



**INSTITUTO FEDERAL DO PIAUÍ – IFPI
PRÓ-REITORIA DE ENSINO
DIRETORIA DE ENSINO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**RELATÓRIO TÉCNICO DE SOFTWARE
HOUSING - IMÓVEIS PARA ESTUDANTES**

EDMARQUES SANTOS DE LIMA

JESIEL VIANA DA SILVA
ORIENTADOR

**PICOS, PI
2024**

EDMARQUES SANTOS DE LIMA

HOUSING - IMÓVEIS PARA ESTUDANTES

Relatório Técnico de Software apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos.

Orientador: Prof. Me. Jesiel Viana da Silva

PICOS
2024

RESUMO

A mudança de estudantes para novas cidades em busca de oportunidades de educação superior está se tornando cada vez mais comum no Brasil. Contudo, essa jornada envolve desafios significativos, sendo a busca por moradia adequada um dos principais. Este estudo explora esses desafios e apresenta como solução o desenvolvimento de uma aplicação web que visa fornecer informações essenciais e detalhadas para auxiliar os estudantes na escolha de moradias adequadas na cidade de Picos–PI. Este relatório descreve detalhadamente as tecnologias implementadas no projeto e inclui uma avaliação feita por uma profissional em garantia de qualidade. Essa avaliação confirma a eficiência e a confiabilidade do sistema, ressaltando o empenho em manter um alto padrão de qualidade tanto na execução quanto na entrega do projeto. A aplicação é projetada para atender especificamente às necessidades desses estudantes, facilitando sua transição para a vida universitária em um novo ambiente.

***Palavras-chave:** Moradia; Estudantes; Tecnologia; Aluguel; Picos–PI.

SUMÁRIO

1. INTRODUÇÃO	5
1.1. OBJETIVOS	6
1.1.1. GERAL	6
1.1.2. ESPECÍFICOS	6
2. TECNOLOGIAS ENVOLVIDAS	7
3. MODELAGEM DO PROJETO	10
3.1. LEVANTAMENTO DE REQUISITOS	10
3.2. DIAGRAMAS DE CLASSE	11
3.3. ARQUITETURA DO SISTEMA	11
3.4. INTERFACE	12
4. SOFTWARE	18
4.1. IMPLANTAÇÃO	19
4.2. TESTES	19
4.3. TESTE DE DESEMPENHO	20
4.4. TESTES EXPLORATÓRIOS	21
5. CONSIDERAÇÕES FINAIS	21
REFERÊNCIAS	22

1. INTRODUÇÃO

A escolha de se mudar para cidades maiores a fim de frequentar instituições de ensino superior de maior prestígio tem se tornado cada vez mais comum, refletindo o desejo dos estudantes de obter uma educação de qualidade que os prepare melhor para o mercado de trabalho. Graças às transformações no processo do Exame Nacional do Ensino Médio (Enem) e à introdução do Sistema de Seleção Unificada (Sisu), que unificou o Programa Universidade para Todos (Prouni) e o Fundo de Financiamento Estudantil (Fies), a realidade dos estudantes brasileiros mudou significativamente (BLOG MACKENZIE, 2020).

Com isso, o aumento da busca de moradia é uma atividade comum na vida desses estudantes que se deslocam para outras cidades para trabalhar e estudar. Estudo publicado pelo IBGE (2018), aponta que estudantes universitários brasileiros percorrem, em média, 92 quilômetros para frequentar seus cursos.

Esses desafios apresentam uma demanda por moradia adequada e acessível para esses estudantes que buscam novas oportunidades educacionais. A necessidade de encontrar um lugar para morar em outra cidade tornou-se uma preocupação constante para muitos jovens que desejam cursar a faculdade.

Em particular, alunos que se mudam para a cidade de Picos-PI relatam que encontrar moradia em uma cidade desconhecida pode ser um desafio. Os estudantes enfrentam diversas dificuldades, como a falta de informações sobre o mercado imobiliário local, os altos preços dos aluguéis e a escassez de opções adequadas às suas necessidades e orçamento, conforme dados de pesquisa de campo realizadas com os estudantes do Instituto Federal do Piauí, campus Picos.

Dessa forma, com os avanços constantes da tecnologia, surgem cada vez mais ferramentas que proporcionam praticidade e economia no dia a dia da população. Um exemplo notável é o Airbnb, que se destaca não apenas por sua conveniência e economia, mas também pela rapidez na realização dos objetivos desejados. Conforme Brandão (2019), somente no ano de 2018 o Airbnb intermediou 3,8 milhões de locações e é considerada atualmente a plataforma de hospedagem que mais cresce em todo o mundo. Ele atua como um mercado online

de hospedagem para aluguel temporário, funciona como uma comunidade em que pessoas que possuem casas, sobrados ou até mesmo quartos disponíveis, divulgam esse espaço livre para pessoas que estão viajando e pretendem ficar poucos dias no local, com isso percebe-se que a ferramenta tem maior foco em atividades de turismo, e não atende bem às necessidades do público estudantil (OLIVEIRA, MACEDO e PIRES, 2021, p. 2).

Diante do contexto apresentado, foi desenvolvido uma solução que atendesse as principais necessidades do público estudantil em questão, visando disponibilizar aos estudantes, as principais informações que fazem a diferença para a escolha de local de moradia que também sirva de estudo.

1.1.OBJETIVOS

1.1.1. GERAL

O objetivo principal deste trabalho é desenvolver uma aplicação web, destinada a otimizar o processo de busca e seleção de imóveis para locação, especialmente projetada para atender às necessidades específicas do público estudantil. Esta solução visa facilitar a identificação de moradias compartilhadas a preços acessíveis, que estejam estrategicamente localizadas em relação às instituições de ensino e ofereçam proximidade a serviços essenciais como transporte público, supermercados e padarias, atendendo assim às principais demandas dos estudantes que se deslocam para outras cidades para prosseguir com seus estudos.

1.1.2. ESPECÍFICOS

- Realizar um levantamento das dificuldades enfrentadas na procura de moradia pelos estudantes que se mudam para a cidade de Picos-PI;
- Elaborar uma proposta de solução tecnológica que entregue informações detalhadas sobre os imóveis disponíveis, incluindo, preços, localização, pontos de referência relevantes e imagens atualizadas;
- Documentar arquitetura da aplicação;

- Desenvolver a ferramenta e submetê-la a testes de desempenho e exploratório.

2. TECNOLOGIAS ENVOLVIDAS

As tecnologias escolhidas para o desenvolvimento da ferramenta tiveram foco na atualidade, desempenho e prévio conhecimento do desenvolvedor, visando proporcionar um desenvolvimento rápido e funcional da aplicação. A mesma será apresentada nos contextos descritos.

Tecnologia	Versão	Descrição
TypeScript	5.2.2	O <i>TypeScript</i> é um <i>superset</i> da linguagem <i>JavaScript</i> , criado pela Microsoft para adicionar a tipagem estática a linguagem e integrar os recursos utilizados na orientação a objetos como a própria tipagem, modificadores de acesso, interfaces e classes abstratas (TYPESCRIPT, 2023). O <i>TypeScript</i> foi escolhido por ter um preparo mais rico para implementação da orientação a objetos, modelo de programação escolhido, e pela facilidade de se trabalhar numa ferramenta de edição de código.
Express	4.18.2	O <i>Express</i> é uma biblioteca de desenvolvimento de <i>APIs RESTful</i> com <i>Node.js</i> , e facilita o desenvolvimento por ter configuração fácil e agilidade no desenvolvimento, e por esses motivos foi escolhido como tecnologia para desenvolvimento da API (EXPRESS, 2024).
PostgreSQL	15	<i>PostgreSQL</i> é um poderoso sistema de banco de dados relacional de objeto de código aberto

		com mais de 35 anos de desenvolvimento ativo que lhe rendeu uma forte reputação de confiabilidade, robustez de recursos e desempenho (POSTGRESQL, 2024). O PostgreSQL foi escolhido pela confiabilidade e pela facilidade em encontrar plataformas que ofereciam instâncias gratuitas do banco para utilização.
Prisma	5.3.1	O <i>Prisma</i> é um <i>framework</i> para trabalhar com banco de dados, sendo muito utilizado a sua parte de ORM, que conecta a vários bancos relacionais e alguns não relacionais (PRISMA, 2024). O mesmo foi escolhido por sua versatilidade, e facilidade para fazer operações básicas e complexas como consultas que envolvem múltiplos relacionamentos, aumentando a produtividade durante o desenvolvimento.
VueJS	3.3.8	O <i>VueJS</i> é um <i>framework</i> de desenvolvimento <i>front-end open source</i> criado em 2014 que implementa renderização em <i>Single Page Application (SPA)</i> , que renderiza as <i>views</i> sem recarregar a página, trazendo performance a aplicação (VUEJS, 2024). O <i>VueJS</i> foi escolhido pela facilidade de desenvolvimento, dando produtividade, e pelo vasto ecossistema de bibliotecas para utilização em conjunto como o <i>Veutify</i> apresentado posteriormente.
NuxtJS	3.8.1	O <i>NuxtJS</i> é um <i>framework</i> para <i>VueJS</i> , o qual implementa renderização <i>server-side</i> , funcionalidades de <i>Search Engine Optimization</i>

		(SEO), e automatização de configurações, e foi utilizado para aumentar a produtividade com o <i>VueJS</i> , evitando configurações manuais que seriam necessárias utilizando somente o <i>VueJS</i> (NUXTJS, 2024).
Vuetify	3.4.3	O <i>Vuetify</i> é uma biblioteca de componentes para <i>VueJS</i> , que implementa o estilo do <i>material design</i> do Google (VUETIFY, 2024). E foi escolhido por facilitar o design da aplicação e conter configurações prontas para utilizar, CSS Grid, tipografia, alinhamento de itens e responsividade de interface.
Pinia	2.1.7	O <i>Pinia</i> é uma biblioteca de gerenciamento de estado global para <i>VueJS</i> , que possibilita o compartilhamento de estado entre componentes de maneira simples (PINIA, 2024). O <i>Pinia</i> foi escolhido para facilitar o compartilhamento de dados entre componentes que não possui descendência direta, para centralizar rotinas específicas evitando repetição de código e por ser o gerenciador de estado mais indicado para o <i>Vue 3</i> , por implementar o modo de <i>Composition API</i> .
Git	1.1	O <i>Git</i> é um sistema de controle de versão distribuído de código aberto, podendo ser utilizado para versionar quaisquer tipos de arquivo, e principalmente utilizado para o versionamento de código de software (GIT, 2024).

3. MODELAGEM DO PROJETO

Neste capítulo, será abordado a modelagem do projeto, detalhando-a por meio do levantamento de requisitos e do diagrama de classes. Essas informações podem ser acompanhadas nos tópicos a seguir.

3.1. LEVANTAMENTO DE REQUISITOS

Os requisitos são definidos em:

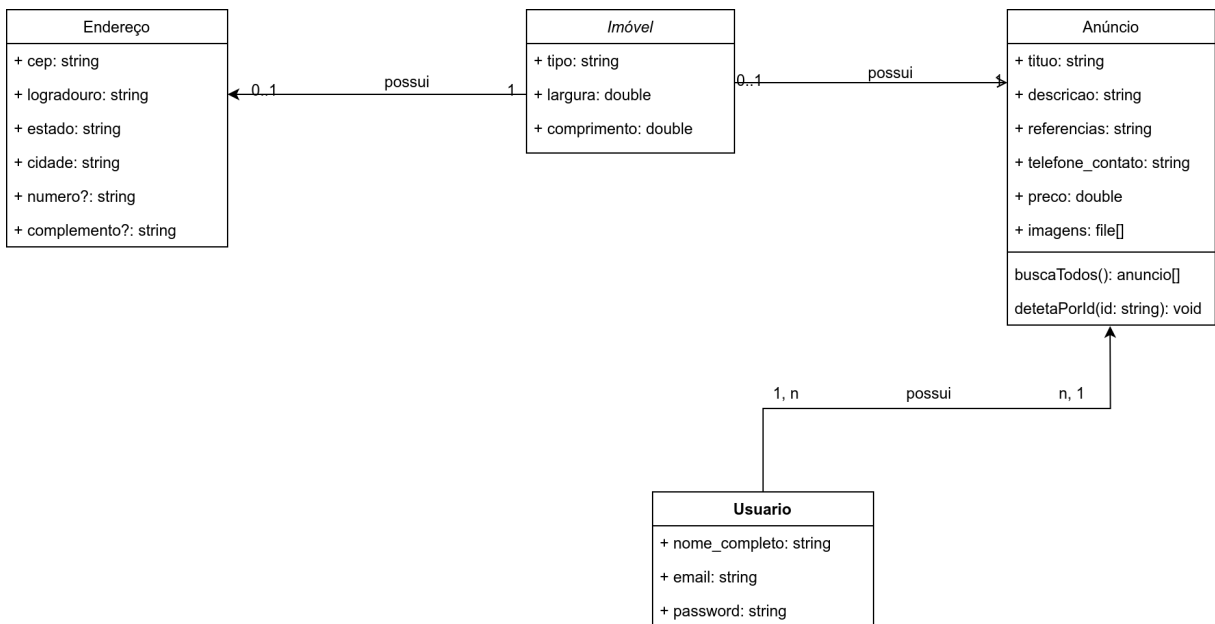
- Requisitos funcionais:
 1. Cadastro de usuário – O sistema deve oferecer a capacidade para usuários sem conta realizarem seu cadastro, preenchendo informações pessoais como nome, e-mail e telefone;
 2. Login de usuário – O sistema deve possibilitar que usuários com contas existentes efetuem o login no sistema;
 3. Cadastro do imóvel – O usuário logado deve ser capaz de criar um anúncio de imóvel, disponibilizando uma galeria de imagens, e informações de localização, tamanho, disponibilidade, preço, pontos de referência e descrição;
 4. Pesquisa por local – O sistema deve fornecer uma barra de pesquisa de local para listagem dos imóveis;
 5. Listagem de imóveis – O sistema deve fornecer uma listagem dos imóveis disponíveis com base em filtro de local aplicado;
 6. Visualização de detalhes de anúncio – O sistema deve permitir a visualização completa do anúncio de imóvel;
 7. Edição de anúncio – O sistema deve permitir a edição das informações de um anúncio;
 8. Exclusão de anúncio – O sistema deve permitir a exclusão de um anúncio.
- Requisitos não-funcionais:
 1. Disponibilidade – O sistema deve estar acessível 24 horas por dia, 7 dias por semana;
 2. Usabilidade – O sistema deve ter interface intuitiva;
 3. Responsividade – O sistema deve ter interface adaptável aos diversos tamanhos de tela de celulares, notebooks e desktops;

4. Compatibilidade – O sistema deve ser compatível com os principais navegadores da atualidade, Google Chrome, Microsoft Edge, Samsung Internet e Firefox.

3.2. DIAGRAMAS DE CLASSE

Os diagramas de classe são utilizados para exemplificar o funcionamento das principais classes do sistema, e é a representação da estrutura e relações das classes que servem de modelo para objetos (FONSECA, 2022).

Figura 1 - Diagrama de classes



Fonte: Próprio Autor

3.3. ARQUITETURA DO SISTEMA

A arquitetura implementada adota o modelo cliente-servidor em três camadas. Na primeira camada, temos o front-end, que proporciona a interface para o usuário. Em seguida, a segunda camada consiste na *Application Programming Interface (API)*, responsável por consultar o banco de dados e retornar os resultados à camada anterior. Por fim, a terceira camada abrange o armazenamento de dados, onde a API realiza chamadas para utilização dos mesmos. Essa abordagem oferece

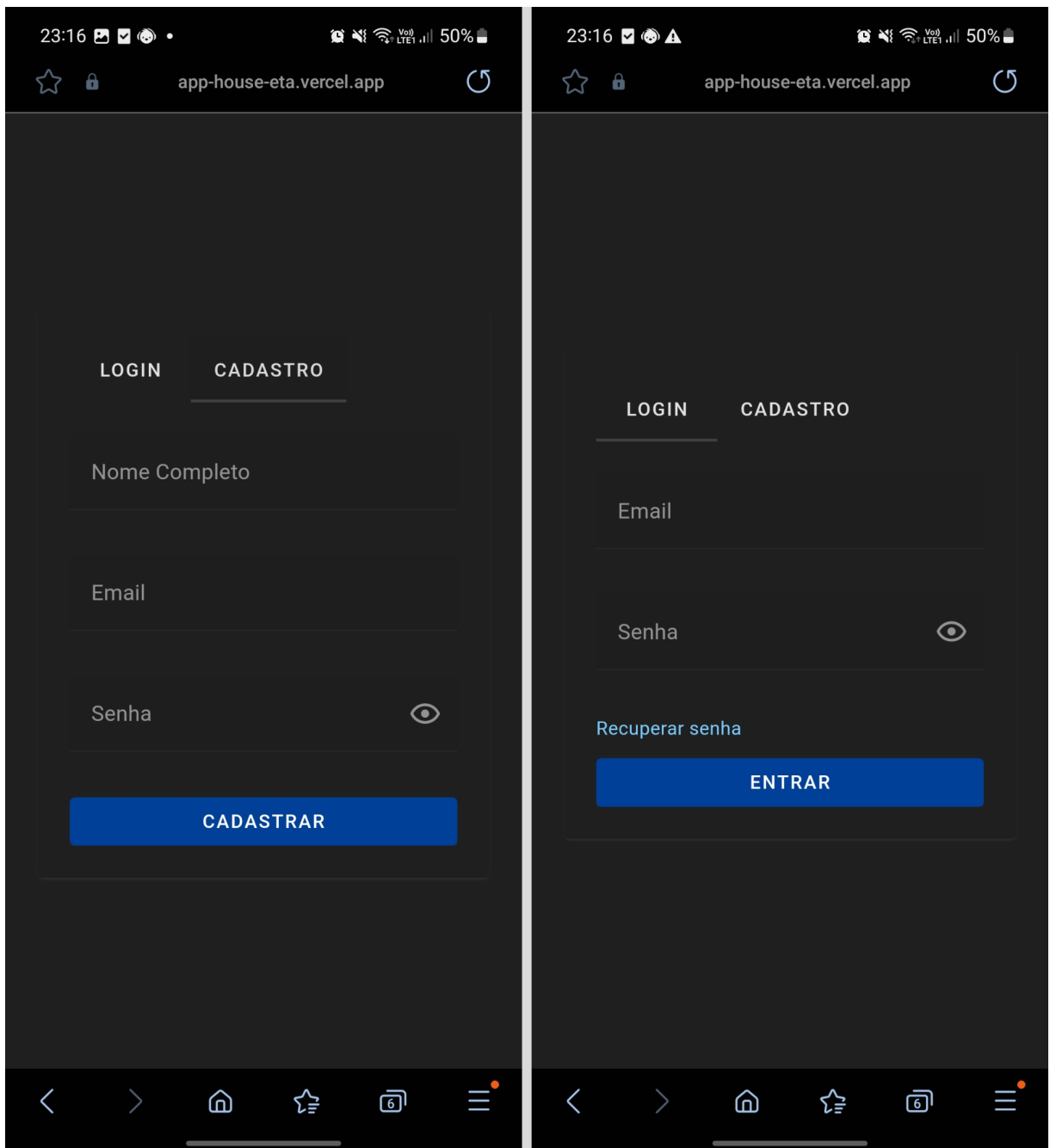
uma clara separação de responsabilidades e promove a manutenção e escalabilidade do sistema.

3.4. INTERFACE

Toda a interface do sistema foi construída com VueJS utilizando componentes do Vuetify, e está acessível em qualquer dispositivo com navegador web. A seguir serão apresentadas capturas de tela tiradas em um celular rodando o sistema no navegador Samsung Internet.

Na Figura 2, encontram-se as telas de cadastro e a tela de login do sistema.

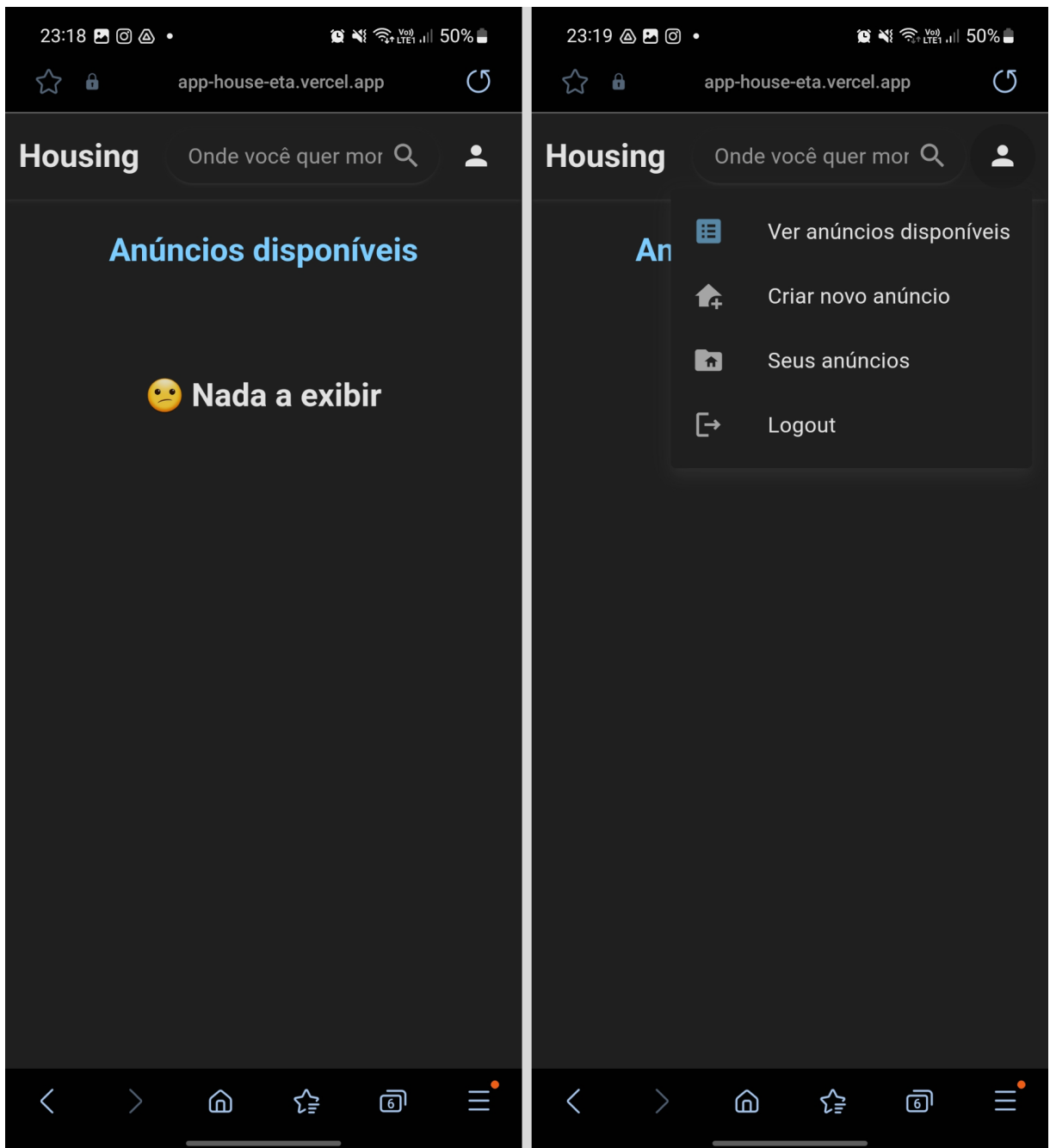
Figura 2 - Telas de Cadastro e Login



Fonte: Próprio Autor

Após a conclusão do cadastro, o usuário é redirecionado para a página inicial, conforme ilustrado na Figura 3. Essa página exibe os anúncios disponíveis e, na ausência destes, apresenta uma mensagem de aviso. A interface inclui também um menu com várias funcionalidades mostradas também na figura 3, tais como o cadastro de novos anúncios, visualização dos anúncios criados pelo usuário e a opção de sair do sistema.

Figura 3 - Página inicial sem anúncios e menu



Fonte: Próprio Autor

Quando o usuário seleciona a opção de criar um novo anúncio, é redirecionado para um formulário. Para que o anúncio seja publicado com sucesso, é necessário preencher todos os campos do formulário adequadamente, conforme ilustrado na figura 4.

Figura 4 - Formulário para cadastro de anúncios

The image displays two side-by-side screenshots of a mobile application interface for housing listings. Both screens feature a dark theme and a status bar at the top showing the time (23:20 on the left, 23:21 on the right) and battery level (50%).

The left screenshot shows the initial form with the title "Housing" and a user icon. Below the title, there is a large text input field with the placeholder "Insira o endereço do imóvel". At the bottom, there is a blue button labeled "AVANÇAR" and a back arrow icon.

The right screenshot shows the form filled out with the following information:

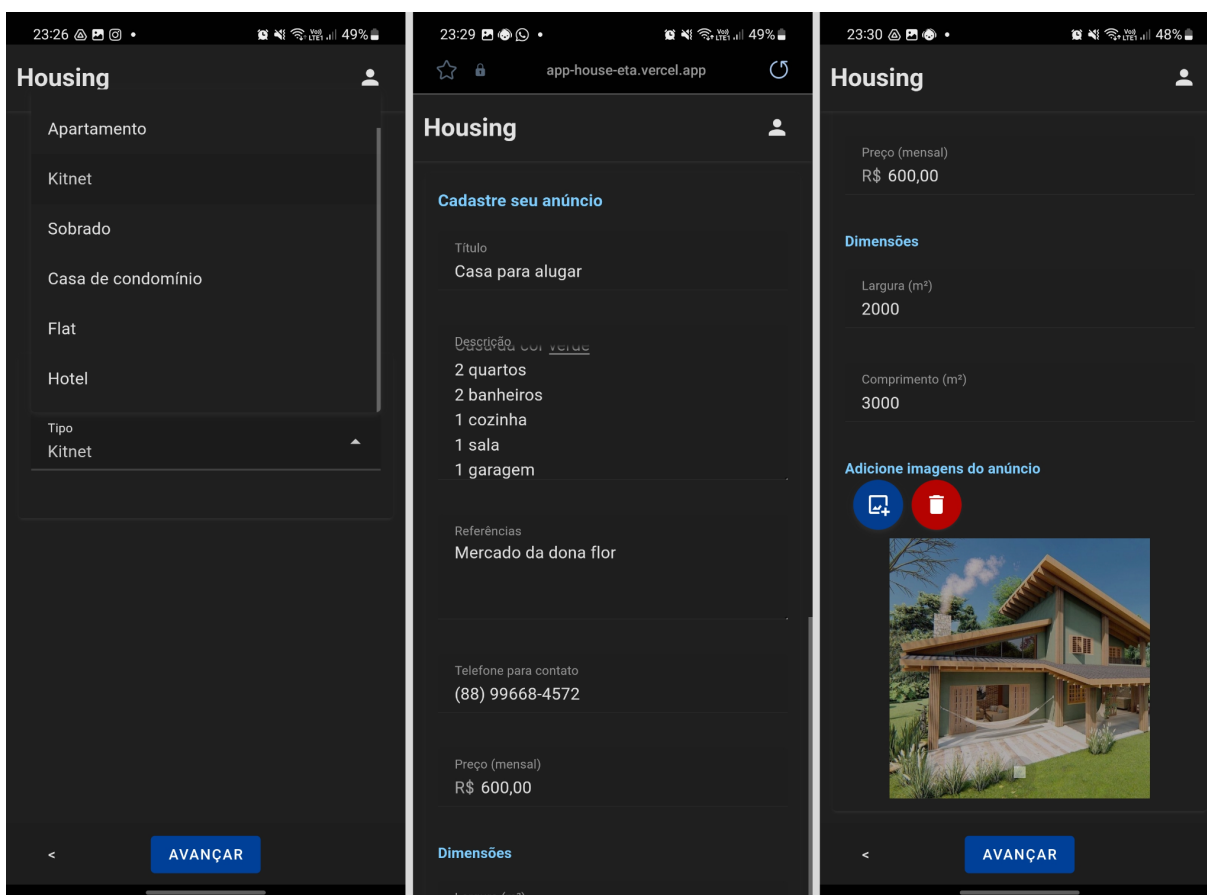
- CEP: 64605-460
- Estado: PI
- Bairro: Parque Industrial
- Cidade: Picos
- Logradouro: Quadra 04
- Número: (empty)
- Complemento: (Conj Waldemar M Santos)

At the bottom of the right screenshot, there is a blue button labeled "AVANÇAR" and a back arrow icon.

Fonte: Próprio Autor

Na etapa seguinte do formulário, é obrigatório incluir o tipo de imóvel, posteriormente será necessário informar os detalhes específicos do imóvel que está sendo disponibilizado para locação, adicionalmente, pode-se incluir imagens, conforme ilustrado na Figura 5.

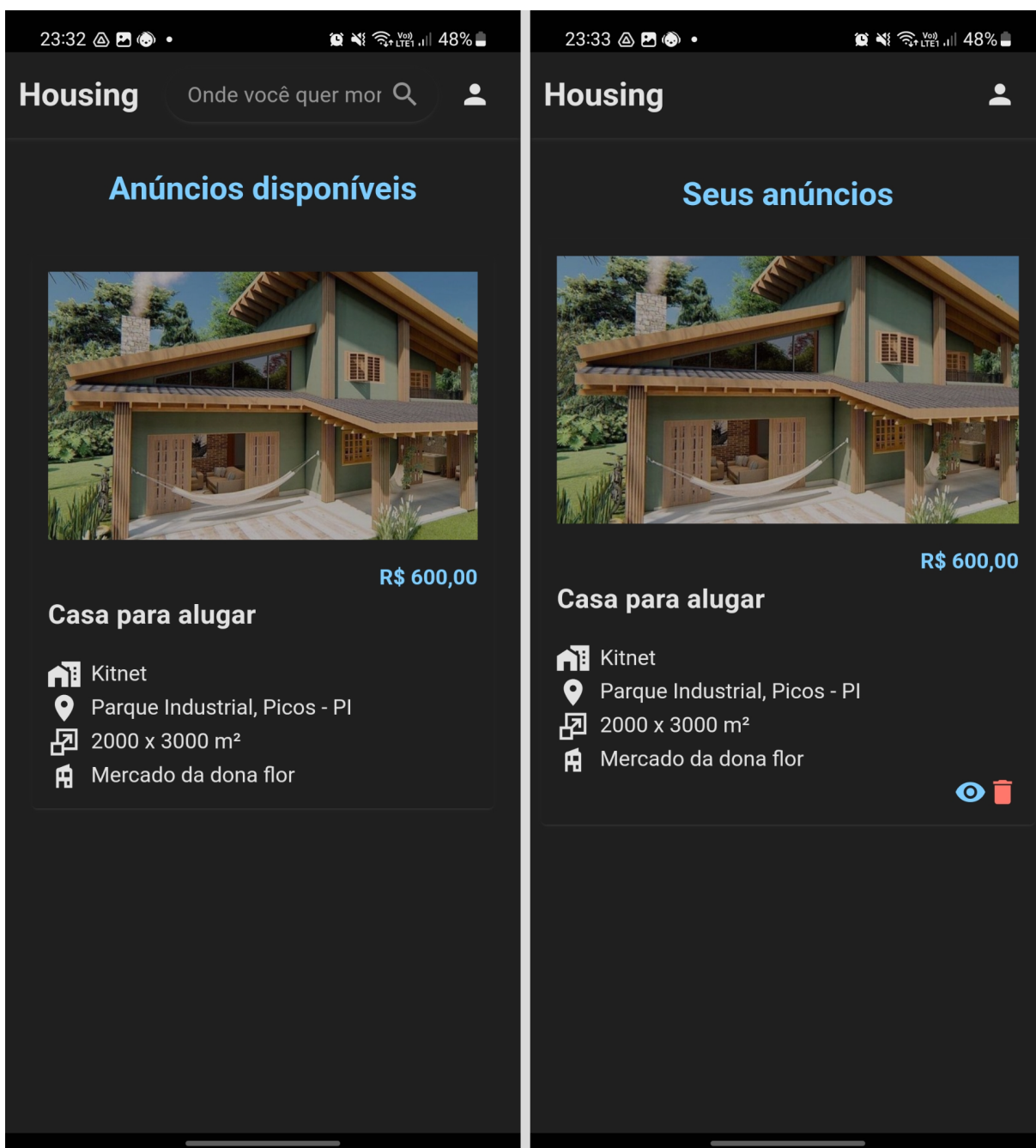
Figura 5 - Seleção do tipo de imóvel e detalhes do imóvel



Fonte: Próprio Autor

Após clicar no botão 'Avançar', o anúncio do usuário será cadastrado. Ele poderá visualizá-lo na tela principal do sistema, ou selecionar a opção 'Meus Anúncios', como exibido na Figura 6.

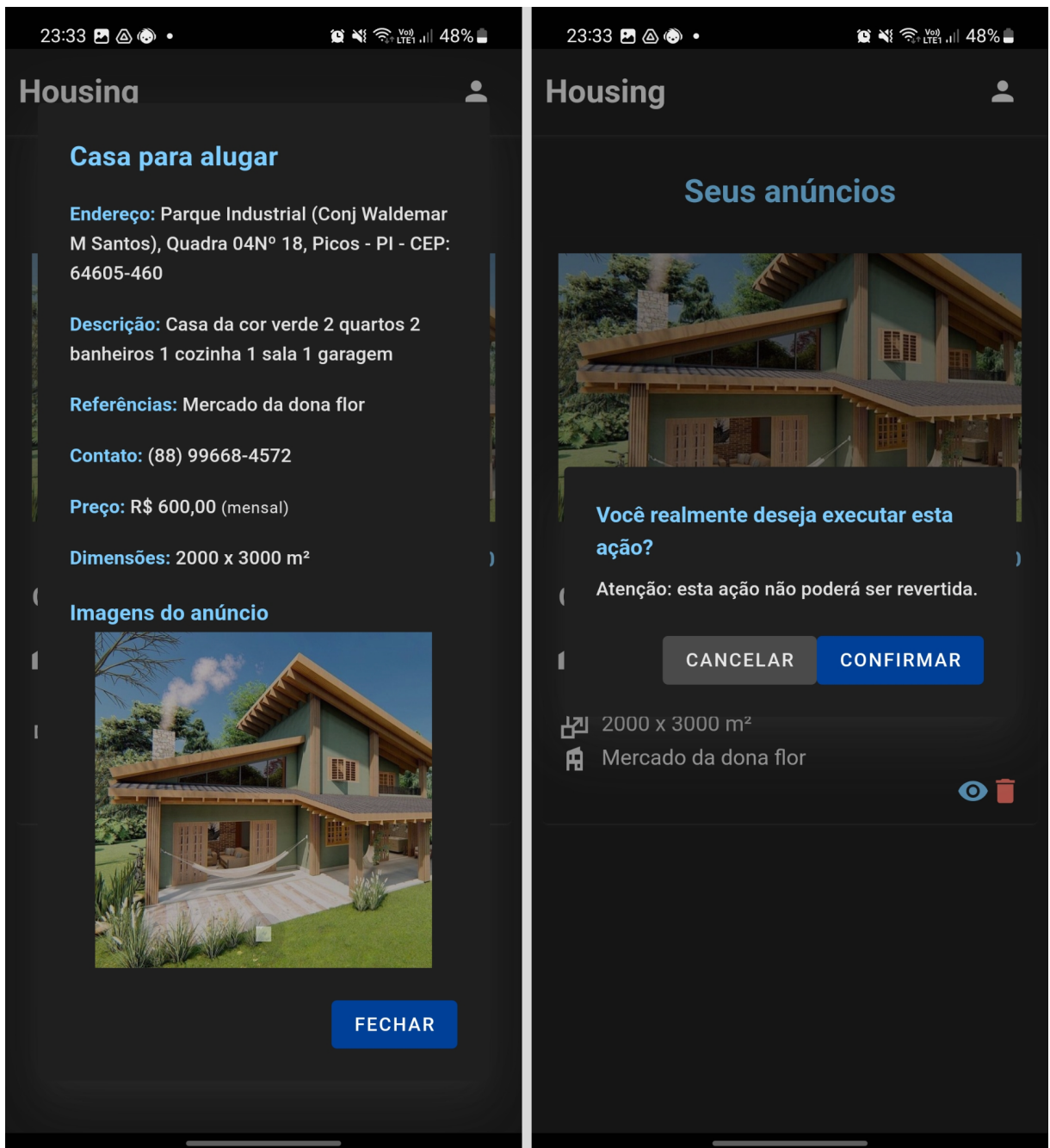
Figura 6 - Anúncio disponível na tela principal e funcionalidade meus anúncios



Fonte: Próprio Autor

Dentro da funcionalidade 'Meus Anúncios', o usuário terá a opção de visualizar os detalhes de um anúncio, ou excluir o anúncio conforme representado na Figura 7.

Figura 7 - Detalhes do anúncio e exclusão



Fonte: Próprio Autor

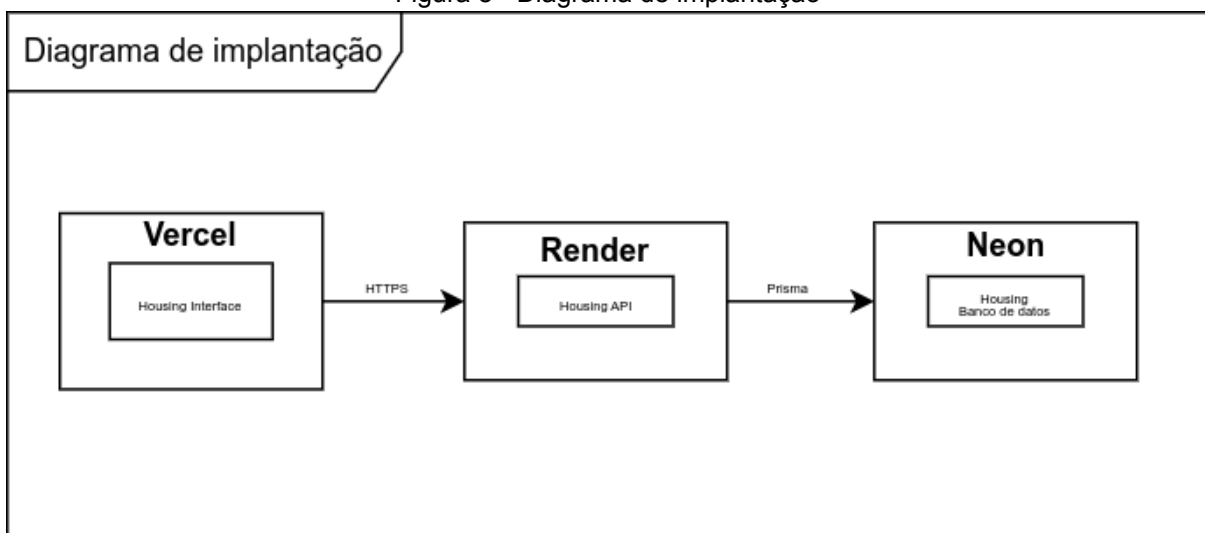
4. SOFTWARE

Este capítulo apresenta a implantação do sistema com diagrama exemplificando o processo e os resultados, sendo eles, teste de desempenho e testes exploratórios. A versão atual do software se encontra como mostrado no tópico [3.4. Interface](#) e as futuras implementações são encontradas nas [considerações finais](#).

4.1. IMPLANTAÇÃO

A implementação do software foi realizada de forma escalonada, seguindo uma abordagem que separa os componentes em três servidores distintos: front-end, back-end e banco de dados. Esse processo garantiu uma gestão mais eficiente de recursos e organização do sistema todo. O diagrama de *Unified Modeling Language* (UML), apresentado na Figura 8 oferece uma representação visual da arquitetura de implantação adotada para o software.

Figura 8 - Diagrama de implantação



Fonte: Próprio Autor

O código-fonte do front-end e back-end da aplicação estão armazenados no github, disponíveis nos links a seguir:

- Front-end: https://github.com/edmarques11/app_house_frontend
- Back-end: https://github.com/edmarques11/app_house_backend

4.2. TESTES

Nesta seção, será apresentada uma análise detalhada dos testes de infraestrutura e usabilidade realizados. Os procedimentos foram conduzidos conforme descrito no material de Marques (2023), que abrangeu testes de desempenho, além da utilização da ferramenta *Pagespeed Insights*. Adicionalmente, foi conduzido um teste exploratório no sistema, seguindo uma abordagem mais fluida e adaptativa para o teste de software. Essa abordagem permite que possa ser

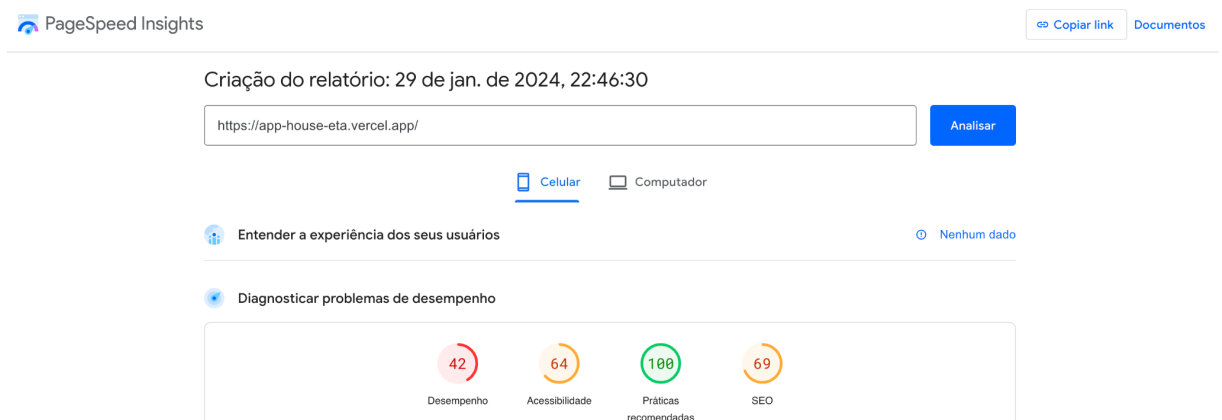
utilizado tanto a criatividade quanto a experiência técnica para identificar problemas de forma eficaz, conforme sugerido por (DE ABREU, 2016).

4.3. TESTE DE DESEMPENHO

Visando garantir a qualidade da plataforma, foi utilizado a ferramenta *Pagespeed Insights* para avaliação da qualidade de código e usabilidade. O *Pagespeed Insights* é uma coleção de ferramentas desenvolvidas pela Google, destinada a aprimorar o desempenho de websites *Pagespeed Insights* (2024). Essa ferramenta oferece a capacidade de identificar questões associadas à codificação, acessibilidade e desempenho.

A figura 9 mostra os resultados da ferramenta *Google Pagespeed Insights*, essencial para identificar e diagnosticar erros relacionados à acessibilidade e desenvolvimento em websites, oferecendo uma visão abrangente das áreas que requerem aprimoramento.

Figura 9 - Resultado do desempenho no PageSpeed Insights



Fonte - Próprio Autor

Através da ferramenta, foram identificados erros de acessibilidade, desempenho e SEO, como erros de desempenho foram apontados carregamento de CSS e *JavaScript* não utilizado, para acessibilidade foram apontados a falta do atributo *name* nos botões, links e itens de menu, falta do elemento *title* nas páginas e falta de contraste entre as cores de primeiro e segundo plano e para SEO foram apontados falta da tag *title* nas páginas, falta de tags de meta descrição e invalidez

do arquivo *robots.txt* do projeto. Esses problemas foram priorizados para correção como trabalho futuro, antes de serem apresentados ao usuário.

4.4. TESTES EXPLORATÓRIOS

Os testes exploratórios foram realizados por uma profissional em Garantia de Qualidade (Quality Assurance - QA), treinada não apenas para identificar erros, mas também para perceber e detectar discrepâncias que poderiam passar despercebidas por outros. Adicionalmente, o QA tem a habilidade de reconhecer e propor oportunidades de melhoria.

Durante os testes, foram identificados erros no cadastro de anúncios, como uma mensagem de erro incorreta após preencher os campos corretamente. Além disso, ao tentar carregar uma imagem grande, a mensagem de erro era confusa e continha um erro gramatical. Recomenda-se esclarecer o limite de tamanho para arquivos e corrigir o erro de português na mensagem. Outra falha identificada está no campo de valor, onde, ao cadastrar um anúncio com R\$ 1.000.000,00, o sistema impede a conclusão sem informar adequadamente o limite, causando confusões no processo de cadastro.

Também foram propostas melhorias, incluindo a otimização da velocidade do sistema. Observou-se que a barra de pesquisa só funciona adequadamente ao inserir a sigla do estado, o que indica a necessidade de aprimoramento dessa funcionalidade. Além disso, recomenda-se uma revisão no design da funcionalidade de adicionar imagens. Atualmente, sua proximidade com o ícone da lixeira pode levar ao equívoco de que serve para excluir o anúncio, não a imagem. Portanto, é sugerido realçar esta funcionalidade para evitar confusões.

A QA também destacou aspectos positivos, elogiando a interface intuitiva e a estabilidade do sistema. As funcionalidades além de serem de fácil utilização, são complementadas por uma organização lógica, contribuindo para uma experiência do usuário ainda mais fluida.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema Housing representa uma resposta direta a uma necessidade premente identificada na cidade de Picos-PI: a ausência de uma ferramenta centralizada para busca de imóveis disponíveis. Ao longo deste trabalho,

foi possível observar de perto os desafios enfrentados pelos estudantes universitários ao procurar por moradia na região. Com base nessas observações, foi implementado um *Minimal Viable Product (MVP)* que atende às necessidades essenciais dos usuários, oferecendo uma solução prática.

Ao alcançar os objetivos estabelecidos para este projeto, foi possível não apenas desenvolver uma ferramenta funcional, mas também fornecer uma contribuição significativa para a comunidade estudantil e para a cidade toda, ajudando a reduzir barreiras enfrentadas pelos estudantes ao se mudarem para uma nova cidade.

Contudo, é importante ressaltar que como MVP, o sistema possui diversas melhorias a serem implementadas em trabalhos futuros, como as que foram explanadas nos testes exploratórios e de desempenho, além de outras melhorias observadas no contexto geral da aplicação, como a necessidade de obrigatoriedade de informação de distância das instituições de ensino da cidade e informação quando o anúncio é do imóvel inteiro ou para dividir.

REFERÊNCIAS

IBGE. **Alunos de outros municípios percorrem quase 100 km para frequentar ensino superior.** Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/28047-alunos-de-outros-municipios-percorrem-quase-100-km-em-media-para-frequentar-ensino-superior/>>. Acesso em: 18 de junho de 2023.

BRANDÃO, Thales. **Airbnb cresce 71% no Brasil e registra mais de 3,8 milhões chegada de hóspedes em 2018.** Cidade Marketing, 18 de janeiro de 2019. Disponível em: <<https://www.cidademarketing.com.br/marketing/2019/01/18/airbnb-cresce-71-no-brasil-e-registra-mais-de-38-milhoes-chegada-de-hospedes-em-2018/>>. Acesso em: 24/06/2019.

DE ABREU, Jean Carlos; MARTINO, Flávio Augusto; SCHIAVONI, Marcos Alexandre. **Testes exploratórios em software.** Revista Eletrônica Científica do CRA-PR-RECC, v. 3, n. 1, p. 62-75, 2016.

EXPRESS. **Framework web rápido, flexível e minimalista para Node.js.** Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em: 30/01/2024.

FONSECA, Leticia. **Diagrama de classes.** Disponível em: <<https://pt.venngage.com/blog/diagrama-de-classe/>>. Acesso em: 04/07/2022.

GIT. **Git --fast-version-control.** Disponível em: <<https://git-scm.com/>>. Acesso em 30/01/2024.

GITHUB. **Let's build from here.** Disponível em: <<https://github.com/>>. Acesso em 30/01/2024.

MARQUES, Daniella Ferreira. **Artigo de Software Front-end.** Material fornecido para as disciplinas de Projeto Integrador I e III. Instituto Federal de Ciências e Tecnologia do Piauí - Campus Picos, 2023.

NUXTJS. **The Intuitive Vue Framework.** Disponível em: <<https://nuxt.com/>>. Acesso em 30/01/2024.

OLIVEIRA, A. S., MACEDO, L. V., PIRES, D. F. (2021). **Student Housing: Aplicação Para Aluguel De Quartos E Imóveis Para Estudantes.** Reca, volume(2), p. 2.

Os desafios de mudar de cidade para estudar, Blog Mackenzie. Disponível em: <<https://blog.mackenzie.br/vestibular/atualidades/os-desafios-de-mudar-de-cidade-para-estudar/>>. Acesso em 18/06/2023.

TYPESCRIPT. **JavaScript that scales.** Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 24/11/2023.

PAGESPEED INSIGHTS. **Aumente a velocidade das suas páginas da Web em todos os dispositivos.** Disponível em: <<https://pagespeed.web.dev/>>. Acesso em: 15/02/2024.

PINIA. **Pinia The intuitive store for Vue.js.** Disponível em: <<https://pinia.vuejs.org/>>. Acesso em 30/01/2024.

POSTGRESQL. **PostgreSQL: The World's Most Advanced Open Source Relational Database.** Disponível em: <<https://www.postgresql.org/>> . Acesso em: 30/01/2024.

PRISMA. **Next-generation Node.js and TypeScript ORM.** Disponível em: <<https://www.prisma.io/>>. Acesso em: 30/01/2024.

VUEJS. **The Progressive JavaScript Framework.** Disponível em: <<https://vuejs.org/>>. Acesso em: 30/01/2024.