Sistema para auxiliar na mudança de imóveis

Cesar H. Sousa[1], **Keila D. Fernandes**[1], **Leonardo B. Estacio**[1]

[1] Instituto Federal de Santa Catarina (IFSC) - Câmpus Lages
Curso Superior de Ciência da Computação
R. Heitor Villa Lobos, 222 - São Francisco, Lages - SC, 88506-400

{cesar.hs,keila.df}@aluno.ifsc.edu.br, leonardo.bravo@ifsc.edu.br

Abstract. *When searching for a property, the user looks for a place where he will feel as satisfied as possible, with this, the lack of information when searching for properties on the Internet can end up leaving users unsure about the choice of their home, as well , to ensure security in the negotiation this application centralized all the information about the property and its region. The construction of this application was subdivided into research and study of the bibliography, of the tools used so that its development could be carried out, which was based on Node.js for the back-end and Vue.js on the front-end. As a result of this work, we obtained a system developed and ready for use, which was modeled after the MongoDB document structure. The results were satisfactory as to the construction of this application, as it has all the necessary functions for a good user experience.*

Resumo. *Na pesquisa por um imóvel, o usuário procura um lugar em que irá sentir-se o mais satisfeito possível, com isso, a falta de informações ao se procurar por imóveis na Internet pode acabar deixando os usuários inseguros sobre a escolha de sua moradia, assim, para garantir a segurança na negociação esta aplicação centralizou todas as informações sobre o imóvel e sua região. A construção desta aplicação foi dividida em pesquisa e estudo da bibliografia, das ferramentas utilizadas para que então fosse realizado seu desenvolvimento, que tinha como base o Node.js para o back-end e o Vue.js no front-end. Como resultado deste trabalho obtivemos um sistema desenvolvido e pronto para o uso, sendo este modelado a partir da estrutura de documentos do MongoDB. Os resultados se mostraram satisfatórios quanto a construção desta aplicação, pois ele possui todas as funções necessárias para uma boa experiência de usuário.*

1. Introdução

A migração humana é um fenômeno antropológico muito antigo em que as pessoas se deslocam para novas regiões. Pode ser desencadeado por diversos fatores: em consequência de desastres ambientais, guerras, perseguições políticas, étnicas ou culturais, causas relacionadas a estudos em busca de trabalho e melhores condições de vida, entre outros (Marinucci and Milesi, 2005).

Para um indivíduo realizar tal mudança, há toda uma pesquisa relacionada ao que mais lhe atrai em uma região, desde a diferença de salários, a facilidade de obtenção de um emprego, questões relacionadas a educação, o custo da moradia e até mesmo ao lazer. De uma forma

geral, a migração se torna uma excessiva busca por informações descentralizadas (Yuan et al., 2013).

O avanço tecnológico e o desenvolvimento de Sistemas de Computacionais foram os grandes responsáveis por muitos avanços nesse processo, pois oferecem uma vasta gama de informações sobre diversos lugares e culturas, também proporcionam uma forma de comunicação entre pessoas e/ou empresas distantes. A tecnologia já auxilia em vários campos relacionados à migração, conforme Tamburu (2018) isso vai desde pesquisas sobre a qualidade de vida em determinadas regiões à possibilidade de conseguir um emprego em uma lugar mais afastado, porém mais propício para viver.

De acordo com Klein et al. (2012) "o acesso às informações, de maneira rápida e praticamente sem custo, possibilita aos migrantes potenciais terem mais consciência do mundo à sua volta. Assim, a agilidade e a rapidez dessas informações ao alcance da população fazem com que o indivíduo conheça melhor as oportunidades disponíveis nas mais diversas regiões do mundo."

Pichek (2009) discute que a partir do início do século 20 começou a surgir no mercado aplicações que buscam reunir informações sobre os imóveis disponíveis para locação onde o locatário pode encontrar algo que seja de seu gosto com maior facilidade. Contudo, encontram-se na maioria das vezes informações sobre o imóvel, enquanto que dados relacionados a região em questão acabam ficando de lado e se tornando escassas.

Essa falta de conhecimento sobre uma região pode acabar resultando em uma visão simplificada sobre o imóvel e não ao que há na região ao seu redor, isto pode influenciar na adaptação e/ou satisfação do locatário. Com isso, existe essa demanda por uma aplicação que consiga centralizar todo o conhecimento da propriedade dispondo de uma macro visão sobre a mesma.

Para tentar suprir tal demanda, foi proposto o desenvolvimento de um sistema em que o locador possa registrar informações que ele julga ser interessante aos locatários, além disso o locatário pode visualizar dados sobre a região a qual ele não havia pensando em procurar. Essa busca também tentar evitar futuras insatisfações por parte do locatário, já que a região foi previamente introduzida a ele e assim diminui as chances de futuras surpresas. Assim, este projeto se propõe a facilitar a migração de pessoas para uma região que fica previamente conhecida pelo locatário, antes mesmo da mudança, garantindo uma maior segurança e confiabilidade ao realizar a negociação.

O projeto tem como objetivo desenvolver um sistema de locação de imóveis, onde os locadores podem cadastrar as informações relevantes sobre seus imóveis e aqueles que desejam alugar possam visualizar. Para atingir esse objetivo principal, foram estabelecidos os seguintes objetivos específicos:

- Estudar como ocorre o processo de troca de moradia e identificar as características mais relevantes deste processo;
- Levantar e registrar os requisitos da aplicação;
- Projetar a camada de armazenamento de dados;
- Implementar o aplicativo usando tecnologias web.

Este artigo está dividido em seis seções. A Seção 2 (Materiais e Métodos) detalha os

caminhos tomados para a construção deste trabalho e como este se encontra estruturado. A Seção 3 (Referencial Teórico) aborda os temas mais relevantes para o entendimento do trabalho como um todo. A Seção 4 (Desenvolvimento) conta com maior detalhamento a aplicação dos materiais e métodos discutidos na Seção 2. A Seção 5 (Resultados e Discussões) irá discutir os resultados obtidos no desenvolvimento. Por fim, tem-se a Seção 6 (Conclusões) que irá apresentar as considerações finais e discutir possíveis trabalhos futuros.

2. Materiais e Métodos

A pesquisa realizada para o desenvolvimento deste trabalho possui caráter aplicado, por apresentar um problema identificado no âmbito da sociedade, tendo como objetivo a resolução deste através da sua alteração, melhoria ou criação. Marconi et al. (2002) debate que a pesquisa aplicada possui o interesse prático no qual busca solucionar problemas da realidade, sendo assim, os resultados utilizados e aplicados no problema em questão.

Este artigo apresenta como procedimento técnico, a pesquisa bibliográfica e documental, na qual Prodanov and de Freitas (2013) cita que a pesquisa bibliográfica procura reunir dados e informações sobre o estudo em questão, através da análise de materiais relacionados ao tema e a pesquisa documental utiliza de fontes secundárias para a realização do presente estudo.

A forma de abordagem do problema se dá através de uma pesquisa qualitativa, onde o estudo é realizado sobre informações adquiridas do ambiente. "Os dados coletados nessas pesquisas são descritivos, retratando o maior número possível de elementos existentes na realidade estudada" (Prodanov and de Freitas, 2013).

Para a elaboração deste trabalho, foi realizada uma segmentação, conforme Figura 1, onde o primeiro ramo deste segmento se deu com a pesquisa e estudo bibliográfico, assim ela detalha como ocorreu a pesquisa bibliográfica e documental para se criar tanto o artigo, quanto o sistema.

O artigo teve um estudo mais detalhado através do uso da revisão sistemática, sendo esta melhor explicada na Seção 2.1.1. No segundo ramo da Figura 1 é possível visualizar como ocorreu o estudo teórico das ferramentas de desenvolvimento e os caminhos tomados durante esta etapa, podendo ser melhor compreendida na Seção 2.2.

Por último, se tem o segmento de desenvolvimento do sistema, encontrado na Seção 2.3 em que será abordado todo o ciclo de desenvolvimento e as escolhas ao se desenvolver o software.

2.1. Pesquisa e Estudo Bibliográfico

Esta etapa foi realizada inicialmente em março de 2020, sendo feita até setembro de 2020, durante este período houve uma leitura crítica dos artigos, através de uma metodologia chamada revisão sistemática.

As ferramentas de busca foram o Google Acadêmico e o portal de periódico Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), com foco em estudos do período de 2005 até 2020, para que as informações encontradas pudessem ser o mais atualizadas possíveis.

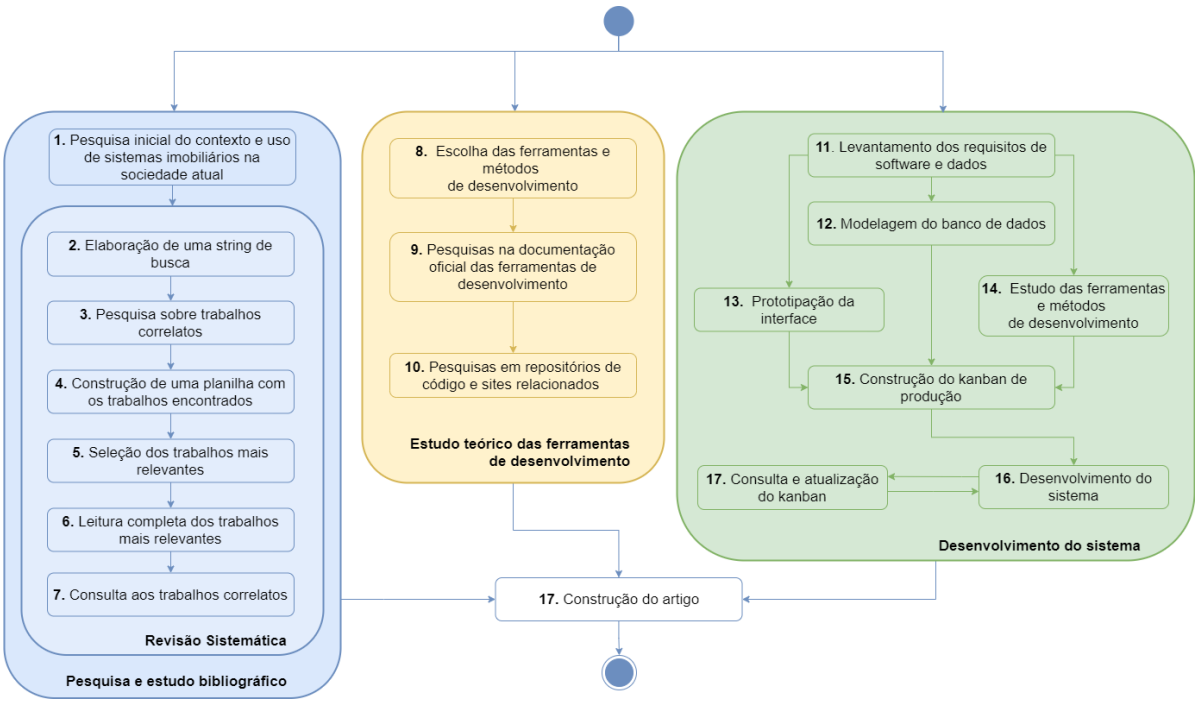

Figura 1. Fluxograma Geral da pesquisa e desenvolvimento do sistema

Como é possível observar na Figura 1, a pesquisa bibliográfica foi a primeira ramificação feita no Fluxograma Geral da pesquisa e desenvolvimento do sistema, ela ainda fora subdividida em 7 etapas, onde é possível observar uma sequência lógica e temporal do processo de pesquisa e desenvolvimento.

Na etapa 1 realizou-se uma pesquisa sobre a importância do desenvolvimento do presente sistema e qual sua contribuição para o mercado imobiliário, sendo também analisado os sistemas já existentes e qual a importância dos mesmos na sociedade atual, para que se tivesse um paradigma em que o sistema aqui relatado pudesse contribuir com novas funcionalidades. Da etapa 2 à etapa 7 é considerado uma metodologia chamada revisão sistemática que é tratada na Seção 2.1.1.

2.1.1. Revisão Sistemática

De acordo com Kitchenham (2004) o método da revisão sistemática tem em vista identificar, avaliar e interpretar as pesquisas disponíveis relacionadas a um tema específico. Para a realização deste método, foi preciso elaborar uma planilha como mostrado na Figura 2, assim utilizou-se o Google Docs para o compartilhamento com a equipe, para que fosse realizado a etapa 2 à etapa 7.

Na etapa 2 se tem a elaboração de uma *string* de busca, nela se criou uma cadeia de caracteres que facilitam para que as ferramentas de busca possam encontrar artigos, livros e periódicos de forma mais eficiente, dessa forma, foi importante se construir uma *string* com



Figura 2. Planilha da revisão sistemática feita no Google Docs

caracteres relacionados ao tema do artigo. A equipe utilizou termos tanto em inglês como em português, algumas dessas palavras chaves foram "aluguel de imóveis, imobiliárias, locação e aplicação imobiliária".

Seguindo ainda esta ramificação da Figura 1, durante a etapa 3 houve a utilização da *string* criada no passo anterior, para que se realizasse a pesquisa a trabalhos correlatos, utilizando as ferramentas de busca do Google Acadêmico e do portal de periódico CAPES.

Para a realização da etapa 4 reuniu-se os materiais encontrados na etapa 3 e foi criada uma tabela organizada conforme a Figura 2. Esta tabela possui 7 colunas, com cada coluna contendo respectivamente título, autores, ano publicado, um comentário relevante, o resumo do artefato, nota e forma de citá-la.

Após uma leitura dos conteúdos reunidos na tabela, passou-se para a etapa 5, nela houve o preenchimento da coluna referente a nota de cada material encontrado,levando em consideração sua importância para o presente trabalho, a nota variava entre 1 e 5, sendo 1 com uma importância pequena e 5 com muita importância, durante a procura de artigos foram selecionados 30 trabalhos que combinavam com o tema,

Com os trabalhos mais relevantes encontrados, houve então na etapa 6 uma leitura na íntegra dos artigos classificados com uma nota mais importante, estes trabalhos foram lidos na íntegra e se tornaram uma fonte constante para consulta de referências.

Por fim, na etapa 7 se deu a consulta nos trabalhos encontrados, para que estes servissem como suporte para o desenvolvimento deste artigo.

2.2. Estudo Teórico das Ferramentas de Desenvolvimento

O estudo teórico das ferramentas de desenvolvimento foi realizado conforme o segundo segmento da Figura 1. Esta etapa foi iniciada em março de 2020 e se estendeu à fevereiro de 2021, porém, tendo fim apenas ao final do desenvolvimento do sistema.

O primeiro item desta etapa se concentrou na escolha das ferramentas e o método de desenvolvimento a ser adotado, houve então uma pesquisa tanto em trabalhos relacionados ao tema, quanto em documentações das ferramentas que já eram conhecidas pelos desenvolvedo-

res, assim, efetuou-se a criação de tabelas no Google Drive para a comparação das ferramentas e quais atendiam o esperado para este projeto.

Para a escolha do método de desenvolvimento pesquisou-se qual seria a metodologia de mais fácil adaptação conforme o tempo, o tamanho da equipe e os conhecimentos dos desenvolvedores.

Após a escolha de quais seriam as ferramentas e os métodos a serem utilizados, passou-se para a segunda etapa, nela houve uma pesquisa mais profunda na documentação das ferramentas de desenvolvimento para que guiasse os desenvolvedores durante a elaboração do código. O estudo da documentação é importante para a construção de um sistema, pois de acordo com Fernandes (2012) estas consultas se dão de forma a estudar as funcionalidades e até mesmo as limitações que uma ferramenta pode oferecer.

Como nem todos os problemas que acontecem em um código estão apresentados na documentação de uma ferramenta, houve uma etapa seguinte a de pesquisa na documentação, a pesquisa em repositórios de códigos já existentes e em sites relacionados a ferramenta. A pesquisa em repositórios se deu principalmente em plataformas de hospedagem de código fonte gratuitos.

2.3. Metodologia de Desenvolvimento do Sistema

Após o processo de pesquisa na bibliografia e um estudo inicial das ferramentas utilizadas, foi iniciado o desenvolvimento do sistema conforme a Figura 1. Essa esta etapa foi dividida em sete etapas.

A primeira etapa presente no desenvolvimento é o levantamento dos requisitos de software e dados iniciais da aplicação, foi então criado um documento no Google Drive para que todos os autores pudessem contribuir. Nele foi dividido os requisitos em dois tipos, requisitos para o usuário e requisitos para o imóvel.

Um requisito tem como função definir os critérios para a aceitação de um produto, assim, conforme de Pádua Paula Filho (2003) "O valor de um produto vem de suas características". Este ainda afirma que em engenharia de software um sistema pode ter suas características divididas em **características funcionais** e **características não-funcionais**. A primeira refere-se ao modo como o sistema irá responder a ações tomadas pelo usuário no sistema, enquanto que a característica traz a forma como a resposta pode ser apresentada ao usuário.

Para cada requisito definido pela equipe, foi criado uma tabela seguindo o modelo da Figura 3, informando os Requisitos Funcionais e Requisitos Não-Funcionais do sistema.

De acordo com Vazquez and Simões (2016), nos Requisitos Funcionais têm-se a: descrição; fonte de informação; usuários, a fim de que ficasse explícito qual usuário iria usufruir de tal funcionalidade; informações de entrada, para que ajudasse tanto na elaboração do modelo do banco de dados, quanto para o modelo das interfaces gráficas; e informações de saída, tendo de forma clara uma resposta para o usuário. Para os Requisitos Não-Funcionais têm-se apenas a identificação do requisito; descrição; e sua categoria. conforme é visto na Figura 3.

RF1. Nome do requisito		
Nome	Descrição	
Descrição		
Fonte de Informação		
Usuários		
Informação de Entrada		
Informação de Saída		
Requisitos Não-Funcionais		
Nome	Descrição	Categoria
RNF 1.1		

Figura 3. Tabela modelo para os requisitos do sistema

Para a modelagem do banco de dados houve a consulta ao documento de requisitos, aos artigos pesquisados anteriormente na Subseção 2.1 (Pesquisa e estudo bibliográfico), consultas a sites de locação de imóveis citados também na Subseção 3.3 (Trabalhos Correlatos) e consultas ao protótipo das telas do sistema. Criou-se então um documento de desenho no Google Drive para que fosse modelado o banco de dados de forma a atender as necessidades que foram aparecendo. A modelagem seguiu de forma a reunir os dados e agrupá-los de acordo com as suas similaridades.

A prototipação da interface foi realizada em um *software* de forma compartilhada com os desenvolvedores e teve como base os dados reunidos na modelagem do banco de dados e os sites de locação de imóveis citados na Subseção 3.3 (Trabalhos Correlatos).

Após a produção dos requisitos, modelagem do banco de dados e a prototipação das interfaces gráficas, passou-se então para a elaboração do Kanban, já que de acordo com de Oliveira (2014), o primeiro passo para aplicar o Kanban com sucesso é entender seu processo atual, visualizar o fluxo de trabalho e mapear o fluxo/cadeia de valor.

Com base nas tarefas colocadas no Kanban, foi possível iniciar o desenvolvimento do sistema seguindo esse fluxo. A cada funcionalidade terminada no desenvolvimento, se atualizava o Kanban de produção e consultava novas funcionalidades a serem desenvolvidas, assim mantendo um progresso visível na etapa de desenvolvimento do sistema.

3. Referencial Teórico

Esta seção foi dividida em três subseções onde a Subseção 3.1 irá dar uma visão sobre o mercado imobiliário e os motivos que impulsionaram a expansão deste setor a partir dos anos 2000 para que se tenha um entendimento sobre a relevância dele para a execução do presente trabalho, assim como na Subseção 3.2 onde será abordado os sistemas imobiliários baseados

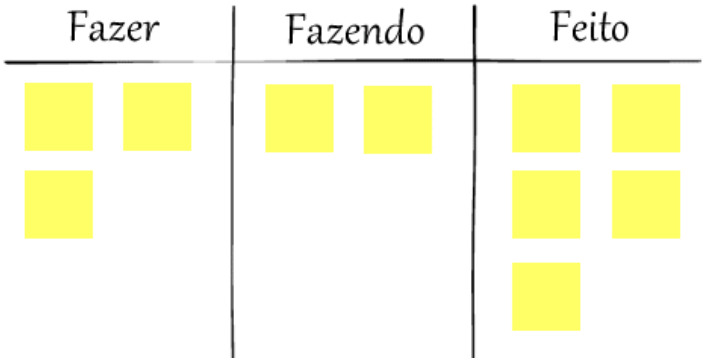

Figura 4. Modelo do Fluxo de trabalho no Kanban

na web e como esse meio de comunicação auxilia na procura por imóveis, a Subseção 3.5 apresenta a escolha das ferramentas utilizadas. Por fim na Subseção 3.3 será citado trabalhos relacionados que ajudaram a desenvolver o sistema atual de forma mais eficiente.

3.1. Mercado Imobiliário

Para se compreender o mercado imobiliário é preciso entender como ele é percebido e composto, assim, o mercado é percebido como um conjunto de compradores e vendedores dispostos a realizar trocas de bens de serviços por uma unidade monetária ou por outros bens, sendo também composto por outros agentes como Matos and Bartkiw (2013) menciona: imobiliárias até empresas prestadoras de serviços em propaganda e marketing, que atuam nas atividades de administração e comercialização dos empreendimentos imobiliários .

Sanfelici (2013) afirma que o mercado imobiliário vem sofrendo uma forte expansão desde os anos 2000, onde é possível observar o crescente aumento das metrópoles, tornando-se assim uma característica marcante das metrópoles. Essa expansão se deve também ao déficit de moradias, que segundo uma pesquisa feita pela Associação Brasileira de Incorporadoras Imobiliárias (Abrainc) em parceria com a Fundação Getulio Vargas (FGV) mostra que o déficit de moradias cresceu 7% em apenas dez anos, de 2007 a 2017, tendo atingido 7,78 milhões de unidades habitacionais em 2017 (Gavras, 2019).

Um tópico que também contribuiu com a expansão do mercado imobiliário é a mudança no perfil das famílias. Segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE, 2016), o tamanho da família brasileira diminuiu em todas as regiões: de 4,3 pessoas por família em 1981, chegou a 3,3 pessoas em 2001. O número médio de filhos por família no início de 2019 é de 1,6 filhos. Essa diminuição da quantidade de filhos causa um impacto onde naturalmente as casas diminuem. Assim, “Para o futuro, alguns especialistas descrevem que as famílias tenderão a ser menores ainda. Há uma tendência e um mercado a ser desenvolvido para solteiros” (Matos and Bartkiw, 2013).

O aumento e facilitação de linhas de financiamento possibilitou também esse crescimento do mercado imobiliário, principalmente após uma iniciativa do Governo Federal com a criação do programa minha casa, minha vida (PMCMV) em 2009, que de acordo com Ministério do Desenvolvimento Regional (MDR) “é a maior iniciativa de acesso à casa própria já

criada no Brasil", onde é oferecido condições especiais de financiamento, através de parcerias com estados federativos, municípios, empresas e entidades sem fins lucrativos.

Tendo em conta os vários motivos apresentados que impulsionaram o mercado imobiliário, é possível perceber que este setor começou a investir em plataformas para a web, pois, "a popularização da Internet permitiu reunir pessoas com interesses comuns, independentemente da sua localização geográfica, e agrupá-las em comunidades virtuais, transferindo a seus membros um grande poder de negociação" (da Cunha Maya and Otero, 2002).

3.2. Sistemas Imobiliários Baseados na Web

Conforme da Cunha Maya and Otero (2002) afirma, o aumento de investimentos em novos portais, sites e conteúdo tem sido realmente significativo, e muitas empresas que ainda não entraram no mundo *on-line* já estão definindo estratégias para usufruir das vantagens do negócio eletrônico.

O *e-commerce* é uma abreviação de *eletronic commerce*, traduzindo comércio eletrônico. Ele se fortaleceu com a chegada da Internet facilitando assim todo o processo de compra e venda, no início apenas pequenos produtos eram comercializados como cd's, dvd's, livros, etc. Porém, "começou-se também a comercializar carros, casas, iates, aviões, obras de arte e qualquer outro tipo de produto de luxo" (de Mendonça, 2016).

Já o *Marketplace* é um modelo de negócio que serve para intermediar a compra e a venda de outras pessoas, ele centraliza os produtos de diferentes vendedores e facilita a procura dos possíveis clientes, dentro dele há diversos modelos de transações como o Cosumer to Cosumer (C2C), que é utilizado no presente trabalho, alguns outros tipos serão listados abaixo.

3.2.1. Tipos de *Marketplace*

Os tipos de vendas foram se especializando ao ponto de terem mecânicas e modelos de negócios específico. E com isto, várias siglas foram sendo criadas para denominar cada um destes modelos (Bonifacio, 2016). Há diversas categorias de marketplace, tendo quatro como principais. O primeiro deles é denominado de Business to Business (B2B). É um sistema de vendas entre duas empresas, na maior parte das vezes serve como compra para a reposição de estoques e/ou revenda.

O segundo é o Business to Consumer (B2C), que ao contrário do B2B as empresas têm como público alvo o cliente final, como por exemplo, um comércio estilo varejo.

O terceiro se chama Consumer to Consumer (C2C). É uma plataforma onde pessoas anunciam seus produtos para outras pessoas, como, sites de leilão, OLX, Mercado Livre entre outras, onde mesmo tendo empresas vendendo o foco é nas vendas C2C.

Por fim existe o Marketplace de Nicho, que se trata de empresas que atendem um público específico, focando em um nicho, pois mesmo com menos variedade de produtos é possível vender mais para os clientes.

3.3. Trabalhos Correlatos

A ideia de alugar e vender imóveis é antiga, por isso há trabalhos de diferentes épocas, o tema relacionado com a tecnologia começou a ficar mais frequente após os anos 2000. Buscando validar nossa ideia a equipe percebeu ser importante utilizá-los como base, para que assim se possa analisar e comparar os resultados aos existentes. Com isso, foram selecionados cinco sistemas relacionados ao trabalho realizado nesta pesquisa, outros sistemas existentes foram descartados devido à similaridade com os escolhidos.

3.3.1. ZAP

O ZAP imóveis começou inicialmente em 2001 com o nome Planeta Imóvel, sendo a princípio comprado pelos grupos O Globo (Infoglobo) e O Estado de São Paulo (OESP) em 2001, mas logo ficando sob o comando apenas do Infoglobo em 2012, desse momento, começou a se tornar um portal totalmente dedicado a imóveis, tendo várias outras marcas relacionadas a imóveis pertencentes ao Infoglobo, conforme (ZAP, 2018).

Analisando a plataforma *web* do ZAP, foi possível identificar 4 perfis de usuários que poderiam cadastrar imóveis para venda ou aluguel. Esses 4 perfis são: corretor; imobiliária; incorporadora; usuário comum. Dentre estes apenas o último foi possível testar as funcionalidades disponíveis, pois para os demais usuários era necessário um cadastro em que o próprio ZAP entra em contato para confirmar as credenciais.

Ao cadastrar o imóvel com o perfil de usuário comum, as informações solicitadas sobre o imóvel são: Localização do imóvel, Custo do imóvel, outras despesas (se não isentas) e dados considerados importantes sobre o imóvel como a área útil e total, além da quantidade de banheiros, quartos e outros cômodos opcionais, ao final é gerado uma descrição do que foi marcado presente no imóvel.

Do ponto de vista do presente trabalho, as informações que o ZAP exige podem acabar deixando a desejar, por não abranger maiores características sobre o imóvel, se limitando apenas ao usuário deixar explícito ao mudar na descrição, o que há a mais no imóvel.

Por vez, o ZAP possui outras características consideradas importantes para a procura de um imóvel, uma delas é poder conferir a distância dos imóveis até um ponto de referência escolhido previamente no campo de pesquisa, o que torna importante esta funcionalidade de acordo com uma pesquisa feita por Lepkova et al. (2016) que estudava a satisfação do cliente com condições de vida em novos edifícios e apartamentos, é a questão do usuário se concentrar em conhecer a localização de sua futura casa com antecedência.

3.3.2. Kijiji

Lançado em 2005, conhecido também como Classificados eBay, o Kijiji é uma marca pertencente ao eBay e conforme o próprio (Kijiji, 2011) é o maior site de classificados do Canadá com milhões de anúncios por dia, tem como objetivo proporcionar aos canadenses uma maneira fácil dos usuários realizarem negociações de mercadorias, carros, casas e empregos.

Por se tratar de um sistema de classificados, onde o negócio não é focado apenas em imóveis, suas funcionalidades citadas aqui, serão apenas referentes as negociações de imóveis. Assim, nota-se a princípio que o Kijiji possui um sistema de busca detalhado e uma apresentação dos imóveis de forma precisa, mostrando uma visão do que possui ou não no imóvel, desde estacionamento à aceitação de animais de estimação, passando por informações sobre a mobília(caso mobiliada), até questões sobre o terreno e/ou edifício.

Apesar da sua popularidade e robusta fonte de informações sobre os imóveis, o Kijiji está presente apenas no Canadá e na Itália. Deste modo, ele acaba se tornando inacessível para alguns lugares. Outro item que poderia ser melhor explorado na visualização do imóvel, são as informações da região em volta do mesmo, já que é apenas mostrado a localização do imóvel no mapa que o *Google Maps* disponibiliza, podendo ter uma maior participação do proprietário do imóvel no cadastrá-lo, colocando informações sobre segurança, pontos de ônibus e outros dados que podem ser considerados relevantes.

3.3.3. OLX

Presente em 45 países, chegou ao Brasil em 2010 sob o comando de dois grupos, o sul-africano Naspers e o grupo norueguês Schibsted. Até agosto de 2019, possuía uma média de 2 milhões de vendas por mês (cerca de 50 vendas por minuto) apenas no Brasil (OLX, 2019).

Como o Kijiji, o OLX também se denomina como um sistema de classificados, onde um consumidor pode negociar com outro consumidor através do sistema, porém, no caso do OLX, a negociação é intermediada pelo sistema, sendo considerado um *marketplace* do tipo C2C.

Sendo considerado um tipo de *marketplace* C2C, o OLX não possui no seu cadastro de anúncio qualquer restrição quanto ao tipo de usuário específico que pode realizar esta ação, assim, sendo livre para qualquer pessoa realizar o cadastro. No cadastro de anúncio está presente informações consideradas básicas como título, descrição e categoria, o OLX não trata apenas de anúncios de imóveis, logo a análise do mesmo neste trabalho será apenas na parte imobiliária, concentrando-se apenas categoria de imóveis.

A categoria de imóveis possui mais algumas informações dependendo do tipo de imóvel escolhido, os tipos são: apartamentos; casas; aluguel de quartos; temporada; terrenos, sítios e fazendas; comércio e indústria. Somente os 3 primeiros também estão sendo tratados neste trabalho, pois se direcionam a pessoas que querem realizar uma mudança de moradia. Analizando-os percebe que não se tem um detalhamento tão grande das características dos imóveis e principalmente da localização que apenas solicita o endereço do imóvel. Na visualização de um imóvel, há também a ausência de um mapa para visualizar em que lugar da região a propriedade fica localizada.

3.3.4. Booking

Vindo dos países baixos, mais precisamente de Amsterdã, foi fundada em 1996 através de uma *start-up*. De uma pequena *start-up* fundada em 1996 em Amsterdã, nos Países Bai-

xos, a Booking se transformou em uma das maiores empresas de *e-commerce* de viagens do mundo (Booking.com™, 2016). A Booking é direcionado para pessoas que querem viajar e querem um lugar para permanecer durante o período da viagem.

A proposta da Booking e a deste trabalho se assemelham por conta da rica fonte de informações que os usuários encontram ao procurar um imóvel, o mesmo possui tanto informações da estrutura do imóvel, como informações das instalações e serviços prestados naquele imóvel, além de disponibilizar a quantidade de pessoas que um cômodo acomoda. Todas essas funcionalidades podem ser cadastradas por qualquer pessoa que queira anunciar seu imóvel, porém, a Booking cobra uma comissão por imóvel cadastrado.

Suas funcionalidades são realmente abundantes, mas são apenas para quem possui o objetivo de viajar e se acomodar por um curto período de tempo em um local, logo pessoas que querem mudar de moradia não se encaixam nesse público destinado ao Booking.

3.3.5. Imobille

A Immobile está localizada em Balneário Camboriú e está sob o comando da empresa Imobille Negócios Imobiliários, atua na sua sede e região da Praia Brava, em Itajaí. Possui desde imóveis comerciais a imóveis para moradia, além de contar com a parceria de mais de 10 construtoras da região.

Possui um tipo de *marketplace* B2C, já que possui uma empresa sob seu comando, assim o cadastro de um imóvel somente é feito por um corretor previamente cadastrado, com isso não foi possível avaliar seu sistema de cadastro de imóvel. Segundo Imobille (2021) "utilizamos um time de profissionais qualificados que se identificam com o perfil dos clientes, imóveis criteriosamente selecionados e uma plataforma virtual de alta performance, com informações claras e imagens de alta qualidade".

Teve-se então uma análise nas informações dos imóveis já cadastrados, neles havia informações sobre a opção de incluir mobília, mas não informava quais mobílias estavam inclusas, uma informação considerada importante e apresentada pelo Immobile é a questão da metragem do imóvel, tendo não só da área privativa, como da área total da propriedade.

É possível visualizar em uma opção ao lado das fotos, o mapa em que demarca a região que o imóvel se encontra, a vantagem dele é a de poder visualizar o que há ao redor do imóvel, mas a localização exata do imóvel não é mostrado no mapa, assim fazendo o usuário ter que procurar visualmente ou em outro site o imóvel através da localização fornecida de forma textual.

3.4. Comparação Entre os Trabalhos Correlatos

Tendo agora a noção de como cada sistema cresceu e adentrou ao mercado *online*, foi possível elaborar a Tabela 1 com as principais características nos sistemas mostrados anteriormente.

As caixas marcadas com **x** marcam os sistemas em que aquela funcionalidade se encontra ausente, as marcadas com **-** representam a presença parcial de alguma daquelas funcionalidades citadas, enquanto que as caixas marcadas com **v** representam a presença de todas aquelas funcionalidades citadas.

Característica	ZAP	Kijiji	OLX	Booking	Imobille
Mostrar outros imóveis que estão próximos ao imóvel quando pesquisado no mapa.	x	x	x	v	x
Visualizar informações sobre o contrato, o que há ao redor do imóvel e quais mobílias inclui nele.	-	v	-	-	-
Contatar diretamente o anunciante por e-mail ou telefone.	v	v	v	x	v
Funcionalidade de cadastro de imóvel livre para qualquer usuário realizar.	v	v	v	v	x
Mostra atrações, pontos turísticos e comércios presentes ao redor do imóvel.	-	-	x	-	-

Tabela 1. Comparação entre os sistemas de alugar e vender imóveis.

A primeira funcionalidade analisada na tabela, especifica se o usuário pode visualizar no mapa outros pontos em que também se tem imóveis para alugar ou vender, além do imóvel que ele está visualizando. Esta funcionalidade serviria de apoio visual, caso o usuário quisesse encontrar outros imóveis em uma região em que lhe chamou atenção.

Visualizar informações sobre o contrato, o que há ao redor do imóvel e quais mobílias inclui nele é a principal funcionalidade analisada neste trabalho, uma vez que esta funcionalidade será um dos principais diferenciais deste trabalho para outros já existentes. Dos cinco sistemas analisados, apenas um possuía todas as características citadas desta funcionalidade, porém, de acordo com o próprio este não está presente no Brasil.

A quarta funcionalidade analisada, trata-se da possibilidade do comprador contatar o anunciante do imóvel diretamente por telefone ou e-mail, ela se torna importante para a fácil comunicação e troca de dúvidas entre os usuários, assim, não dependendo da espera da realização do acesso na plataforma por parte de ambos os usuários para visualizar as informações sobre seus e/ou imóveis em que esteja interessado.

Como visto nos trabalhos correlatos é importante que todo e qualquer usuário possa ser capaz de cadastrar um imóvel sem a necessidade da utilização de algum corretor ou empresa, deste modo a quinta funcionalidade explorada nos sites relacionados é a possibilidade de qualquer usuário cadastrar um imóvel, como resultado todos exceto o Imobille haviam esta opção presente em suas plataformas.

3.5. Tecnologias para Desenvolvimento Web

Nesta subseção será apresentado uma breve explicação sobre as ferramentas utilizadas para o desenvolvimento do presente sistema, também apresentado na Seção 4. Todas as tecnologias apresentadas possuem uma licença gratuita ou de estudante para sua utilização.

3.5.1. MongoDB

Um banco de dados pode ser definido como uma "estrutura computacional compartilhada e integrada que armazena um conjunto dados do usuário e metadados" (Rob and Coronel, 2011). Nele é possível realizar diversas operações, entre elas é possível destacar a possibilidade de acrescentar novos arquivos, acrescentar novos dados em arquivos já existentes, modificar, excluir e buscar os dados já presentes. Dependendo do tipo de sistema a ser implementado, deve-se escolher o tipo de banco de dados a ser utilizado.

O MongoDB é um banco NoSQL (*Not Only SQL*), seu código é aberto e o sistema possui uma versão disponibilizada de forma gratuita, é do tipo *document database* (banco de dados de documentos), estes documentos são no formato BSON (JSON (*JavaScript Object Notation*) Binário), a "ideia é o documento representar toda a informação necessária, sem a restrição dos bancos relacionais" (Boaglio, 2015).

Conforme Lóscio et al. (2011), O modelo de dados do MongoDB é dividido da seguinte forma: o banco de dados armazena um conjunto de coleções; esta coleção armazena um conjunto de documentos; o documento possui um conjunto de campos; onde cada campo possui um par chave-valor; a chave por sua vez é uma *string*; e o valor pode ser de um formato escolhido, como string, inteiro, documento, array e outros formatos disponíveis na documentação.

Um assunto importante debatido para a escolha de um banco de dados e que o MongoDB está presente, é o conhecimento do teorema CAP (*Consistency, Availability, Partition-resilience*), criado em 1999 por Eric Brewer, de acondo com Brewer (2012) qualquer sistema de dados compartilhados em rede pode ter apenas duas das três propriedades desejáveis. Essas propriedades são:

- *Consistency* (consistência)- Uma vez um estado é gravado no banco, cada leitura posterior recebe a gravação mais recente ou um erro.
- *Availability* (disponibilidade) - Mesmo quando há uma falha de um ou vários nós, o sistema continua funcionando, mesmo de forma inconsistente.
- *Partition-resilience* (resistência a fragmentação) - o sistema continua a operar apesar da rede interromper (ou atrasar) um número arbitrário de mensagens entre os nós.

Assim, existe uma "troca entre as propriedades CAP, onde CA (garante consistência e disponibilidade), CP (garante consistência e resistência a fragmentação) e AP (garante disponibilidade e resistência a fragmentação)" (Sampaio, 2019). O MongoDB está presente com a consistência e a sua resistência a fragmentação.

Um serviço disponibilizado pelo MongoDB que é importante ressaltar, onde segundo o MongoDB (2020) facilita a criação de bancos de dados para equipes que preferem gas-

tar tempo focando na criação da aplicação do que gerenciando bancos de dados, é o MongoDB Atlas, nele é possível ter mais produtividade com seus drivers, integrações e ferramentas nativas para manipular, visualizar e analisar os dados, além de ser altamente disponível, com tolerância a falhas distribuídas e contar com opções de backup para atender objetivos de recuperação de dados.

3.5.2. Javascript

"A linguagem de programação Javascript surgiu em 1995, criada por Brendan Eich na Netscape, empresa que criou o navegador homônimo" (DE BRITO, 2013, p. 24). E de acordo com Flanagan (2006) é uma linguagem de programação funcional interpretada capaz de reproduzir os princípios da programação orientada a objetos. Segundo a rede de Desenvolvedores da Mozilla (MDN), é utilizada principalmente no lado do cliente para *scripts* de páginas web dinâmicas, mas também é utilizado no lado do servidor.

Possui ainda bibliotecas padrões de objetos, como: Array, Date, Math e um conjunto de elementos que formam o núcleo da linguagem, tais como: operadores, estruturas de controle e declarações.

O Javascript está presente no *front-end*, carregando informações na página *web*, tanto estáticas, como dinâmicas e no *back-end* juntamente com o Node que é melhor detalhado na Seção 3.5.3, onde juntos conseguem criar um servidor *web*, com acesso á Interfaces de Programação de Aplicativos (API) nativas do sistema assim como outras linguagens de servidores o fazem.

3.5.3. Node.js

No final de 2009, Ryan Dahl origiu o Node.js com mais 14 colaboradores, conforme Pereira (2014), foi criado criado com o propósito de crias infraestruturas de redes escaláveis. O "Node.js é um ambiente de tempo de execução de código aberto, multiplataforma, JavaScript. Ele executa o código JavaScript fora de um navegador" (Foundation, 2019).

Darski (2016) afirma que é importante destacar que o Node.js não é uma nova linguagem ou uma plataforma como o Java, na verdade, trata-se de um aperfeiçoamento do Javascript que conseguiu inúmeros adeptos.

Conforme Pereira (2014), a adoção do Node.js é indicado para sistemas que produzem uma alta carga de processamento, já que sua arquitetura é totalmente *non-blocking thread* (não bloqueante), além de que, se a aplicação trabalhar com processamento de arquivos e/ou realiza muita entrada ou saída de dados, o Node.js será uma ótima escolha, pois, resulta em uma boa performance quando se trata de consumo de memória, pois, o mesmo usa ao máximo e de forma eficiente o poder de processamento dos servidores.

A MDN (*Mozilla Developer Network*) também enumerou vários benefícios do uso do Node.js no desenvolvimento de um servidor *web*, entre eles se destaca a existência do gerenciador de pacotes do Node (NPM - *Node Package Manager*) que fornece acesso a pacotes

reutilizáveis que a própria comunidade desenvolve.

3.5.4. Express

Criado por TJ Holowaychuk, onde o mesmo descreve o Express como uma estrutura da *web* inspirada no Sinatra que é uma DSL (*Domain Specific Language*) para se criar aplicações web de maneira rápida e com o mínimo de esforço em Ruby e no Connect, uma biblioteca de *"plugins"* para o Node (Brown, 2019) .

O Express é um framework para o desenvolvimento de aplicações do lado do servidor (*back-end*), mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos *web* e móvel (Express, 2017). Escrito em JavaScript e hospedado dentro do próprio ambiente de execução node.js, é possível utilizá-lo, atráves de um gerenciador de pacotes (Brown, 2019).

Conforme a MDN (2020), o Express fornece diversos mecanismos, entre eles a gerencia das requisições de diferentes requisições, rotas e URLs e a definição das configurações comuns da aplicação *web*, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta.

3.5.5. Figma

Criado em 2012, de acordo com o Figma (2020) é a primeira ferramenta *on-line* de nível profissional criada especificamente para o design de interface. O Figma é construído inteiramente no navegador, permitindo que todo o processo de design da equipe ocorra em uma ferramenta *online*, possui também uma versão *desktop* disponível para macOS e Windows.

Utilizado para os mais diferentes casos na área de *design*, se destaca principalmente como uma ferramenta de prototipação que segundo Unger and Chandler (2012) a prototipação pode ser definida no contexto de *UX(User Experience) Design* como o ato e a arte de criar uma, ou várias partes de um sistema, sendo este um processo interativo, e não um conjunto de entregáveis constantes.

Este ainda afirma que a prototipação pode ser trabalhada de duas formas: prototipagem no papel, onde os primeiros rascunhos das interfaces são feitas com papel e lápis, sendo uma forma econômica de criar e validar as ideias; e prototipagem digital, onde se tem o auxílio de um software para criar as interfaces, por ter este auxílio na criação, é comum que se tenha uma maior fidelidade na criação das telas.

O Figma se destaca de outras ferramentas de prototipação por: ser simples, pois junta várias funcionalidades com a utilização de apenas alguns *plugins* desenvolvidos pelo próprio Figma e pela comunidade, enquanto que outras ferramentas de prototipação podem necessitar que se conheça ferramentas além a de que se está desenvolvendo o protótipo; fácil compartilhamento de arquivos de design e colaboração em tempo real; dispensável o uso de aplicações, podendo utilizá-lo pelo navegador, pois utilizam de um poderoso mecanismo de renderização 2D WebGL que suporta documentos muito grandes, assim possibilitando um melhor desem-

penho mesmo para equipes maiores (Figma, 2020).

3.5.6. Vue.js

O Vue.js é um projeto de código aberto, desenvolvido inicialmente por Evan You em 2014 e mantido pela comunidade e apoiadores. Segundo o próprio, é uma estrutura progressiva para a construção de interfaces de usuário. Inicialmente foi desenvolvido para o propósito de ser adotado de forma incremental e pode ser facilmente escalável entre uma biblioteca e uma estrutura, dependendo dos diferentes casos de uso. Ela consiste em uma biblioteca principal acessível que se concentra apenas na camada de visualização e em um ecossistema de bibliotecas de suporte que ajuda a lidar com a complexidade em grandes aplicativos de página única (You, 2020).

Conforme o site do projeto do Vue.js, desenvolvido por You (2020), o mesmo se destaca por ser acessível, não sendo necessário conhecimentos além de HTML, CSS e Javascript; versátil, por ser simples e capaz de lidar com aplicações de pequeno e grande porte; e por sua alta performance, pois "sua biblioteca minificada pode chegar a um tamanho de 20kb quando comprimida com GZIP e tem mínima necessidade de esforço para otimizações" (Mendes et al., 2018).

3.5.7. Kanban

A palavra "Kanban" vem do japonês e significa Cartão visual, é um método de gestão de mudanças originado do Sistema Toyota de Produção (TPS), juntamente com o JIT (*just in time*) que focava apenas na demanda do cliente, sem fabricar peças fora da mesma. No processo de desenvolvimento de um software é considerada uma "ferramenta de controle visual aplicada para visualizar e otimizar o fluxo de trabalho" (de Oliveira, 2014).

De acordo com Baehr (2016), na área de desenvolvimento de software, técnicas Kanban começaram a ser utilizadas com o surgimento de métodos ágeis, no qual as equipes posicionam quadros em áreas visíveis da sala de projeto, identificam as possíveis fases da iteração, sendo elas: backlog da iteração, em desenvolvimento, em validação e finalizado, e preenchem cada fase com cartões que indicam os itens de trabalho selecionados para uma dada iteração, havendo assim um movimento contínuo no cartão (Prikladnicki et al., 2014).

Em concordância com Anderson (2010), é possível afirmar que o Kanban baseia-se em cinco princípios fundamentais, sendo eles:

Visualização do fluxo de trabalho pois assim se torna mais fácil identificar qual tarefa será importante realizar e qual terá um gasto desnecessário de mão de obra;

Trabalho em progresso limitado para que não se tenha um acumulo de tarefas para um tempo tão pequeno, assim um cartão somente é puxado para o próximo passo quando ele já foi concluído naquela fase;

Gerenciar o fluxo para que haja uma melhor gestão do trabalho, não focando somente em manter as pessoas ocupadas, mas em como gerenciar os processos para acelera-los;

Políticas de processos explícitos afim de que todas as pessoas entendam os processos definidos, assim deve-se tornar o processo bem definido para todos, pois quando as pessoas possuem um objetivo comum, ela são capazes de trabalhar e tomar decisões com relação a mudanças que o moverão em uma direção positiva;

Uso de modelos e métodos para reconhecer oportunidades de melhoria com o intuito que as pessoas possam ”manter um ciclo constante de melhorias, no qual a melhoria contínua seja responsabilidade de todos”(de Oliveira, 2014).

3.5.8. Trello

Trello é uma ferramenta baseada em nuvem que usa o método Kanban de gerenciamento de projetos. Sob o método Kanban, todas as atividades relacionadas ao projeto são exibidas em um único cenário que é visível para todos os membros da equipe do projeto. Com o Trello, os usuários podem organizar visualmente projetos em quadros, dividir projetos em grupos e subdividir grupos em tarefas. A interface amigável do Trello o torna ideal para uma ampla variedade de usuários, desde indivíduos que gerenciam projetos pessoais, como reformas de residências, até organizações que gerenciam vários projetos e equipes grandes. Para atender às necessidades dos usuários, o Trello oferece vários níveis de serviço com diferentes preços (Johnson, 2017).

Segundo Johnson (2017), o Trello possui três opções principais de preço: Grátis, *Business Class* e o *Enterprise*.

- *Grátis:* A versão básica permite que os usuários criem placas ilimitadas e se integrem ao Dropbox, Google Drive e Box. Ele contém três power-ups básicos (integração de calendário, votação e vencimento de cartões) e permite que os usuários anexem arquivos de até 10 megabytes (MB).
- *Business Class:* fornece visões gerais da equipe e oferece integração com vários aplicativos, incluindo, entre outros, Google Hangouts, Github, Slack e Mailchimp. Os usuários podem anexar arquivos com tamanho de até 250 MB.
- *Enterprise:* Enterprise é destinado a grandes organizações que gerenciam vários projetos e equipes. Os usuários corporativos desfrutam dos mesmos recursos que os da Business Class com camadas extras de segurança.

3.5.9. Visual Studio Code

Lançado em 29 de abril de 2015, o Visual Studio Code é um editor de código-fonte leve, mas poderoso, que roda na sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte interno para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens e tempos de execução (Microsoft, 2020), Além disso, é customizável, gratuito e de código aberto.

3.5.10. GitHub

Criado em outubro de 2007 com sua sede em San Francisco, pode ser definido como um serviço de hospedagem distribuída para armazenamento e colaboração de código que utiliza o Git para controle de versão (Github, 2019). De acordo com Chacon and Straub (2014) é considerado o maior *host* único para repositórios Git, possui uma forte colaboração de milhões de desenvolvedores que realizam o rastreamento de problemas, revisão dos códigos e ajuda à comunidade como um todo em projetos de código aberto.

3.5.11. Heroku

Lançado em junho de 2007, o Heroku é uma plataforma em nuvem que permite com que as empresas possam criar, entregar, monitorar e dimensionar aplicativos (Heroku, 2007), ele facilita o desenvolvimento, pois acaba com a necessidade de se preocupar com a infraestrutura e tem suporte a Node, Ruby, Java, Scala e PHP entre outras.

4. Desenvolvimento

A partir da metodologia apresentada na Seção 2.3 é possível compreender como foi organizado o processo de desenvolvimento do sistema, assim, nesta seção será elucidado como cada processo metodológico foi aplicado, detalhando a utilização das ferramentas e técnicas utilizadas.

4.1. Requisitos do Sistema

A Tabela 2 lista todos os requisitos funcionais levantados do sistema. Ao total foram 13 requisitos, sendo que cada requisito funcional possuía ainda os seus requisitos não-funcionais.

Requisitos do imóvel	
Nome do requisito	Prioridade
RF01. Adicionar imóvel para venda	alta
RF02. Adicionar imóvel para alugar	alta
RF03. Remover imóvel	média
RF04. Editar imóvel para venda	média
RF05. Editar imóvel para alugar	média
RF06. Visualizar imóvel para venda	média
RF07. Visualizar imóvel para alugar	média
RF08. Pesquisar imóvel	alta
Requisitos do usuário	
Nome do requisito	Prioridade
RF09. Login	alta
RF10. Cadastrar usuário	alta
RF11. Excluir usuário	alta
RF12. Editar usuário	média
RF13. Contatar proprietário do imóvel	média

Tabela 2. Requisitos funcionais do sistema

Do Requisito Funcional 01 ao 08 é possível visualizar as funcionalidades referentes ao imóvel, desde como adicionar um imóvel, até a opção de pesquisar por um imóvel cadastrado no banco de dados e do Requisito Funcional 09 ao 13 tem-se as funcionalidades referentes ao usuário.

A escolha de prioridade dos requisitos a serem implementados seguiu o julgamento dos autores. Esse julgamento baseou-se em desenvolver primeiro as partes que julgaram mais necessárias e que agregariam maior valor ao sistema desenvolvido.

4.2. Diagrama de Caso de Uso

A Figura 5 representa o cenário do sistema, ele tem como atores o locador e o locatário, estando eles associados aos seus respectivos casos de uso. O locador pode criar e gerenciar seus imóveis, assim como editar informações do imóvel ou definir se ele está locado. Ambos os atores podem realizar *login*, navegar pelo catálogo ou ver um imoveis em específico, além disso, o locatário pode ver as informações de contato do proprietário.

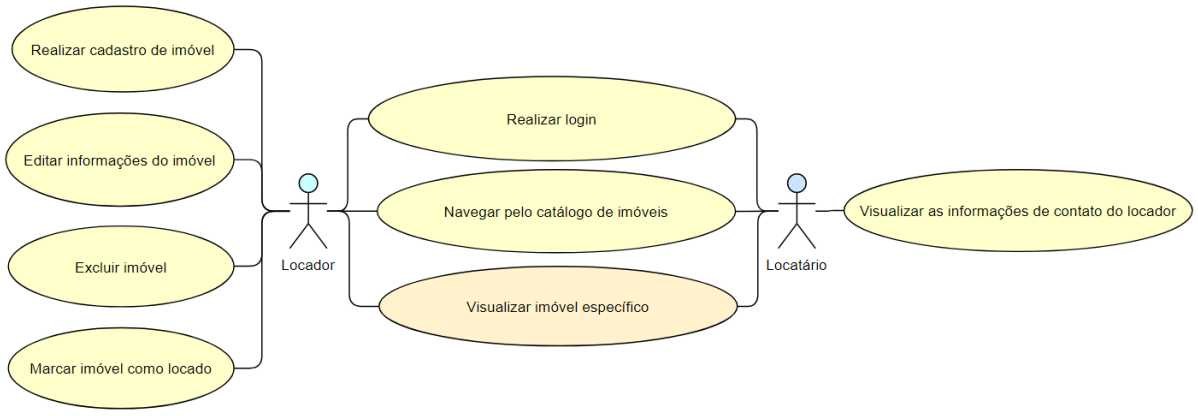

Figura 5. Modelagem do caso de uso

4.3. Diagrama de Classes

O diagrama de classes representado pela Figura 6, demonstra as associações entre classes bem como suas cardinalidades. Nele é possível ver as classes *Immobile* e *User* que servem para definir a estrutura do projeto. O *AuthController*, *ImmobileController* e *UserController* fazem toda a lógica, como autenticação e cadastros, já o restante das classes servem como telas para a visualização do usuário.

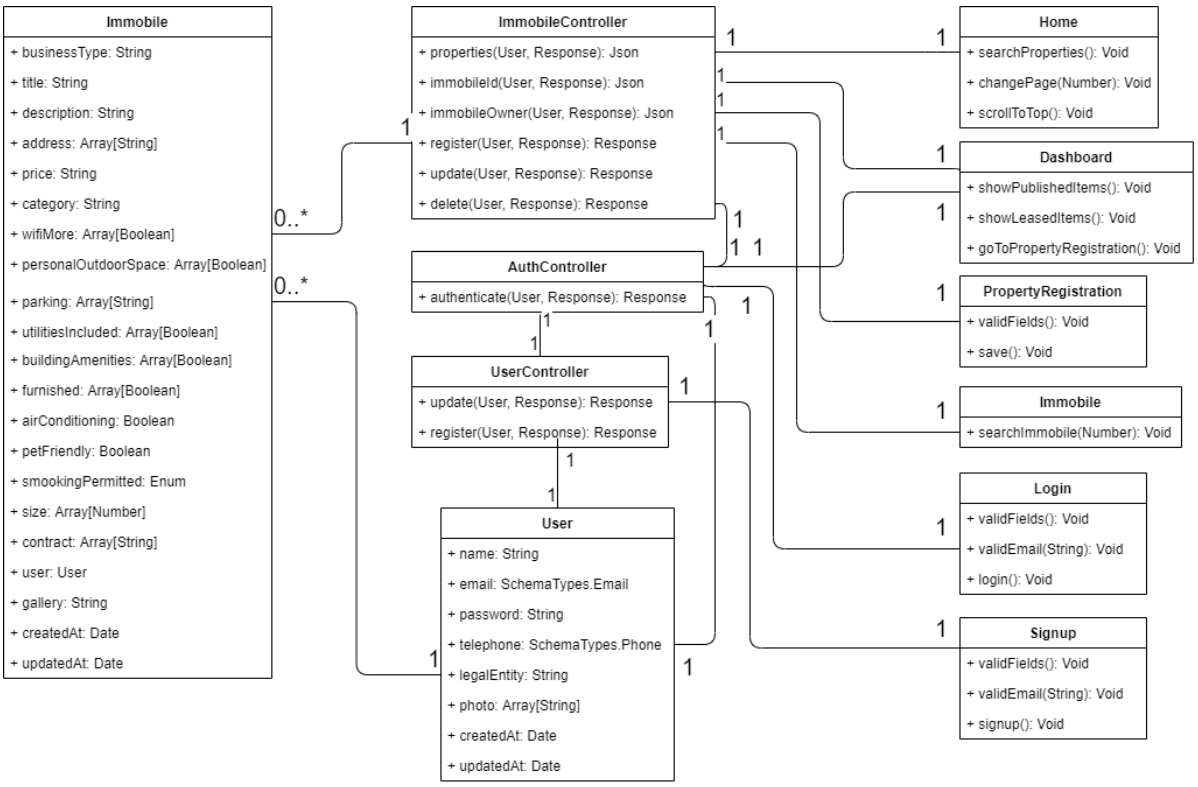

Figura 6. Modelagem do diagrama de classe

4.4. Modelo do Banco de Dados

Como mencionado na Subseção 2.2, para a utilização do banco de dados foi selecionado o MongoDB. A escolha dele se deu após uma análise do modelo do banco dados, do relacionamento entre as entidades, quantidade de relações existes no problema e a familiaridade da equipe com a ferramenta. Com isso, percebeu-se que o sistema poderia sofrer um aumento de volume de dados de uma forma muito fácil, porém, mesmo com o aumento de dados, era importante que o sistema mantivesse disponível.

Com a escolha desta ferramenta, pensou-se também em realizar a modelagem direcionada para ela que é baseada em documentos, com isso desenvolveu-se um esquema de banco de dados apresentado na Figura 7.

Para uma melhor estruturação do banco de dados, a modelagem da mesma ficou dividida em duas coleções, sendo elas:

- **user**: com informações sobre a conta do usuário.
- **house**: com informações sobre o imóvel cadastrado.

As duas coleções se relacionam através do documento *houses* presente na coleção ***user***, pois este armazena um *array* formado por objetos do tipo ObjectId que referenciam a coleção ***house***.

Na Figura 7 é possível ver as duas coleções e suas respectivas chaves e tipagens de cada valor, juntamente com algumas validações que o próprio MongoDB disponibiliza na sua documentação para a garantia da padronização dos dados.

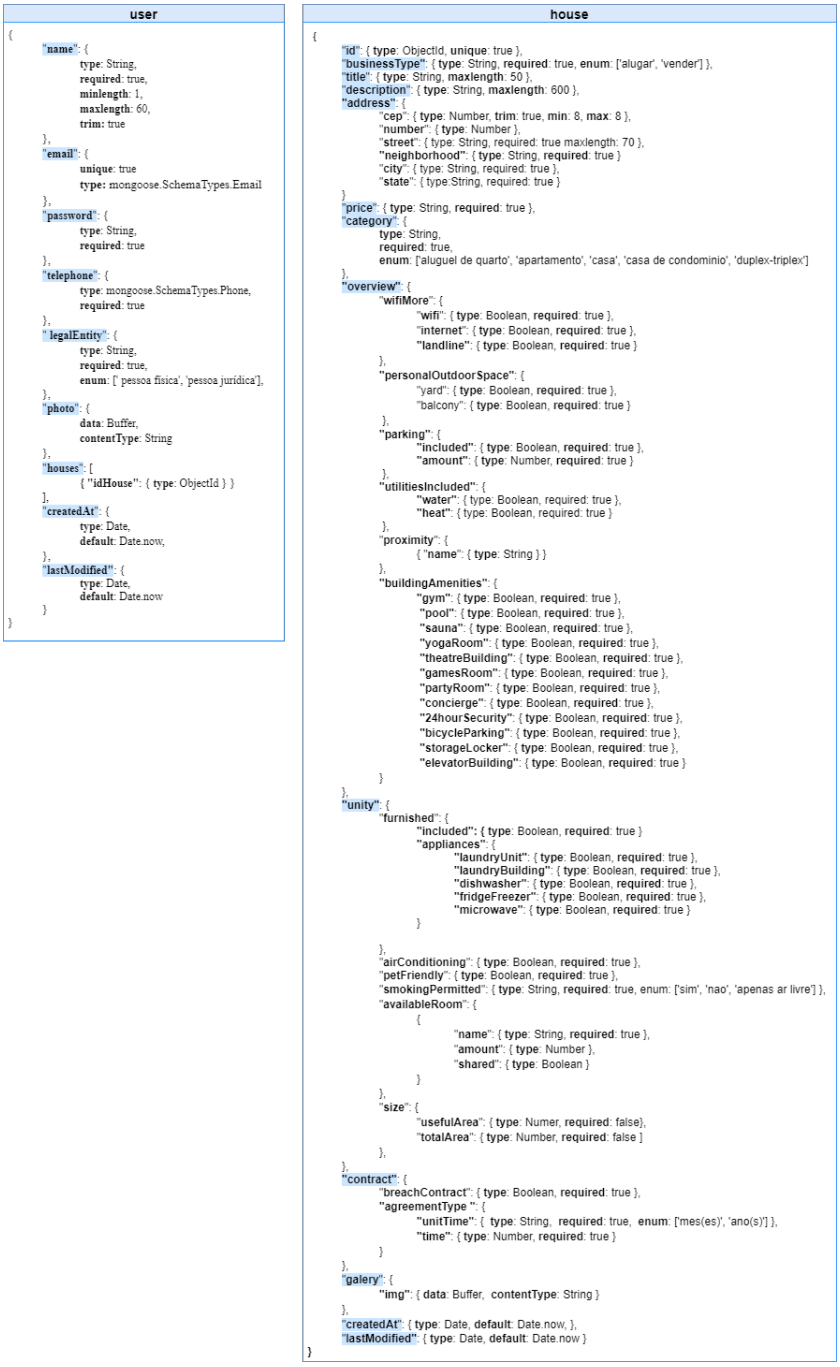

Figura 7. Modelagem do banco de dados

4.5. Rotas da Aplicação

O roteamento da aplicação utiliza o estilo de arquitetura *representational state transfer (REST)*, que segundo Dal Moro et al. (2011) o mesmo provê uma interface uniforme com quatro métodos básicos para as quatro operações mais comuns. *GET* para recuperar uma representação de um recurso, *PUT* para criar um novo recurso ou modificar um existente, *DELETE* para deletar um recurso e *POST* comumente utilizado para criação de um novo recurso.



Figura 8. Rotas da aplicação.

Conforme descrito na Figura 8, as rotas são divididas em usuário e imóvel. O usuário possui duas requisições *POST* uma para autenticação e após ter o *token* de autenticação é possível utilizar a outra para fazer o registro e atualizações. O imóvel só exige o *token* para fazer operações de cadastro *POST* e exclusão *DELETE*. Já as consultas *GET* são liberadas a todos os usuários da plataforma.

4.6. Protótipos das Interfaces Gráficas

Para a prototipação do presente sistema escolheu-se pela forma de prototipagem digital por se ter uma maior fidelidade com o produto final, assim, teve-se a ajuda do Figma que é apresentado na Subseção 3.5.5 e dele originou-se as seguintes telas:

1. Tela de cadastro do usuário.

   O cadastro do usuário será efetuado em apenas uma tela, onde nela irá conter o nome completo do usuário, e-mail, senha onde visualmente esconde os caracteres para uma maior proteção e um campo para a confirmação da senha, como pode ser visto na Figura 9.



Figura 9. Protótipo da tela inicial de cadastro do usuário.

2. Tela de cadastro do imóvel.
   Na Figura 10 é possível visualizar a tela inicial de cadastro do imóvel, nela é possível selecionar o tipo de imóvel a ser comercializado, um título para o imóvel e a categoria de imóvel em que ele se encaixa.

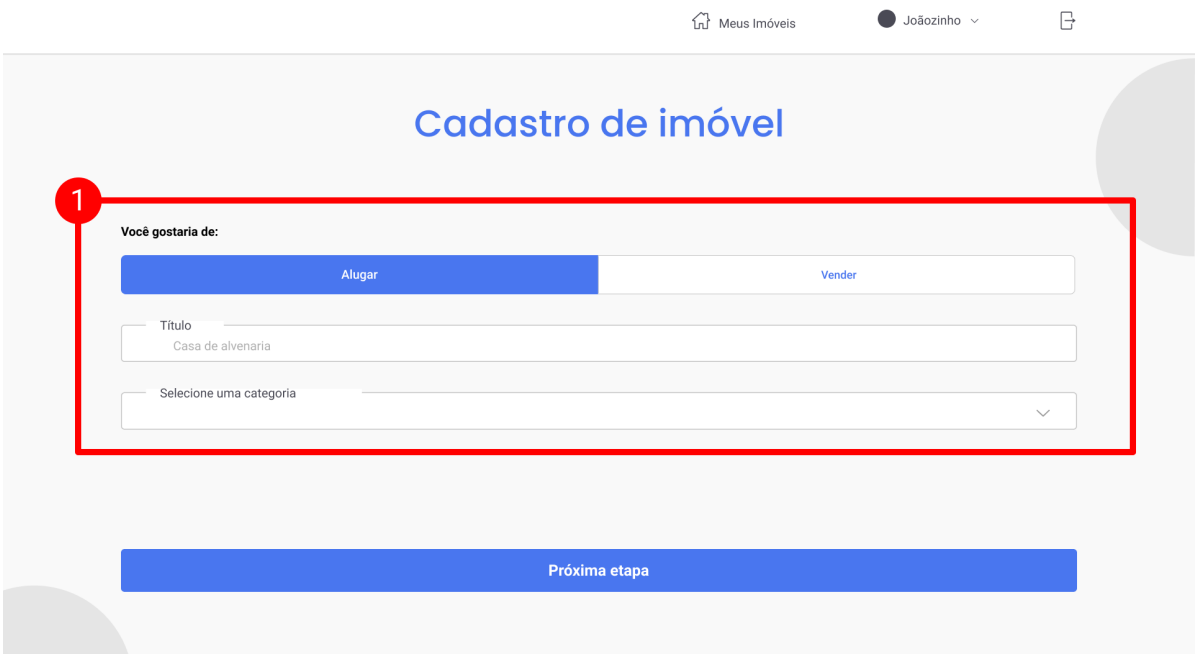

Figura 10. Protótipo da tela inicial do cadastro de imóvel.

A Figura 11 é dividida em 3 seleções, a seleção 1 mostra ao usuário em que ponto

ele se encontra durante a jornada do cadastro do imóvel; a seleção 2 conta com as informações referentes ao que há no imóvel; e por último, tem-se a seleção 3 onde é possível adicionar até 8 imagens do imóvel e uma descrição sobre o mesmo.

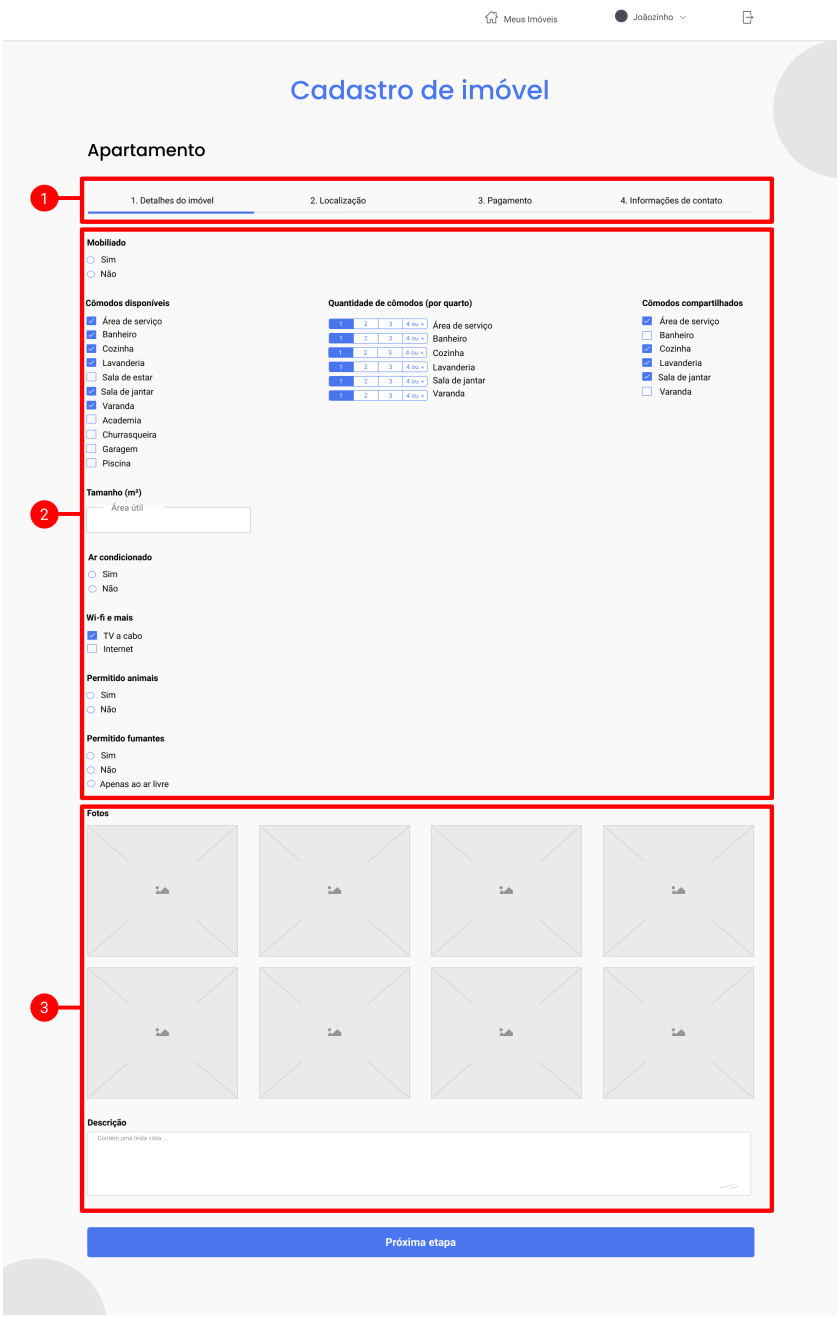

Figura 11. Protótipo da tela do cadastro de detalhes sobre o imóvel.

A etapa 3 do cadastro do imóvel, é apresentado na Figura 12 sendo ele dividido em 2 seleções, sendo a primeira referente a localização do imóvel, podendo o usuário selecionar o local no mapa e ter uma busca automática para preencher os campos de localização que se encontram a até 1km do imóvel, já na segunda seleção, o usuário

pode marcar locais de lazer, comerciais, educacionais e outros lugares importantes na área especificada.

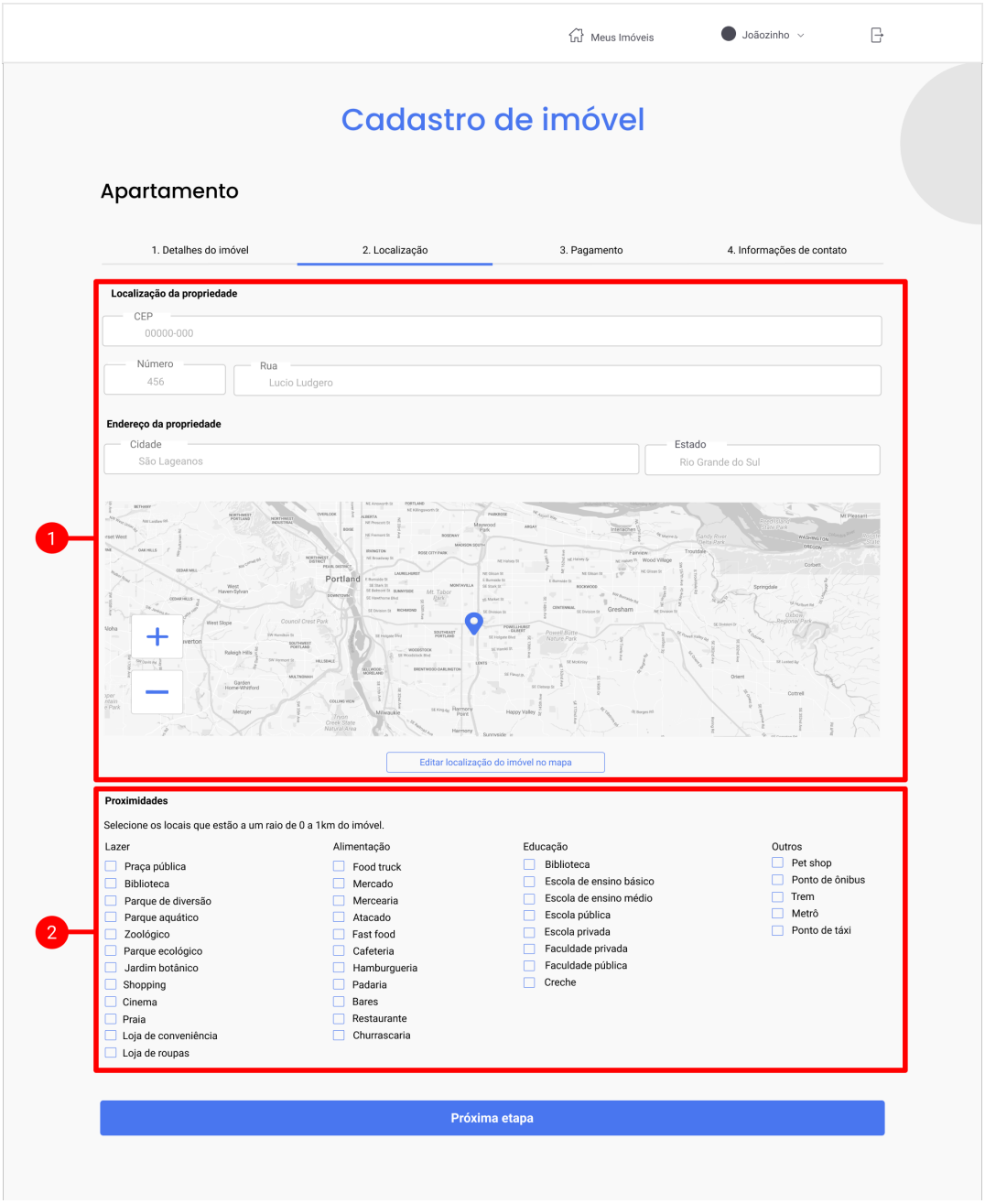

Figura 12. Protótipo da tela do cadastro da localização do imóvel.

Na etapa 4 é apresentado a Figura 13, onde se concentra as informações sobre a negociação do imóvel, sendo eles o preço, o período para a renovação do contrato e se há multa caso ocorra uma quebra de contrato.

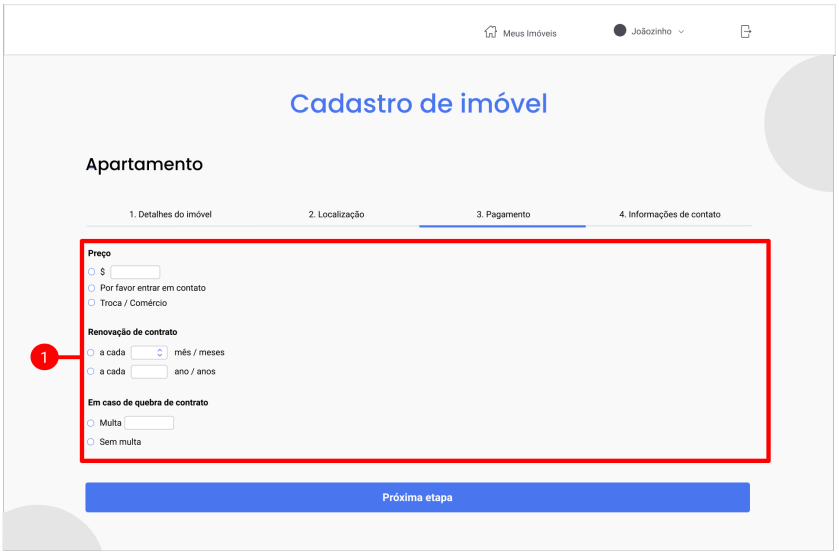

Figura 13. Protótipo da tela do cadastro do pagamento do imóvel.

Como última tela de cadastro do imóvel, tem-se a Figura 14, nela o usuário entra com o seu número de telefone e e-mail para contato. Caso o usuário já tenha alguma dessas informações cadastradas em seu perfil, o próprio *front-end* irá preencher estas informações, podendo também o usuário modifica-las após o preenchimento automático.

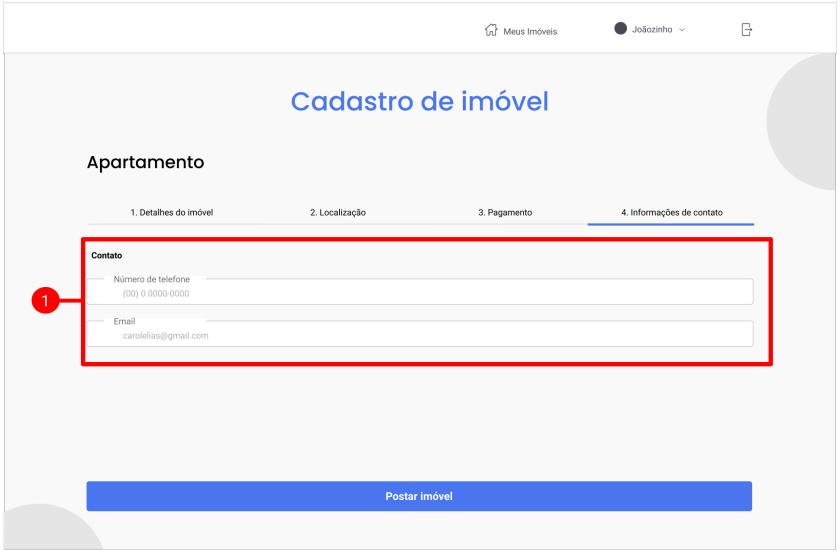

Figura 14. Protótipo da tela do cadastro das informações de contato do imóvel.

3. Tela de visualização do imóvel.
   Com o imóvel já cadastrado é possível visualizar todas as informações cadastradas como na Figura 15, ela é dividida em 3 recortes, onde a seleção 1 irá mostrar as imagens cadastradas do imóvel, juntamente com um botão para entrar em contato com o proprietário por e-mail ou telefone, já a seleção 2 irá conter todas as informações sobre

a região do imóvel e o que há na proximidade e por último, na seleção 2 é mostrado as informações obre o imóvel em questão, desde a sua localização, até alguns detalhes como a área e o que seria permitido ou não no imóvel.

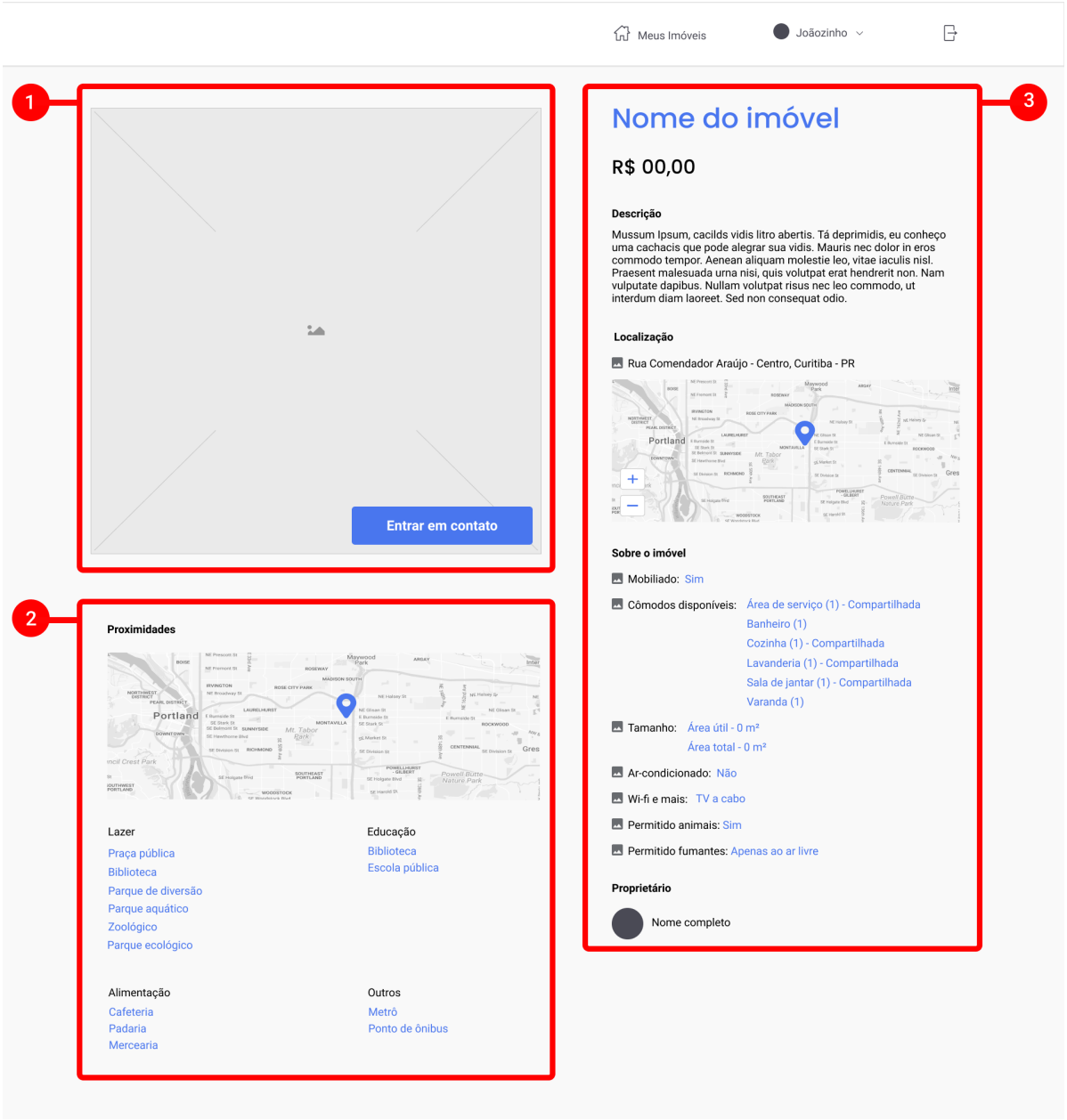

Figura 15. Protótipo da tela de visualização do imóvel cadastrado.

4.7. Kanban de Produção

Conforme dito na Subseção 2.3, com o propósito de melhorar a produtividade, foi elaborado modelo de desenvolvimento do tipo *Kanban* detalhado na Subseção 3.5.7, também foi utilizado o *Trello* descrito na Subseção 3.5.8 como ferramenta para facilitar a organização e desen-

volvimento dos requisitos, desenvolvendo-os por ordem de prioridade, conforme explicado na Subseção 4.1 e demonstrado na Figura 16.

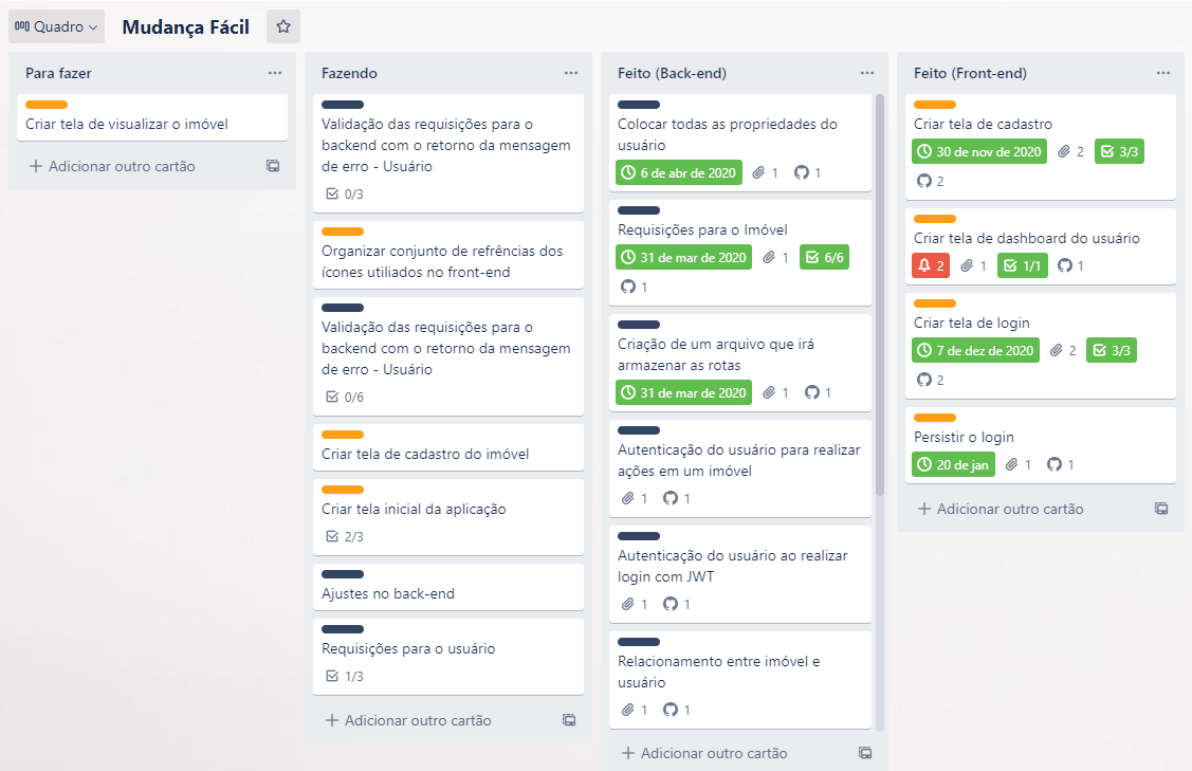

Figura 16. Organização das atividades desenvolvidas.

4.8. Implementação do Sistema

A equipe analisou as funcionalidades e foi definida uma ordem de requisitos a serem implementados. Essa ordem foi escolhida com base nas funcionalidades que geram maior agregação de valor ao produto. Pois, o objetivo era reduzir custos enquanto se desenvolvia um produto com o máximo de qualidade possível. Como descrito na Subseção 4.1, os requisitos foram definidos por ordem de relevância e desenvolvidos nessa estrutura até que contemplassem o que foi elaborado na prototipação.

Visando aproveitar o conhecimento prévio da equipe e da ampla biblioteca disponível, o projeto foi todo desenvolvido em *Javascript*, descrito na Subseção 3.5.2 com Node.js mencionado na Subseção 3.5.3, ele foi dividido em front-end utilizando o Vue.js explicado na Subseção 3.5.6, pois ele facilita o uso da *API* e desenvolvimento das telas, enquanto no back-end foi utilizado o *Express* citado na Subseção 3.5.4, pois ele dá as ferramentas necessárias para implementar o *REST* e fazer o roteamento da *API*, para o desenvolvimento de ambos a equipe utilizou *Visual Studio Code* especificado na Subseção 3.5.9.

O projeto está hospedado no *Heroku*, como mencionado na Subseção 3.5.11, ele foi escolhido com o objetivo de facilitar a implementação reduzindo o tempo e recursos de *hardware*, ficando sempre disponível ao uso quando necessário. Para utilizar nossa plataforma

basta ter Internet e um aparelho conectado nela, pois por ser um site ele não requer recursos de *hardware* do usuário.

4.8.1. Sistema de Controle de Versão

Conforme descrito na Subseção 3.5.10, foi utilizado o *GitHub* para hospedagem e controle de versão do código-fonte dos dois projetos. A lista de funcionalidades e suas datas de implementação podem ser observadas no Apêndice A. Como retratado nas Figuras 17 e 18 o desenvolvimento do *backend* teve um total de 15 *commits* e foram escritas 1289 linhas de código, enquanto o *frontend* teve 33 *commits* e 2118 linhas. As imagens apresentam a linguagem escrita, o número de arquivos de cada projeto, as linhas em branco, comentários e as linhas de código-fonte, assim como a soma de seu respectivo elemento.

Language	files	blank	comment	code
JavaScript	12	80	67	1286
JSON	1	0	0	3
SUM:	13	80	67	1289

Figura 17. Dados do projeto *backend*.

Language	files	blank	comment	code
Vuejs Component	14	128	11	1718
JavaScript	8	34	25	256
SVG	10	0	0	144
SUM:	32	162	36	2118

Figura 18. Dados do projeto *frontend*.

5. Resultados e Discussões

Todas as características na Tabela 1 estão presentes na versão final da aplicação, é possível visualizar outros imóveis próximos no mapa, visualizar informação de contrato, características do lugar, contatar o anunciante, cadastrar livremente os imóveis e ver as atrações, pontos turísticos e comércios presentes ao redor do imóvel.

Na Subseção 5.1 são apresentadas as capturas de telas do sistema desenvolvido, todas as telas apresentadas nesta seção possuem seus respectivos protótipos na Seção 4.6. Na Subseção 5.2 é discutido os principais problemas enfrentados durante o desenvolvimento de todas as partes do sistema e como foram solucionados.

5.1. Telas Finais do Sistema

Para que o sistema tivesse uma maior validade, buscou-se seguir a prototipagem apresentada na subseção 4.6, assim, as funcionalidades presentes no protótipo estão presentes no sistema, mas algumas funcionalidades que estão no sistema não estão no protótipo, pois durante o desenvolvimento fez-se necessário implementá-las para suprir necessidades que não haviam sido pensadas ao se desenvolver o protótipo.



Figura 19. Captura da tela de cadastro de usuário

A Figura 19 demostra o resultado final da tela de cadastro do usuário, onde os campos requeridos permaneceu conforme havia sido projetado no protótipo.

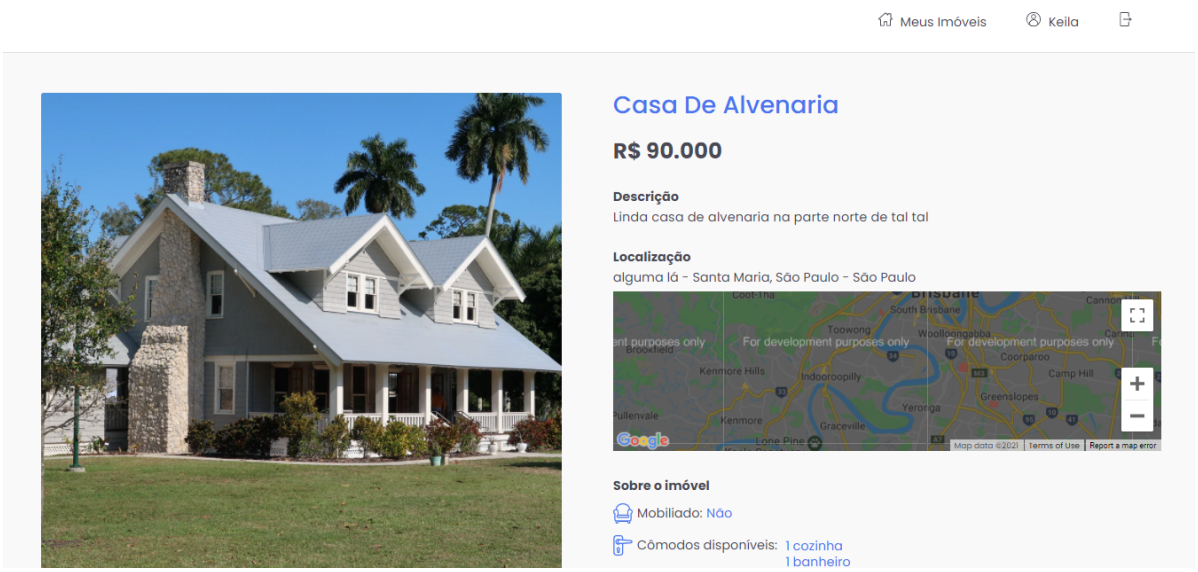

Figura 20. Captura da tela de visualizar imóvel

Para a tela de visualização do imóvel como apresentado na Figura 20, assim como em outras telas, buscou-se seguir com fidelidade o protótipo, porém, para a funcionalidade de mostrar a localização do imóvel, foi possível somente utilizar uma versão de desenvolvimento já que a API do Google Maps possui algumas restrições para o uso das chaves de acesso.

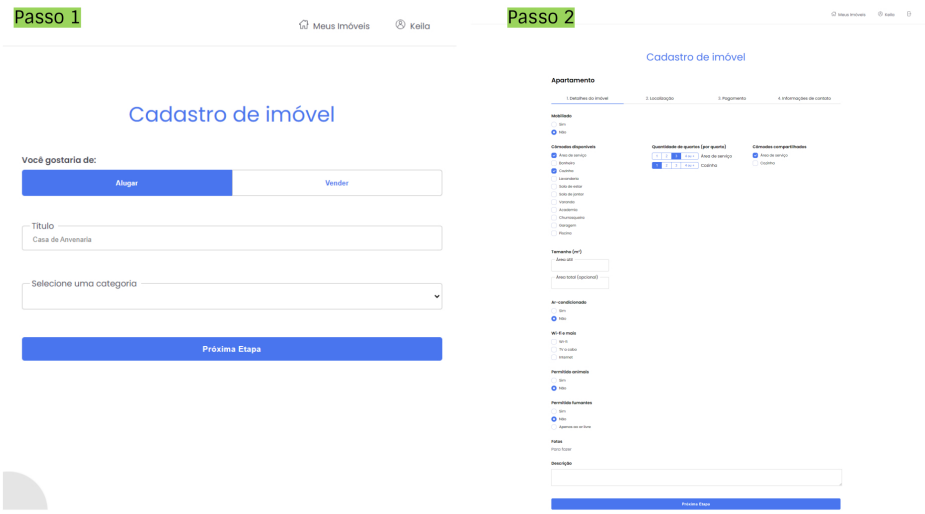

Figura 21. Captura da tela do primeiro e segundo passo da tela de cadastro do imóvel

A tela de cadastro de um imóvel foi dividida em 5 passos como mostrado na Subseção 4.6. A implementação final do primeiro e segundo passo são mostrados na figura 21, uma funcionalidade que é importante destacar é que ao marcar os cômodos disponíveis, automa-

ticamente irá aparecer ao lado os campos para marcar a quantidades de cômodos daquele secionado e se ele é compartilhado.

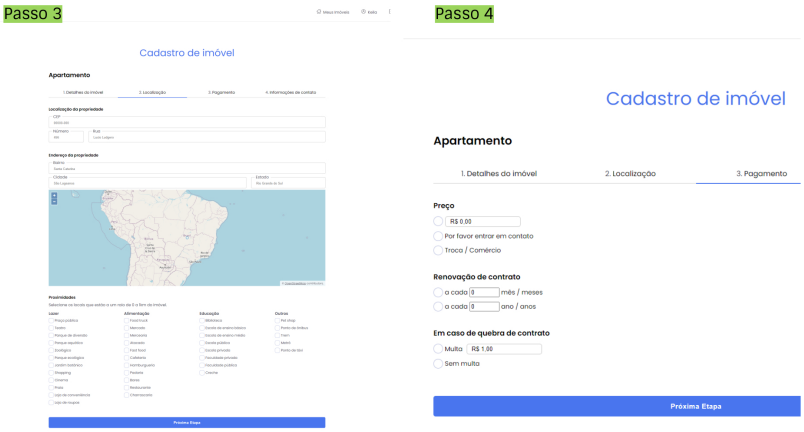

Figura 22. Captura da tela do terceiro e quarto passo da tela de cadastro do imóvel

Na Figura 22 é mostrado o passo 3, onde se tem uma grande obtenção de informações sobre o imóvel, no usuário colocar o CEP (Código de Endereçamento Postal) do imóvel que irá ser negociado, os campos de Rua, Bairro, Cidade e Estados são preenchidos automaticamente se o CEP for válido, pois é feita uma busca na API dos correios para facilitar o preenchimento desses campos. E no Passo 4 é onde o usuário irá informar o valor e outras informações sobre o pagamento do imóvel, foi adicionado máscaras nos campos de moeda, para formatar conforme o Real.

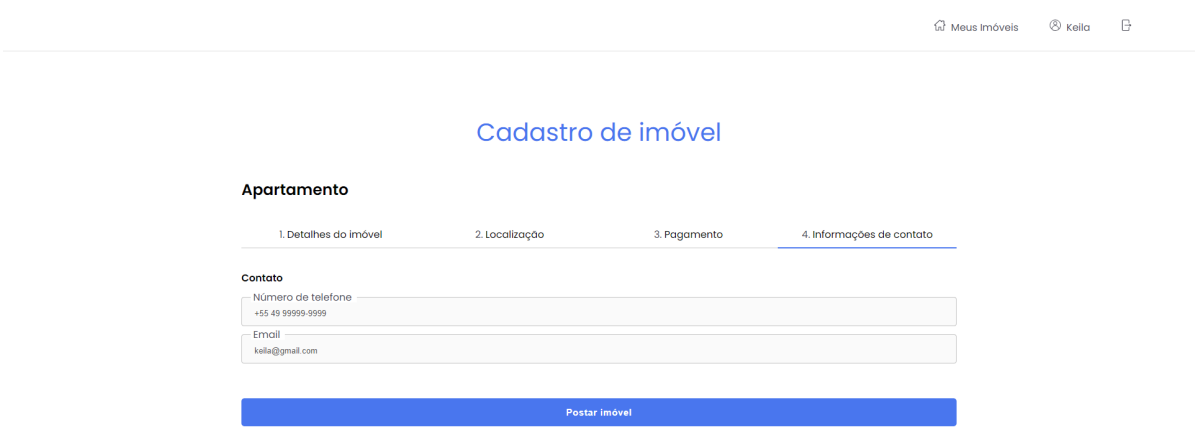

Figura 23. Captura da tela do quinto passo da tela de cadastro do imóvel

Por último, é apresentado uma tela com as informações de contato do vendedor do imóvel, eles campos são desabilitadas para edição, pois essas informações são buscadas diretamente no perfil do usuário, caso ele queira mudar, precisa editar seu perfil. Assim, após está tela, o usuário clica em Postar Imóvel e ele será adicionado a lista de imóveis para negociação.

5.2. Dificuldades Encontradas

Durante todo o processo de criação dos protótipos, da modelagem de banco de dados, da escolha de ferramentas a serem utilizadas e do desenvolvimento do sistema, foram necessárias muitas revisões para que não houvesse dados redundantes ou desnecessários para o sistema, assim como, quais informações seriam obrigatórias e quais seriam opcionais.

Já no desenvolvimento houve alguns problemas ao implementar a paginação dos imóveis, já que esta funcionalidade não estava prevista nos protótipos das interfaces, logo, a modelagem do código do projeto também não previa esta funcionalidade, mas, a dificuldade da paginação foi contornada pela estrutura utilizada no sistema, que possibilitava fazer alterações na resposta da requisição sem que houvesse efeitos colaterais tanto à modelagem do sistema quanto à outros *end-points* do sistema.

Outra funcionalidade que é encontrada facilmente nos sistemas atuais e que houve uma dificuldade em implementar, foi a persistência da sessão do usuário no sistema, pois ao recarregar a página o usuário saia da sessão já que ela não era salva em nenhum lugar, para resolver este problema foi utilizado o *local storage* do navegador, pois nele é armazenado informações que mesmo após fechar o navegador é possível recuperar ao abri-lo novamente.

Deste modo, ao usuário conectar-se em sua conta, automaticamente as informações são salvas em uma variável chamada MF-USER no *local storage* do usuário e cada vez que o usuário reabrir o navegador, o sistema irá verificar se essa variável existe no *local storage*, se existir a conta irá conectar-se automaticamente, se não o usuário precisa conectar-se novamente.

6. Conclusões

O objetivo do trabalho de conclusão de curso foi desenvolver uma plataforma de locação de imóveis, que permitisse aos usuários obterem informações relevantes sobre a região que rodeia o imóvel. Este tema é importante, pois como descrito na Introdução 1 existe uma grande escassez de dados centralizados sobre os lugares onde os imóveis estão situados.

A equipe atingiu todos os objetivos propostos durante o artigo, ele reúne diversas informações sobre a região dos imóveis, propiciando aos usuários facilidade e economia de tempo em suas buscas, alguns requisitos planejados não foram possíveis de implementar em todo o sistema, assim como alguns requisitos foram adicionados no decorrer do projeto.

Trabalhos futuros incluem: indicações personalizadas com base no perfil e localização do usuário, traduzir e localizar para diversos idiomas, implementar um chat interno para facilitar a negociação entre os usuários e hospedar o site para que todos os usuários possam utilizá-lo.

Referências

D. J. Anderson. *Kanban: successful evolutionary change for your technology business*. Blue Hole Press, 2010.

G. Baehr. Sistema para gerenciamento de atividades utilizando a técnica kanban aplicado a desenvolvimento de software. 2016.

F. Boaglio. *MongoDB Construa novas aplicações com novas tecnologias*. Casa do Código, Rua Vergueiro, 3185 - 8º andar 04101-300 – Vila Mariana – São Paulo – SP – Brasil, first edition, jan 2015.

M. D. Bonifacio. B2c, b2b, b2e, b2g, b2b2c, c2c e marketplace. qual a diferença entre eles?, 2016. URL https://www.fastchannel.com/blog/b2c-b2b-b2e-b2g-b2b2c-c2c-e-marketplace-qual-diferenca.

Booking.com™. Sobre a booking.com™, 2016. URL https://www.booking.com/content/about.pt-br.html.

E. Brewer. Pushing the cap: Strategies for consistency and availability. *Computer*, 45(2): 23–29, Feb. 2012. ISSN 0018-9162. doi: 10.1109/MC.2012.37. URL https://doi.org/10.1109/MC.2012.37.

E. Brown. *Web development with node and express: leveraging the JavaScript stack*. O'Reilly Media, 2019.

S. Chacon and B. Straub. *Pro git*. Apress, 2014.

P. C. da Cunha Maya and W. R. I. Otero. A influência do consumidor na era da internet. *Revista da FAE*, 5(1), 2002.

T. Dal Moro, C. Dorneles, and M. T. Rebonatto. Web services ws-* versus web services rest. *Revista Eletrônica de Iniciação Científica em Computação*, 11(1), 2011.

R. Darski. Busca de indicadores de eficiência no uso de recursos em aplicações de transmissão por upload em um estudo comparativo de um servidor node. js para middleware versos um apache tomcat. 2016.

A. S. DE BRITO. Ferramentas para desenvolvimento multiplataforma de aplicativos móveis com javascript e html5: Uma análise comparativa. 2013.

H. G. de Mendonça. E-commerce. *Revista IPTEC*, 4(2):240–251, 2016.

K. G. S. de Oliveira. Análise da utilização do lean e kanban em serviços de software. 2014.

W. de Pádua Paula Filho. *Engenharia de software*, volume 2. LTC, 2003.

Express. Express framework web rápido, flexível e minimalista para node.js, 2017. URL https://expressjs.com/pt-br/.

A. I. S. Fernandes. *Estudo comparativo entre Interfaces de Programação de Aplicações de mapas*. PhD thesis, Faculdade de Ciências e Tecnologia, 2012.

Figma. Figma, 2020. URL https://www.figma.com.

D. Flanagan. *JavaScript: the definitive guide*. "O'Reilly Media, Inc.", 2006.

O. Foundation. Node, 2019. URL https://nodejs.org/en/.

D. Gavras. Déficit habitacional é recorde no país - economia, Jan 2019. URL https://economia.estadao.com.br/noticias/geral,deficit-habitacional-e-recorde-no-pais,70002669433.

Github. Github, aug 2019. URL https://github.com.

Heroku. O que é heroku, 2007. URL https://www.heroku.com/what#summary.

IBGE. Tamanho da família brasileira, 2016. URL https://labsfac.ufsc.br/2016/05/23/.

Imobille. Sobre nós, 2021. URL https://br.linkedin.com/company/imobille-neg%C3%B3cios-imobili%C3%A1rios.

H. A. Johnson. Trello. *Journal of the Medical Library Association: JMLA*, 105(2):209, 2017.

Kijiji. About us, sep 2011. URL https://www.kijiji.ca/kijijicentral/general/about-us/.

B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

M. R. Klein, A. Massuquetti, and G. Spricigo. Migrações internas: um estudo do município de novo hamburgo (rs). *Ensaios FEE*, 33(2), 2012.

N. Lepkova, E. Butkiene, and M. Bełej. Study of customer satisfaction with living conditions in new apartment buildings. *Real Estate Management and Valuation*, 24(3):52–70, 2016.

B. F. Lóscio, H. d. OLIVEIRA, and J. d. S. PONTES. Nosql no desenvolvimento de aplicações web colaborativas. *VIII Simpósio Brasileiro de Sistemas Colaborativos*, 10(1):11, 2011.

M. d. A. Marconi, E. M. Lakatos, et al. *Técnicas de pesquisa*, volume 2. São Paulo: Atlas, 2002.

R. Marinucci and R. Milesi. Migrações internacionais contemporâneas. *Instituto Migrações e Direitos Humanos*, 2005.

D. Matos and P. I. N. Bartkiw. Introdução ao mercado imobiliário. *Curitiba: Instituto Federal de Educação, Ciência e Tecnologia–Paraná–Educação a distância*, 2013.

MDN. O que é javascript?, mar 2019. URL https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Introduction.

MDN. Introdução express/node, abr 2020. URL https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs.

MDR. Programa minha casa, minha vida, Mar . URL https://www.mdr.gov.br/habitacao/programa-minha-casa-minha-vida-pmcmv.

R. V. Mendes et al. Desenvolvimento de uma ferramenta para organização e gerenciamento de atividades de docentes. 2018.

Microsoft. Visual studio code, 2020. URL https://code.visualstudio.com/docs.

MongoDB. Mongodb, 2020. URL https://www.mongodb.com.

OLX. Quem somos, aug 2019. URL https://ajuda.olx.com.br/s/article/quem-somos.

C. R. Pereira. *Aplicações web real-time com Node. js*. Editora Casa do Código, 2014.

J. P. Pichek. A importância da internet para o marketing imobiliário: estudo de caso realizado na imobiliária sefrin, cacoal-ro. 2009.

R. Prikladnicki, R. Willi, and F. Milani. *Métodos ágeis para desenvolvimento de software*. Bookman Editora, 2014.

C. C. Prodanov and E. C. de Freitas. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale, 2013.

P. Rob and C. Coronel. Sistemas de banco de dados. *Projeto, implementação e*, 2011.

R. S. Sampaio. Ambiente de dados do sihsus com mongodb. 2019.

D. d. M. Sanfelici. *A metrópole sob o ritmo das finanças: implicações socioespaciais da expansão imobiliária no Brasil*. PhD thesis, Universidade de São Paulo, 2013.

L. d. C. Tamburu. Instituto federal de educação, ciência e tecnologia do estado de são paulo–campus são roque. 2018.

TPS. Toyota production system. URL https://global.toyota/en/company/vision-and-philosophy/production-system/.

R. Unger and C. Chandler. *A Project Guide to UX Design: For user experience designers in the field or in the making*. New Riders, 2012.

C. E. Vazquez and G. S. Simões. *Engenharia de Requisitos: software orientado ao negócio*. Brasport, 2016.

E. You. Vue, 2020. URL https://vuejs.org.

X. Yuan, J.-H. Lee, S.-J. Kim, and Y.-H. Kim. Toward a user-oriented recommendation system for real estate websites. *Information Systems*, 38(2):231–243, 2013.

G. ZAP. Grupo zap, 2018. URL https://www.grupozap.com/#page-block-kfz7ll94osvn4f4sxnl0ltyb9.

7. Apêndices

A. PLANEJAMENTO DE FUNCIONALIDADES

Backend	
Nome do Requisito	**Data**
RF01.Login	01/03/2020
RF02. Adicionar imóvel	16/03/2020
RF03. Remover imóvel	23/03/2020
RF04. Editar imóvel	31/03/2020
RF05. Cadastrar usuário	09/04/2020
RF06. Excluir usuário	09/04/2020
RF07. Editar usuário	09/04/2020
RF08. Pesquisar imóvel	17/01/2021

Frontend	
Nome do Requisito	**Data**
RF01.Login	27/08/2020
RF02. Adicionar imóvel	19/02/2021
RF03. Remover imóvel	19/02/2021
RF04. Editar imóvel	26/03/2021
RF05. Visualizar imóvel	24/11/2021
RF06. Cadastrar usuário	15/09/2020
RF07. Excluir usuário	15/09/2020
RF08. Editar usuário	15/09/2020
RF09. Pesquisar imóvel	06/10/2020
RF10. Contatar proprietário do imóvel	24/11/2021