicos. *In: ANAIS da I Escola Regional de Sistemas de Informação do Rio de Janeiro*. Porto Alegre: SBC, 2014. p. 103–106. Disponível em: <https://sol.sbc.org.br/index.php/ersi-rj/article/view/5792>. Acesso em: 07/05/2025.
- BARBOZA, L. F. *et al.* Análise comparativa entre as abordagens ágil e tradicional de gestão de projetos: Um estudo de caso no setor industrial. **Anais do V Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade**, São Paulo, 2016.
- BERNARDELLI, M. L. F. d. H. **Pequenas cidades na região de Catanduva–SP: papéis urbanos, reprodução social e produção de moradias**. 2004. Tese (Doutorado em Geografia) – Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente. Disponível em: <https://repositorio.unesp.br/server/api/core/bitstreams/2f37026b-24b0-4cb0-b176-43b8f3b0492c/content>. Acesso em: 04/06/2025.
- BISWAS, A. **RENTEZ: A Simplified Rental App Experience Designed to Promote Transparency and Verification**. 2024. Dissertação (Mestrado em Mídias Digitais) – Universidade Ryerson, Toronto. Disponível em: <https://doi.org/10.32920/26052562.v1>. Acesso em: 31/05/2025.
- BRASIL, A. **Aluguel subiu 13,5% em 2024: veja quais capitais têm preço mais alto**. 2024. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2025-01/aluguel-subiu-135-em-2024-veja-quais-capitais-tem-o-preco-mais-alto>. Acesso em: 08/06/2025.
- CARVALHO, L. V.; RUAS, J. C. X. Housing Prices in Brazil and Their Impacts on Interest Rate and Income in the Inflation Targeting Regime Between 2007 and 2017. **Revista De Estudos Sociais**, v. 23, n. 47, p. 51–74, 2017. Disponível em: <https://dialnet.unirioja.es/servlet/articulo?codigo=8286187>. Acesso em: 08/06/2025.
- CASTRO, L. M. S. de; LESO, B. H.; CORTIMIGLIA, M. N. Transformação Digital no Mercado de Locação de Imóveis. *In: X Congresso Internacional de Conhecimento e Inovação (ciKi)*. 2020. Disponível em: <https://api.semanticscholar.org/CorpusID:228887678>. Acesso em: 31/05/2025.
- CNN BRASIL. **Rentabilidade de imóveis no Brasil chega a 19% em 2024, aponta estudo**. 2024. Disponível em: <https://www.cnnbrasil.com.br/economia/macroeconomia/rentabilidade-de-imoveis-no-brasil-chega-a-19-em-2024-aponta-estudo/>. Acesso em: 08/06/2025.
- CODE, V. S. **Visual Studio Code documentation**. 2025. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 20/07/2025.
- CORDEIRO, G.; BAGGIO, C.; FRANÇA, L. A. **Comportamento do consumidor de imóveis em 2040**. 2019. Disponível em:

<https://www.abrainc.org.br/uploads/2019/09/Abrainc-Pesquisa-v10.pdf>. Acesso em: 07/05/2025.

CORRÊA, R. L. Globalização e reestruturação da rede urbana: uma nota sobre as pequenas cidades. **Território**, Rio de Janeiro, v. 4, n. 6, p. 43–53, 1999.

DHAMODARAN, S. *et al.* House Rental Application System. *In: INTERNATIONAL Conference on Recent Trends in Data Science and its Applications*. 2023. p. 272–275. Disponível em: <https://doi.org/10.13052/rp-9788770040723.053>. Acesso em: 31/05/2025.

DINO. **Jovens mudam de cidade para fazer curso superior**. 2023. Disponível em: <https://valor.globo.com/patrocinado/dino/noticia/2023/07/12/jovens-mudam-de-cidade-para-fazer-curso-superior.ghtml>. Acesso em: 06/05/2025.

FIGMA. **Figma Design**. 2025. Disponível em: <https://www.figma.com/pt-br/design/>. Acesso em: 20/07/2025.

FIREBASE. **Developer documentation for Firebase**. 2025. Disponível em: <https://firebase.google.com/docs>. Acesso em: 20/07/2025.

FJP. **Déficit Habitacional no Brasil**. Fundação João Pinheiro. 2024. Disponível em: <https://fjp.mg.gov.br/deficit-habitacional-no-brasil/>. Acesso em: 04/06/2025.

FLUTTER. **Flutter documentation**. 2025. Disponível em: <https://docs.flutter.dev/>. Acesso em: 31/05/2025.

FREITAS, A. **ERA DIGITAL: Como o Mercado Imobiliário tem reagido?** Artigo publicado no LinkedIn (CEO Terreno Livre). 2020. Disponível em: <https://www.linkedin.com/pulse/era-digital-como-o-mercado-imobili%C3%A1rio-tem-reagido-freitas-j%C3%BAnior/>. Acesso em: 07/05/2025.

GEHARD, T. E.; SILVEIRA, D. T. **Métodos da Pesquisa**. Porto Alegre: Editora da UFRGS, 2009.

GEIPELE, I.; KAUSKALE, L. The Influence of Real Estate Market Cycle on the Development in Latvia. **Procedia Engineering**, v. 57, p. 327–333, 2013. *Modern Building Materials, Structures and Techniques*. Disponível em: <https://doi.org/10.1016/j.proeng.2013.04.044>. Acesso em: 31/05/2025.

HAMDALAH, A. Z. *et al.* A Modular Architecture Framework for Flutter Using Clean Architecture and Design Patterns: A Software Quality Perspective. **2025 8th International Conference on Informatics and Computational Sciences (ICICoS)**, p. 160–165, 2025. Disponível em: <https://api.semanticscholar.org/CorpusID:284806723>. Acesso em: 16/12/2025.

HUSSAIN, W. *et al.* Model Driven Engineering and UML: A Systematic Literature Review. **Computers**, v. 9, n. 4, p. 56, 2020.

IBGE. **92,5% domicílios tinham acesso à Internet no Brasil**. 2023. Acesso em: 06/05/2025.

- IDC. **Smartphone Market Share**. International Data Corporation. 2025. Disponível em: <https://www.idc.com/promo/smartphone-market-share/>. Acesso em: 31/05/2025.
- IPEA. **Habitação e Saneamento**. Instituto de Pesquisa Econômica Aplicada. 2019. Disponível em: <https://www.ipea.gov.br/portal/categorias/263-retratos-banner-principal/retratos-banner-indicadores-habitacao-e-saneamento>. Acesso em: 04/06/2025.
- EL-KASSAS, W. S. *et al.* Taxonomy of Cross-Platform Mobile Applications Development Approaches. **Ain Shams Engineering Journal**, v. 8, n. 2, p. 163–190, 2017. Disponível em: <https://doi.org/10.1016/j.asej.2015.08.004>. Acesso em: 07/05/2025.
- KUNWAR, S. *et al.* Development of an Android Application for Rooms on Rent. **International Journal for Research in Applied Science & Engineering Technology**, v. 11, n. 6, p. 230–234, 2023. Disponível em: <https://doi.org/10.22214/ijraset.2023.53590>. Acesso em: 31/05/2025.
- LIU, Y. *et al.* Real estate rental market: a 10-year bibliometric-based review. **Economic Research-Ekonomska Istraživanja**, Routledge, v. 34, n. 1, p. 1752–1788, 2020. Disponível em: <https://doi.org/10.1080/1331677X.2020.1848605>. Acesso em: 31/05/2025.
- LOPES, L. *et al.* Metodologias ágeis: explorando o impacto do scrum e do kanban na qualidade e produtividade do software. **Revista Multidisciplinar do Nordeste Mineiro**, v. 3, n. 2, 2024. Disponível em: <https://remunom.ojsbr.com/multidisciplinar/article/view/3060>. Acesso em: 12/06/2025.
- MALLADI, N. S. N. R.; KUMAR, S. Human-Centered Product Design in the Tech Industry: Prioritizing User Needs in the Digital Landscape. **Journal of Quantum Science and Technology**, Resagate Global, v. 2, n. 1, p. 727–744, 2025. Disponível em: <http://www.jqst.org/>. Acesso em: 21/07/2025.
- MALLIK, R. *et al.* Development of An Android Application for Viewing Covid-19 Containment Zones and Monitoring Violators Who are Trespassing into It Using Firebase and Geofencing. **Transactions of the Indian National Academy of Engineering**, v. 5, n. 2, p. 163–179, 2020. Disponível em: <https://doi.org/10.1007/s41403-020-00137-3>. Acesso em: 16/12/2025.
- MANALU, S. R. *et al.* Development and evaluation of mobile application for room rental information with chat and push notification. *In*: 2016 International Conference on Information Management and Technology (ICIMTech). 2016. p. 7–11. Disponível em: <https://doi.org/10.1109/ICIMTech.2016.7930293>. Acesso em: 07/05/2025.
- MEILLER, D. Flutter: The Future of Application Development? *In*: KOMMERS, P. *et al.* (org.). **Proceedings of the International Conference on Applied Computing**. IADIS, 2022. p. 197–200. Disponível em: <https://www.iadisportal.org/digital-library/flutter-the-future-of-application-development>. Acesso em: 07/05/2025.
- MILKOVICH, K. *et al.* ZenDen - A Personalized House Searching Application. *In*: 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications

(BigDataService). 2020. p. 173–178. Disponível em: <https://doi.org/10.1109/BigDataService49289.2020.00034>. Acesso em: 31/05/2025.

MODULAR. **What is Modular?** 2025. Disponível em: <https://modular.flutterando.com.br/docs/intro/>. Acesso em: 20/07/2025.

OLIVEIRA, F. F. d. **Projeto de Mapeamento de Processos de Negócio: Aplicação de Metodologias Ágeis**. 2020. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção) – Faculdade de Engenharia, Universidade Federal de Juiz de Fora, Juiz de Fora. Disponível em: <https://www2.ufjf.br/engenhariadeproducao/files/2021/06/felipeferreiradeoliveiraassinado.pdf>. Acesso em: 12/06/2025.

PEREIRA, A. M. **Cidade Média e Região: O Significado de Montes Claros no Norte de Minas Gerais**. 2007. Tese (Doutorado em Geografia) – Instituto de Geografia, Universidade Federal de Uberlândia, Uberlândia. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/15921/1/Anete.pdf>. Acesso em: 04/06/2025.

ROULAC, S. The Strategic Real Estate Framework: Processes, Linkages, Decisions. **Journal of Real Estate Research**, Routledge, v. 12, n. 2, p. 323–346, 1996. Disponível em: <https://doi.org/10.1080/10835547.1996.12090843>. Acesso em: 21/07/2025.

SILVA, A. C. A.; GRACIANO, F. Desenvolvimento Ágil de Software. **RECIMA21 - Revista Científica Multidisciplinar**, v. 3, n. 12, 2022. Disponível em: <https://doi.org/10.47820/recima21.v3i12.2157>. Acesso em: 08/06/2025.

SIQUEIRA, B.; BRITTO, V. **Censo 2022: um em cada cinco brasileiros mora em domicílio alugado**. 2024. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/42197-censo-2022-um-em-cada-cinco-brasileiros-mora-em-domicilio-alugado>. Acesso em: 06/05/2025.

TSUI, F.; KARAM, O.; BERNAL, B. **Essentials of Software Engineering**. 3. ed. Burlington: Jones e Bartlett Learning, 2013.

VARTANIAN, P. R. *et al.* Macroeconomic and Financial Variables' Influence on Brazilian Stock and Real Estate Markets: A Comparative Analysis in the Period from 2015 to 2019. **Modern Economy**, v. 13, n. 5, p. 747–769, 2022. Disponível em: <https://doi.org/10.4236/me.2022.135040>. Acesso em: 08/06/2025.

VIEIRA, A. B.; ROMA, C. M.; MIYAZAKI, V. K. Cidades médias e pequenas: uma leitura geográfica. **Caderno Prudentino de Geografia**, v. 1, n. 29, p. 135–156, 2020. Disponível em: <https://revista.fct.unesp.br/index.php/cpg/article/view/7415>. Acesso em: 04/06/2025.

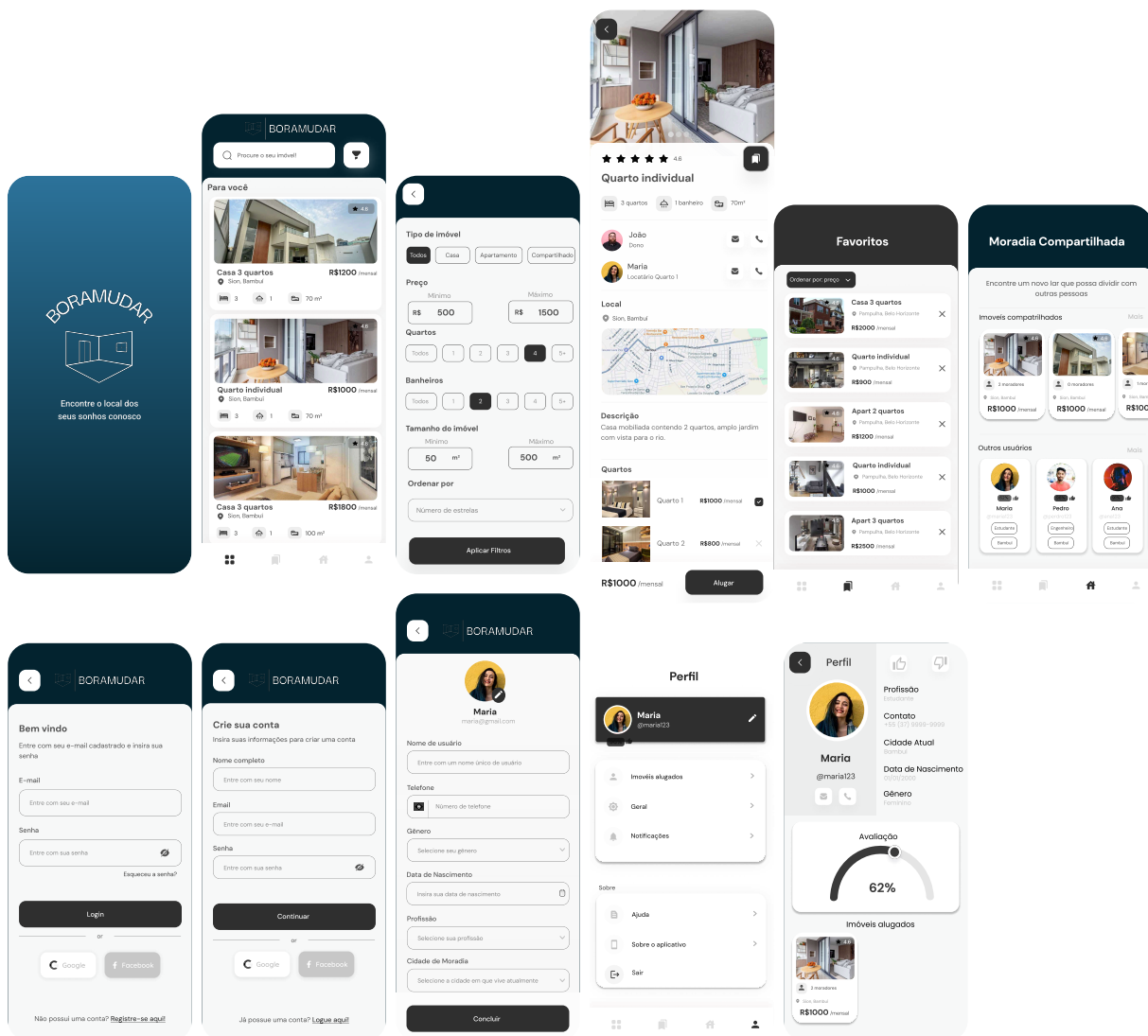
VOUMICK, D. *et al.* Development of Online Based Smart House Renting Web Application. **Journal of Software Engineering and Applications**, v. 14, n. 7, p. 312–328, 2021. Disponível em: <https://doi.org/10.4236/jsea.2021.147019>. Acesso em: 31/05/2025.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier Editora Ltda, 2009.

ZAYAT, W.; SENVAR, O. Framework Study for Agile Software Development Via Scrum and Kanban. **International Journal of Innovation and Technology Management**, v. 17, n. 04, p. 2030002, 2020. Disponível em: <https://doi.org/10.1142/S0219877020300025>. Acesso em: 08/06/2025.

APÊNDICES

APÊNDICE A - PROTOTIPAGEM NO FIGMA



APÊNDICE B - DEPENDÊNCIAS DO PROJETO

```
1 dependencies:
2   flutter:
3     sdk: flutter
4   firebase_core: ^4.2.1
5   firebase_auth: ^6.1.2
6   cloud_firestore: ^6.0.2
7   firebase_storage: ^13.0.2
8   cupertino_icons: ^1.0.8
9   cached_network_image: ^3.4.1
10  google_fonts: ^6.3.2
11  google_sign_in: ^6.2.1
12  image_picker: ^1.2.0
13  google_maps_flutter: ^2.6.0
14  geocoding: ^3.0.0
15  url_launcher: ^6.2.5
```

APÊNDICE C - IMPLEMENTAÇÃO WIDGET

```
1 Padding(  
2   padding: const EdgeInsets.all(12),  
3   child: Column(  
4     crossAxisAlignment: CrossAxisAlignment.start,  
5     children: [  
6       Row(  
7         children: [  
8           Expanded(  
9             child: Text(  
10              title,  
11              maxLines: 1,  
12              overflow: TextOverflow.ellipsis,  
13              style: const TextStyle(  
14                fontSize: 16,  
15                fontWeight: FontWeight.bold,  
16              ),  
17            ),  
18          ),  
19          const SizedBox(width: 8),  
20          Text(  
21            "R\${price.toString()} /mensal",  
22            style: const TextStyle(  
23              fontSize: 16,  
24              fontWeight: FontWeight.bold,  
25              color: Colors.teal,  
26            ),  
27          ),  
28        ],  
29      ),  
30    ],  
31  ),  
32 ),
```

APÊNDICE D - MAPEAMENTO DE ROTAS

```
1 static final Map<String, Widget Function(BuildContext, dynamic)> routes = {
2     splash: (context, args) =>
3         const SplashTransition(nextPage: MainNavigationPage()),
4     main: (context, args) => const MainNavigationPage(),
5     login: (context, args) => const LoginPage(),
6     home: (context, args) => const HomePage(),
7     houses: (context, args) => const HousesPage(),
8     favorites: (context, args) => const FavoritesPage(),
9     profile: (context, args) => const ProfilePage(),
10    filter: (context, args) => const FilterPage(),
11    allHouse: (context, args) => const AllHousesPage(),
12    allUsers: (context, args) => const AllUsersPage(),
13    notifications: (context, args) => const NotificationsPage(),
14    signup: (context, args) => const SignupPage(),
15    signup2: (context, args) {
16        final data = args as Map<String, dynamic>;
17        return SignupPage2(
18            name: data['name'],
19            email: data['email'],
20            password: data['password'],
21        );
22    },
23    profileDetail: (context, args) {
24        final data = args as Map<String, dynamic>;
25        return ProfileDetailPage(userId: data['userId']);
26    },
27    houseDetail: (context, args) {
28        final data = args as Map<String, dynamic>;
29        return PropertyDetailPage(propertyId: data['propertyId']);
30    },
31 };
```

APÊNDICE E - CADASTRO E LOGIN GOOGLE

E.1 Login usando Google

```
1 Future<User?> loginWithGoogle() async {
2   final googleUser = await _googleSignIn.signIn();
3   if (googleUser == null) return null;
4
5   final googleAuth = await googleUser.authentication;
6   final credential = GoogleAuthProvider.credential(
7     accessToken: googleAuth.accessToken,
8     idToken: googleAuth.idToken,
9   );
10
11  final userCredential = await _auth.signInWithCredential(credential);
12  final user = userCredential.user;
13
14  if (user != null) {
15    final doc = await _firestore.collection('users').doc(user.uid).get();
16    if (!doc.exists) {
17      await _auth.signOut();
18      await _googleSignIn.signOut();
19      throw Exception("Usuário não cadastrado. Crie uma conta primeiro.");
20    }
21  }
22  return user;
23 }
```

E.2 Criação de usuário com Google

```
1 Future<Map<String, String?>> signUpWithGoogle() async {
2     final googleUser = await _googleSignIn.signIn();
3     if (googleUser == null) return null;
4
5     final googleAuth = await googleUser.authentication;
6     final credential = GoogleAuthProvider.credential(
7         accessToken: googleAuth.accessToken,
8         idToken: googleAuth.idToken,
9     );
10
11     final userCredential = await _auth.signInWithCredential(credential);
12     final user = userCredential.user;
13
14     return {
15         'name': user?.displayName ?? googleUser.displayName,
16         'email': user?.email ?? googleUser.email,
17     };
18 }
```

APÊNDICE F - GEOLOCALIZAÇÃO GOOGLE MAPS

```
1 Future<Map<String, dynamic>> _getProperty() async {
2   if (_cachedData != null) {
3     return _cachedData!; // retorna cache se já carregou
4   }
5   final doc = await _propertyController.getPropertyRaw(widget.propertyId);
6   final data = doc.data() ?? {};
7
8   if (data.containsKey('latLong')) {
9     final geoPoint = data['latLong'];
10    _propertyPosition = LatLng(
11      geoPoint.latitude,
12      geoPoint.longitude,
13    );
14  }
15  _cachedData = data; // salva no cache
16  return data;
17 }
```