

# Aplicação Mobile: Agregador de Dados Imobiliários

Marco Cezar Moreira de Mattos - a42174

Relatório de estágio apresentado à Escola Superior de Tecnologia e de Gestão de  
Bragança para obtenção do Grau de Mestre em Informática.

Trabalho orientado por:

**Prof. Paulo Jorge Teixeira Matos**

**Prof. Narci Nogueira da Silva**

Bragança

2020-2021



# Aplicação Mobile: Agregador de Dados Imobiliários

Mestrado em Informática  
Escola Superior de Tecnologia e Gestão

Marco Cezar Moreira de Mattos - a42174

Trabalho orientado por:  
**Prof. Paulo Jorge Teixeira Matos**  
**Prof. Narcis Nogueira da Silva**

Bragança  
2020-2021

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

---

Marco Cezar Moreira de Mattos - a42174



# Agradecimentos

Primeiramente gostaria de agradecer a minha família, em especial a minha mãe Adriana Moreia, pela ajuda e pelo apoio incondicional, que muito contribuíram para a realização deste trabalho.

A todos da empresa HydraDev, pela oportunidade de trabalhar no projeto e pelo fornecimento de dados e materiais que foram fundamentais para o desenvolvimento deste trabalho.

Também agradeço ao Instituto Politécnico de Bragança e a Universidade Tecnológica Federal do Paraná por todo o conhecimento e novos horizontes que me proporcionaram. Em especial, gostaria de agradecer os meus orientadores Prof. Paulo Jorge Teixeira Matos e Prof. Narci Nogueira da Silva que foram imprescindíveis na minha jornada acadêmica.

Por último, agradeço a todos os amigos que de alguma forma foram importantes durante a minha jornada acadêmica, em especial Rômulo, Bárbara, André, Diego e John.

# Resumo

O presente trabalho tem por objetivo abordar os conhecimentos que foram adquiridos e colocados em prática durante a realização do estágio. Para tanto, foi desenvolvido toda uma aplicação para suprir as necessidades impostas pela empresa HydraDev, uma empresa de TI que foi fundada em 2015 e é especializada em soluções customizadas. O projeto foi motivado por conta de que a transformação digital está cada vez mais modificando praticamente todas as áreas da indústria. Ainda que o ramo imobiliário ter uma certa resistência por essa transformação, está cada vez mais se moldando nos meios digitais. E para entrar nessa transformação digital esse trabalho tem como objetivo analisar dados imobiliários, agregando dados de diversos *websites* de imobiliárias em um só aplicativo, para assim analisar o comportamento do mercado e possibilitar a que sejam tomadas decisões mais assertivas para quem trabalha nesse ramo. Em razão disso, foi possível atingir os objetivos de aplicar os conhecimentos estudados durante o mestrado e criar um aplicativo que atenda todos requisitos estabelecidos pela empresa. E conseqüentemente, proporcionar aos futuros clientes uma gama de dados e métricas para analisar o mercado imobiliário.

**Palavras-chave:** Imobiliária, React-Native, MongoDB

# Abstract

This work aims to address the knowledge that was acquired and put into practice during the internship performed. Therefore, an entire application was developed to meet the needs imposed by the company HydraDev, an IT company that was founded in 2015 and specializes in customized solutions. The project was motivated by the fact that the digital transformation is increasingly modifying practically all areas of the industry. Although the real estate industry has some resistance to this transformation, it is increasingly shaping itself in digital media. And to enter into this digital transformation, this work aims to analyze real estate data, aggregating data from different real estate websites in a single application, in order to analyze market behavior and enable more assertive decisions to be taken for those who work in that branch. As a result, it was possible to achieve the objectives of applying the knowledge studied during the Master's and creating an application that meets all the requirements established by the company. And consequently, provide prospective clients with a range of data and metrics to analyze the real estate market.

**Keywords:** Real estate, React-Native, MongoDB



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Estrutura do Documento . . . . .	2
<b>2</b>	<b>Contexto e Tecnologias</b>	<b>3</b>
2.1	Empresa . . . . .	3
2.2	Problema . . . . .	4
2.3	Tecnologias . . . . .	4
2.3.1	JavaScript . . . . .	5
2.3.2	Node.js . . . . .	5
2.3.3	Express . . . . .	7
2.3.4	React Native . . . . .	7
2.3.5	Expo . . . . .	8
2.3.6	MongoDB . . . . .	8
2.3.7	Ngrok . . . . .	9
2.3.8	Heroku . . . . .	9
<b>3</b>	<b>Análise e Abordagem</b>	<b>11</b>
3.1	Digitalização do Mercado Imobiliário . . . . .	11
3.2	Trabalhos Relacionados . . . . .	12
3.3	Métricas . . . . .	13

3.4	Filtros . . . . .	15
3.5	Ordenações . . . . .	16
3.6	Âmbito do projeto . . . . .	17
3.7	Requisitos . . . . .	18
3.7.1	Requisitos Funcionais . . . . .	18
3.7.2	Requisitos Não Funcionais . . . . .	20
3.7.3	Casos de uso . . . . .	20
<b>4</b>	<b>Desenvolvimento e Resultados</b>	<b>23</b>
4.1	Ambiente de Trabalho . . . . .	23
4.1.1	Visual Studio Code . . . . .	23
4.1.2	Git . . . . .	24
4.1.3	GitHub . . . . .	24
4.1.4	Trello . . . . .	24
4.2	Arquitetura . . . . .	25
4.3	Base de Dados . . . . .	27
4.4	Bibliotecas . . . . .	29
4.5	Entregas e Testes . . . . .	30
4.6	Aplicativo . . . . .	31
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>43</b>
<b>A</b>	<b>Proposta Original do Projeto</b>	<b>A1</b>

# Lista de Figuras

3.1	Filtro proposito. . . . .	15
3.2	Filtro tipologia. . . . .	15
3.3	Filtro tipo. . . . .	16
3.4	Filtro endereço. . . . .	16
3.5	Ordenação por data. . . . .	17
3.6	Ordenação por área. . . . .	17
3.7	Ordenação por preço. . . . .	17
3.8	Diagrama de caso de uso. . . . .	22
4.1	Tarefas no Trello . . . . .	25
4.2	Arquitetura Aplicativo . . . . .	26
4.3	Arquitetura Completa . . . . .	27
4.4	Exemplo de imóvel na base de dados. . . . .	28
4.5	Tela de Login. . . . .	33
4.6	Tela <i>Home</i> . . . . .	34
4.7	Tela filtros propriedade . . . . .	35
4.8	Tela Detalhes da Propriedade . . . . .	36
4.9	Tela Estatísticas da Propriedade . . . . .	37
4.10	Tela Estatísticas Propriedade 2 . . . . .	38
4.11	Tela de Evolução dos Preços . . . . .	39
4.12	Tela de Evolução do Número de Anúncios . . . . .	40
4.13	Tela Comparativo Preços 1 . . . . .	41

4.14 Tela Comparativo Preços 2 . . . . . 42

# Capítulo 1

## Introdução

No presente capítulo, é apresentado o contexto geral do estudo. Para tanto, o capítulo foi dividido em três seções, a saber: enquadramento, objetivos e estrutura do documento. No enquadramento, é apresentado o contexto em que o trabalho se insere. Nos objetivos, é apresentada a proposta original do projeto. Por fim, na parte de estrutura do documento, é apresentada de forma sucinta como estão organizados os capítulos do presente estudo.

### 1.1 Enquadramento

O presente estudo enquadra-se no contexto da disciplina Estágio pertencente ao Mestrado em Informática, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança.

O presente projeto foi desenvolvido na empresa HydraDev, uma empresa da área de Tecnologia da Informação (TI), que apresenta a necessidade de desenvolver uma plataforma *mobile* para processos internos do setor imobiliário.

A plataforma desenvolvida tem por objetivo agregar dados imobiliários de várias fontes, fornecendo dados para pesquisas e análises de mercado complexas para um futuro comprador ou vendedor individual, como descrito no apêndice A.

## 1.2 Objetivos

O objetivo do presente estudo é abordar os conhecimentos que foram adquiridos e colocados em prática durante a realização do estágio. Como objetivos específicos, foram obtidos os seguintes:

Plano de trabalho:

- Familiarizar com a Application Programming Interface (API) existente na empresa;
- Definir a arquitetura da aplicação;
- Implementar componentes de pesquisas;
- Implementar componentes de métricas do mercado imobiliário;
- Realizar testes de integração;
- Escrita do relatório final.

## 1.3 Estrutura do Documento

O presente trabalho está disposto na seguinte forma: No Capítulo 2 é apresentada a fundamentação teórica sobre todo eco-sistema em que o presente projeto se insere. O Capítulo 3 apresenta uma análise do mercado na qual a solução proposta se insere, juntamente com as métricas utilizadas e seus requisitos.

Já no Capítulo 4 é apresentado todo contexto de desenvolvimento, passando pelo ambiente de trabalho, a arquitetura até os resultados do projeto. Por fim, no Capítulo 5 são apresentadas as considerações finais do projeto.

# Capítulo 2

## Contexto e Tecnologias

Para uma boa compreensão deste documento é necessário entender o contexto em que o problema está inserido juntamente com os conceitos e ferramentas utilizadas. Portanto, neste capítulo será apresentada uma breve descrição da empresa na qual o estágio foi realizado, em seguida será apresentado o problema ao qual o software desenvolvido visa resolver e as tecnologias utilizadas para tal.

### 2.1 Empresa

A HydraDev é uma empresa de TI que foi fundada no ano de 2015 por talentosos programadores. A empresa tem como objetivo desenvolver as melhores soluções tecnológicas para qualquer problema proposto. Para isso, a HydraDev foca em pesquisa e desenvolvimento, criando um ambiente propício para evolução de seus colaboradores. E, assim, possibilita a criação das melhores soluções em um mercado muito competitivo.

A empresa desenvolve tanto projetos próprios como terceiriza programadores para outras empresas. Todavia, apesar de possuir diversos projetos, a empresa preza pela construção interna do quadro de desenvolvedores. Sempre são realizadas dinâmicas para interação e troca de conhecimentos dos desenvolvedores, de forma que todos possam aprimorar seus conhecimentos.

## 2.2 Problema

Observa-se que a transformação digital se trata de um fenômeno que vem modificando praticamente todas as áreas da indústria. Apesar do ramo imobiliário ter um expressivo atraso nessa mudança, está cada vez mais se moldando nos meios digitais (Dias [1]).

Assim, com essa crescente digitalização das imobiliárias, tem-se cada vez mais anúncios de imóveis na Internet. Essa informação online permite não só que os usuários tenham acesso a dados para arrendar ou comprar um imóvel, mas também permite que outras pessoas tenham acesso aos dados de imóveis de diferentes lugares e características.

Deste modo, pode-se dizer que descobrir qual é o melhor valor para anunciar a venda ou arrendamento de uma morada, não se trata de uma tarefa fácil. Existem diferentes fatores que podem aumentar ou diminuir o valor de um imóvel. Um dos maiores desafios de uma imobiliária é descobrir o melhor preço de uma moradia, sala comercial ou terreno, dado suas características.

Hoje, pode-se observar que existem vários sites imobiliários na Internet anunciando diversos tipos de imóveis em diferentes lugares. De tal maneira, isso acaba por criar um acervo gigantesco de dados e informações que podem ser extraídos e compilados, para então formar uma base de dados útil para compradores e interessados.

Frente a isso, é possível utilizar dessa base de dados para criar uma aplicação que forneça diferentes estatísticas para consultores imobiliários. Dessa forma é possível que o mercado imobiliário consiga definir os preços de seus produtos de forma assertiva.

## 2.3 Tecnologias

A plataforma de estatísticas imobiliárias foi desenvolvida em duas partes. Basicamente, tem-se um servidor que disponibiliza dados para uma aplicação *mobile*. Portanto, nesta seção, apresenta-se uma introdução das tecnologias utilizadas para o desenvolvimento da plataforma proposta.

### 2.3.1 JavaScript

JavaScript é uma linguagem de programação criada pela NetScape em parceria com a Sun Microsystems, com o objetivo de viabilizar a interatividade a uma página web. Sua primeira versão foi lançada no ano de 1995 e foi implementada no primeiro navegador no ano seguinte (Silva [2]).

Inicialmente, a proposta do *JavaScript* era ser apenas uma linguagem para executar tarefas no lado do cliente, ou seja, nos navegadores. Entretanto, é notável o tanto que a linguagem evoluiu e está presente em diversos outros ambientes, como aplicações *mobiles* e em servidores. A crescente popularização do *JavaScript* levou ao desenvolvimento de diversos *frameworks* e bibliotecas (Brown [3]).

Hoje em dia, seu nome oficial é ECMAScript, isto devido a associação responsável encarregada pelas padronizações e atualizações da linguagem *European Computer Manufacturer's Association* (ECMA). Com os avanços do ECMA, foram então desenvolvidas novas ferramentas, como o *React Native* e o *Node.js*, que são o *frontend* e o servidor da aplicação respectivamente.

Assim, basicamente, o JavaScript é uma linguagem de script leve, multiplataforma e interpretada. É bem conhecido pelo desenvolvimento de páginas da web, e muitos ambientes que não são de navegador também o usam. O JavaScript pode ser usado para desenvolvimentos do lado do cliente, bem como para desenvolvimentos do lado do servidor. O JavaScript contém uma biblioteca padrão de objetos, como o Array, Date e Math, e um conjunto básico de elementos de linguagem, como operadores, estruturas de controle e instruções

### 2.3.2 Node.js

O *Node*, ou mais formalmente *Node.js* é um ambiente de tempo de execução *JavaScript* de código aberto e plataforma cruzada, sendo uma ferramenta popular para quase qualquer tipo de projeto. O *Node.js* executa o mecanismo V8 *JavaScript*, o núcleo do *Google Chrome* fora do navegador. Isso permite que o *Node.js* tenha um ótimo desempenho

(Node [4]).

Um aplicativo *Node.js* é executado em um único processo, sem criar um novo *thread* para cada solicitação. O *Node.js* fornece um conjunto de primitivas de I/O assíncronas em sua biblioteca padrão que evita o bloqueio do código *JavaScript* e, geralmente, as bibliotecas em *Node.js* são escritas usando paradigmas sem bloqueio, tornando o comportamento de bloqueio a exceção em vez da norma (Masiello e Friedmann [5]).

Quando o *Node.js* executa uma operação de I/O, como ler dados da rede, acessar um banco de dados ou o sistema de arquivos, em vez de bloquear o encadeamento e desperdiçar ciclos de *CPU* a espera, o *Node.js* retoma as operações quando a resposta regressa. De tal modo, isso permite que o *Node.js* lide com milhares de conexões simultâneas com um único servidor sem precisar gerenciar a simultaneidade de *threads*, que pode ser uma fonte significativa de *bugs* (Lundar, Gronli e Ghinea [6]).

O *Node.js* tem uma grande vantagem; trata-se dos milhões de desenvolvedores de *front-end* que escrevem *JavaScript* para o navegador, que agora são capazes de escrever o código do lado do servidor, além do código do lado do cliente, sem a necessidade de aprender uma linguagem completamente diferente (Node [4]).

No *Node.js*, os novos padrões *ECMAScript* podem ser usados sem problemas, já que não é mais preciso esperar que todos os usuários atualizem seus navegadores – o usuário agora é responsável por decidir qual versão *ECMAScript* usar, alterando a versão *Node.js*, e também pode ativar recursos experimentais específicos executando o *Node.js* com sinalizadores (Node [4]). Existe uma diferença entre executar o *JavaScript* com o *Node.js* e executar no navegador, o acesso às Application Programming Interface (API) do sistema operacional só estão disponíveis no *Node.js*, e também apenas é possível ter acesso a informações sobre janelas quando executado no navegador (Alperstedt, Rolt e Alperstedt [7]).

Além do mais, o *Node.js* é uma ferramenta *standalone* com alta capacidade de escala, sendo sua arquitetura totalmente *non-blocking thread*, evitando que aconteçam *dead-locks* em tempo de execução. Isso ocorre devido ao *Node.js* ser orientado a eventos assíncronos e ser single thread, de forma que apenas uma *thread* seja utilizada para processar o *loop*

de eventos (Node [4]).

### 2.3.3 Express

O Express é um pequeno e flexível *framework* que provê um conjunto robusto de funcionalidades para aplicações web e *mobile*. O Express possui boa performance e é altamente utilizado no desenvolvimento de aplicações *back-end* em conjunto com o *Node.js*. Escrito em *JavaScript*, o *framework* é utilizado por diversas empresas ao redor do Mundo, entre elas a Fox Sports, PayPal, IBM e Uber.

Muito popular tanto em grandes empresas quanto na comunidade, o *Express* facilita a criação de aplicações utilizando o Node em conjunto com o *JavaScript*, tornando este ecossistema ainda mais poderoso (Brown [8]).

### 2.3.4 React Native

Por vários anos, o *React Native* tem sido um assunto recorrente no Mundo do desenvolvimento móvel. Frente a isto, não é de admirar que ele conquistou o mundo da tecnologia ao oferecer uma nova maneira de desenvolver aplicativos móveis para *iOS* e *Android* simultaneamente (Horton e Vice [9]).

O *React Native* foi adotado com sucesso por centenas de empresas em todo o mundo, incluindo Uber, Microsoft e Facebook, e é usado em uma ampla gama de indústrias. No entanto, antes de decidir apostar tudo no *React Native*, é crucial que se entenda como funciona e decida se é o mais adequado para um determinado projeto (Boduch [10]).

Basicamente, o *React Native* é uma popular estrutura de aplicativo móvel baseada em *JavaScript* que permite construir aplicativos móveis renderizados nativamente para *iOS* e *Android*. A estrutura permite criar um aplicativo para várias plataformas usando a mesma base de código.

O *React Native* foi lançado pelo *Facebook* como um projeto de código aberto em 2015. Em apenas alguns anos, ele se tornou uma das principais soluções usadas para desenvolvimento móvel. O desenvolvimento em *React Native* é usado para potencializar

alguns dos principais aplicativos móveis do mundo.

Sabe-se que existem várias razões por trás do sucesso global do *React Native*; em primeiro lugar, ao usar o *React Native*, as empresas podem criar código apenas uma vez e usá-lo para potencializar seus aplicativos *iOS* e *Android*. Isso se traduz em grande economia de tempo e recursos (Horton e Vice [9]).

Em segundo lugar, o *React Native* foi construído com base no *React* – uma biblioteca *JavaScript*, que já era muito popular quando o *framework* móvel foi lançado. Em terceiro lugar, a estrutura capacitou os desenvolvedores *front-end*, que anteriormente, só podiam trabalhar com tecnologias baseadas na web, para criar aplicativos robustos e prontos para produção para plataformas móveis (Boduch [10]).

Curiosamente, como acontece com muitas invenções revolucionárias, o *React Native* foi desenvolvido como uma resposta a um grande erro tecnológico. Deste modo, o *React Native* combina as melhores partes do desenvolvimento nativo com *React*, uma das melhores bibliotecas *JavaScript* da classe para construir interfaces de usuário (Antani, Timms e Mantyla [11]).

### 2.3.5 Expo

Expo é uma ferramenta de código aberto que serve para integrar com o *React Native* e facilitar a execução do código para as plataformas *Android* e *iOS*. Expo é fácil de usar, não dependendo de outras bibliotecas e permite acessar APIs nativas do sistema operacional do dispositivo.

Através do Expo é possível testar o código para ambas plataformas sem se preocupar com a preparação de ambiente que cada uma necessita, como a instalação de SDKs para *Android* ou *XCode* para *iOS* (Paul e Nalwaya [12]).

### 2.3.6 MongoDB

O *MongoDB* é um banco de dados orientado a documentos de plataforma cruzada. Classificado como um banco de dados *NoSQL*, o *MongoDB* usa documentos do tipo *JSON*

com esquemas opcionais. O *MongoDB* é desenvolvido pela MongoDB Inc. e licenciado sob a Server Side Public License (SSPL) (Boduch [10]).

Também, o *MongoDB* por ser um banco de dados NoSQL orientado a documentos é usado para armazenamento de dados de alto volume. Em vez de usar tabelas e linhas como nos bancos de dados relacionais tradicionais, o *MongoDB* faz uso de coleções e documentos. Os documentos consistem em pares de valores-chave que são a unidade básica de dados no MongoDB (Banker, Garrett, Bakkum et al. [13]).

As coleções contêm conjuntos de documentos e funções que são equivalentes às tabelas de banco de dados relacional. O *MongoDB* se tornou um dos bancos de dados *NoSQL* mais populares, sendo usado para armazenar dados no *back-end* para muitos sites importantes, incluindo eBay, Craigslist, SourceForge e The New York Times (Banker, Garrett, Bakkum et al. [13]).

### 2.3.7 Ngrok

O Ngrok trata-se de um aplicativo de plataforma cruzada que permite aos desenvolvedores expor um servidor de desenvolvimento local à Internet com o mínimo de esforço. O software faz com que um servidor da web hospedado localmente pareça estar hospedado em um subdomínio de ngrok.com, o que significa que nenhum IP público ou nome de domínio na máquina local é necessário. Deste modo, pode-se dizer que o Ngrok é um aplicativo de plataforma cruzada que expõe as portas do servidor local para a Internet (Siby [14]).

### 2.3.8 Heroku

O Heroku trata-se de um Plataforma como serviço (PaaS) em nuvem baseada em contêiner. Os desenvolvedores usam o Heroku para implantar, gerenciar e dimensionar aplicativos modernos. A plataforma se mostra elegante, flexível e fácil de usar, oferecendo aos desenvolvedores um caminho simples para colocar seus aplicativos no mercado (Boduch [10]).

O Heroku é totalmente gerenciado, dando aos desenvolvedores a liberdade de se concentrar em seu produto principal, sem a distração de manter servidores, hardware ou infraestrutura. A experiência Heroku fornece serviços, ferramentas e fluxos de trabalho (Middleton e Schneeman [15]).

Basicamente, o Heroku é um PaaS que suporta várias linguagens de programação. Uma das primeiras plataformas em nuvem, o Heroku está em desenvolvimento desde junho de 2007, quando suportava apenas a linguagem de programação Ruby, mas, agora, suporta Java, Node.js, Scala, Clojure, Python, PHP e Go. (Middleton e Schneeman [15]).

# Capítulo 3

## Análise e Abordagem

Neste capítulo será apresentada uma análise do contexto, mercado imobiliário, que a solução desse projeto se insere. Também serão apresentadas as métricas que serão utilizadas no projeto.

### 3.1 Digitalização do Mercado Imobiliário

A pouco tempo atrás não parecia tão óbvio que o setor imobiliário e digital iriam se associar da forma que estão atualmente. O setor imobiliário sempre foi percebido como algo pouco moderno e de certa forma resistente a adoção de novos processos tecnológicos. Ao contrário de outros setores que afetam apenas uma parte da população, os imóveis preocupam a todos diariamente. A habitação, seja para alugar ou comprar, é o primeiro item da despesa das pessoas, e uma das suas principais preocupações.

Dito isto, a chegada de novas tecnologias no setor imobiliário, leva os atuais *players* a repensarem suas ofertas e seu marketing imobiliário. Segundo, o setor imobiliário enfrenta uma *uberização*: os usuários do setor exigem inovação e novas funcionalidades. Concretamente, a digitalização do setor imobiliário requer a adoção de ferramentas tecnológicas que vão atrapalhar os processos e usos atuais.

No entanto, o setor não está totalmente transformado, em 2020, ainda se mora em casas e apartamentos, e os incorporadores imobiliários ainda estão construindo imóveis.

Pode-se imaginar que, em um futuro próximo, a profissão de corretor imobiliário irá desaparecer.

Com a chegada de novas tecnologias no setor imobiliário, foi possível que novas empresas entrassem no mercado em que grandes atuantes do mercado imobiliário dominavam. Muitas *startups* de tecnologia criaram ferramentas para reduzir o ponto de dor para o setor imobiliário: visitas virtuais a apartamentos, digitalização de vendas, personalização de acomodações, avaliações de clientes online, residências colaborativas. Essas soluções são principalmente comercializadas para profissionais do setor imobiliário, mas, projetado para os usuários finais de habitação, ou seja, pessoas físicas.

Para os profissionais do setor imobiliário, as novas tecnologias ajudam a aumentar a produtividade, simplificam processos, mas, também permitem que se voltem a focar nas pessoas e nas expectativas dos futuros compradores. Mais do que nunca, as expectativas dos utilizadores em termos de oferta e apoio na procura de alojamento são importantes.

Para os profissionais do setor imobiliário, os desafios da digitalização imobiliária são a adoção de novos processos e/ou novos usos que possibilitem o incremento dos negócios por meio da criação de valor agregado real às suas ofertas. Todavia, não se pode cair na armadilha de adotar processos digitais à toa e sem esperar benefícios reais, mas sim, ser capaz de ter a perspectiva necessária para identificar o “ponto de dor” da atividade e dos clientes, que pode ser corrigido através da adoção de soluções digitais.

Hoje, em Portugal, a digitalização neste setor já se faz presente. Por exemplo, a empresa ERA Portugal disponibiliza às suas mais de duzentas agências em território nacional uma ferramenta tecnológica que facilita a avaliação e busca por imóveis aos seus consultores imobiliários. Essa plataforma consegue em curto espaço de tempo, processar informações e estimativas acerca do mercado imobiliário (Dias [1]).

## **3.2 Trabalhos Relacionados**

O estudo de Aníbal [16] mostra que o mercado imobiliário em Portugal se encontra em um momento de transição. Posteriormente a anos de aumento, tanto nos preços quanto

na quantidade de transações, Portugal adveio por transformações expressivas que afetam inteiramente o segmento imobiliário.

Já o estudo de Sousa [17] aponta que a crise em consequência da pandemia do Coronavírus (COVID-19) veio a alterar as preferências de quem investe no setor, notadamente com a finalidade de residir. Apesar de o número de vendas terem diminuído, o segmento imobiliário não demonstra grandes perdas, como outros setores.

O trabalho de Andrade [18] apresenta que a pandemia trouxe um maior uso de internet na população, o que acaba fazendo com que empresas e segmentos precisem convergir para o mundo digital, desenvolvendo possibilidades de negócios digitais. Com vistas ao período pós-pandêmico em Portugal, o “novo normal” trará adaptações a esta nova realidade.

### 3.3 Métricas

Nesta seção serão apresentadas as métricas que foram implementadas e apresentadas na aplicação. Todas as métricas foram previamente estabelecidas pela empresa e tem como objetivo proporcionar dados para a análise de valores de imóveis que estão ou vão ser colocados à venda.

No próximo capítulo será abordado detalhadamente como está disposta a base de dados do projeto, entretanto, temos basicamente dois tipos de dados armazenados que serão transformados e apresentados como estatísticas. Os dois tipos de dados são dados agregados em função do tempo, mais especificamente mês a mês e também temos dados baseados na média global, levando em consideração apenas o último valor obtido de cada imóvel.

Assim, temos as métricas de evolução no tempo baseadas em característica definidas por parâmetro do imóvel e as métricas que são baseadas no último valor do imóvel e também são comparados seus valores baseados em característica definidas por parâmetro. Segue detalhadamente as métricas e seus parâmetros:

- **Evolução do preço:** É calculado a média mensal dos imóveis baseado nos parâmetros de busca, sendo eles a tipologia, tipo do imóvel e localização.

- **Evolução da quantidade de anúncios:** Nessa métrica é calculado a média mensal da quantidade de anúncios de imóveis ativos baseado nos parâmetros de busca, sendo eles a tipologia, tipo do imóvel e localização.
- **Comparativo pela quantidade de quartos:** Neste item é comparado a evolução de preços mensais de dois imóveis de mesma localização e tipo, porém com a tipologia distinta.
- **Comparativo pela localização:** Neste item é comparado a evolução de preços mensais de dois imóveis de mesma tipologia e tipo, porém com a localização distinta.
- **Comparativo por preço médio:** Compara mostrando a percentagem da diferença de preço do imóvel pela média de todos outros que tenham o mesmo tipo e estejam localizados no mesmo bairro e/ou cidade.
- **Comparativo de preço por área útil:** É a percentagem da diferença de preço da área útil de um imóvel pela média de todos outros que tenham o mesmo tipo e estejam localizados no mesmo bairro e/ou cidade.
- **Comparativo de área útil:** É a percentagem da diferença da quantidade de área útil de um imóvel pela média de todos outros que tenham o mesmo tipo e estejam localizados no mesmo bairro e/ou cidade.
- **Comparativo de preço por área de terreno:** É a percentagem da diferença de preço da área de um imóvel pela média de todos outros que tenham o mesmo tipo e estejam localizados no mesmo bairro e/ou cidade.
- **Comparativo de área de terreno:** É a percentagem da diferença da quantidade de área de um imóvel pela média de todos outros que tenham o mesmo tipo e estejam localizados no mesmo bairro e/ou cidade.

## 3.4 Filtros

Devido ao fato do projeto ser um agregador de dados, e conseqüentemente tenha que manipular quantidades gigantescas de dados, é imprescindível que possua filtros para que seja possível analisar os dados. O sistema utiliza 5 filtros para poder agrupar e mostrar ao usuário dados imobiliários, segue abaixo uma breve descrição dos filtros.

### 1. Filtro Propósito

O filtro propósito serve para definir o propósito do anúncio, podendo ser arrendamento ou venda. Assim como mostra a imagem 3.1.



Figura 3.1: Filtro proposito.

### 2. Filtro Tipologia

A tipologia serve para filtrar a quantidade de quartos, sendo descrito com a letra "T" mais um número que representa a quantidade de quartos. O filtro pode ser escolhido entre "T0" e "T6+", assim como mostra a imagem 3.2.

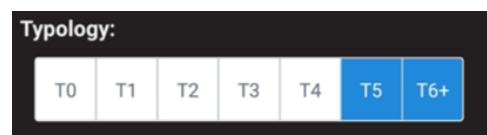


Figura 3.2: Filtro tipologia.

### 3. Filtro Tipo

O tipo serve para filtrar o tipo da propriedade, podendo ela ser: Apartamento; Moradia; Quarto; Terreno; Loja; Armazém; Garagem; Escritório; Prédio; Quinta;

Loft ou Negócio. Após escolhido o tipo fica selecionado conforme a imagem 3.3 mostra.



Figura 3.3: Filtro tipo.

#### 4. Filtro Localização

O filtro de localização pode ser usado de duas maneiras, pode ser selecionado um município e então será filtrado todos imóveis que estão em freguesias desse município. Também é possível selecionar diretamente a freguesia, assim restringindo mais a busca. Na imagem 3.4 pode-se ver um filtro de endereço já selecionado.



Figura 3.4: Filtro endereço.

### 3.5 Ordenações

Além dos filtros, o sistema também utiliza métodos de ordenação para facilitar a busca de dados. O sistema conta com três diferentes parâmetros de ordenação que podem ser usados tanto de forma crescente, como de forma decrescente. Segue abaixo uma breve descrição de cada método de ordenação.

#### 1. Ordenação por Data

A ordenação por data considera a data de criação do anúncio, podendo ser utilizada de duas formas, crescente e decrescente. Assim como mostra a imagem 3.5.

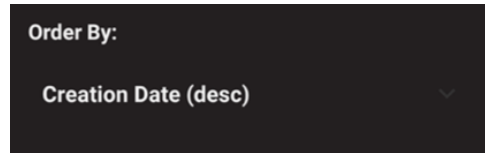


Figura 3.5: Ordenação por data.

## 2. Ordenação por Área

A ordenação por área considera a área útil de um imóvel, podendo ser utilizada de duas formas, crescente e decrescente. Assim como mostra a imagem 3.6.

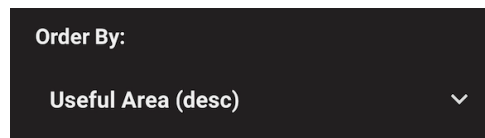


Figura 3.6: Ordenação por área.

## 3. Ordenação por Preço

A ordenação por preço considera o valor anunciado do imóvel, podendo ser utilizada de duas formas, crescente e decrescente. Assim como mostra a imagem 3.7.



Figura 3.7: Ordenação por preço.

## 3.6 Âmbito do projeto

A empresa HydraDev é uma empresa especializada em soluções digitais e recentemente começou a desenvolver soluções para atuar no mercado imobiliário. Nesse contexto nasceu o projeto que foi desenvolvido para o presente relatório. O projeto tem como foco

agregar dados imobiliários de várias fontes e a partir disso disponibilizar ferramentas para pesquisar e realizar complexas análises de mercado.

O projeto é multi plataforma, podendo ser utilizado tanto no iOS como no Android. Para isso foi utilizado o *framework* React Native, pois é preciso escrever apenas um código e é possível gerar o aplicativo para ambas as plataformas. Além disso é uma linguagem de ponta e está em ascensão, assim possibilitando que a empresa HydraDev consiga facilmente novos programadores para continuar e manter o projeto futuramente.

O aplicativo possui duas vertentes principais, sendo a primeira uma pesquisa avançada dos dados imobiliários que já foram previamente processados. A segunda visa apresentar métricas relativas ao mercado imobiliário como um todo, ou a nichos específicos. Em ambas as vertentes a dificuldade está em lidar com uma grande quantidade de dados e apresenta-las com uma interface agradável e relativamente rápida.

## 3.7 Requisitos

O sistema deve prover alguns requisitos que na prática são funcionalidades que esse sistema possui. E nessa seção serão apresentados tanto os Requisitos Funcionais (RF) quanto os Requisitos Não Funcionais (RNF). Todos os requisitos elencados foram desenvolvidos.

### 3.7.1 Requisitos Funcionais

Os requisitos funcionais do sistema estão relacionados a seguir.

#### RF-1. Registrar usuários

O sistema deverá permitir que novos usuários se registrem no aplicativo. Deverão ser preenchidos os campos de usuário, senha e informações pessoais. Sendo que o campo usuário é único, não podendo existir mais de um registro com usuários iguais.

#### RF-2. Fazer o *login* de um usuário

O sistema deverá permitir que usuários registrados façam o *login* no sistema utilizando usuário e senha. Todas funcionalidades do sistema só poderão ser utilizadas após o login, que irá fornecer um *token* de autenticação que deverá ser enviado em todas requisições de dados.

### **RF-3. Listar imóveis**

O sistema deverá permitir que usuários listem imóveis de acordo com diversos filtros, sendo eles: proposito, tipologia, tipo, endereço. A lista deverá carregar mais imóveis conforme o usuário rolar a tela, de forma que o usuário não perceba que estão sendo feita novas requisições.

### **RF-4. Verificar detalhes dos imóveis**

O sistema deverá permitir que usuários selecionem um imóvel na listagem para verificar um detalhamento completo deste imóvel. Devem ser exibidos todas características e descrição completa do imóvel.

### **RF-5. Verificar estatísticas dos imóveis**

Assim como o RF-4, o sistema deverá permitir que usuários selecionem um imóvel na listagem e vejam as estatísticas deste imóvel, que já foram estabelecidas na seção de métricas.

### **RF-6. Mostrar quantidade de anúncios**

O sistema deverá permitir que usuários vejam a quantidade de anúncios ativos ao decorrer do tempo. Deve ser possível filtrar os tipos de anúncios que devem ser considerados na contagem de anúncios ativos.

### **RF-7. Comparar preço de imóveis por tipologia**

O sistema deverá permitir que usuários comparem preços de imóveis ao decorrer do tempo, comparando pela tipologia dos imóveis.

### **RF-8. Comparar preço de imóveis por região**

O sistema deverá permitir que usuários comparem preços de imóveis ao decorrer do tempo, comparando pela localização dos imóveis.

## **3.7.2 Requisitos Não Funcionais**

Os requisitos não funcionais do sistema estão relacionados a seguir.

### **RNF-1. Linguagem de programação do backend**

O backend deverá ser desenvolvido em JavaScript.

### **RNF-2. Banco de dados**

O sistema deverá utilizar o banco de dados MongoDB.

### **RNF-3. Aplicativo multiplataforma**

O aplicativo deverá executar tanto no Android como no iOS.

### **RNF-4. Framework para o aplicativo**

O sistema móvel deverá ser desenvolvido com o framework React Native.

### **RNF-5. Idioma da Interface Gráfica**

O sistema deverá apresentar uma interface gráfica no idioma inglês.

## **3.7.3 Casos de uso**

Nessa seção serão apresentados os usos e aplicações do sistema através de diagramas de casos de usos. O diagrama parte das fases iniciais do projeto, na qual foram elaborados todos requisitos do sistema, sendo esse mais um documento importante para documentação do sistema.

Segundo Pereira [19], um diagrama de casos de uso deve possuir associações entre os dois principais elementos que são os atores e os casos de usos. Os atores são alguma coisa ou alguém que interagem com o sistema.

O presente projeto dispõe apenas de um ator, sendo ele um trabalhador do ramo imobiliário que tenha interesses em métricas do mercado imobiliário ou que tenha interesse em um centralizador de anúncios imobiliários. Pelos fatos de o sistema só poder ser utilizado após o usuário realizar o login e só termos um tipo de usuário, os casos de usos contam apenas com um ator.

Segue uma breve descrição dos casos de uso que estão presentes na figura 3.8.

- **Fazer Registro:** Fazer um breve cadastro com informações pessoais, e-mail e senha.
- **Fazer Login:** Inserir e-mail e senha para aceder as funcionalidades do aplicativo, as quais são todas restritas a usuários autenticados.
- **Listar Imóveis:** Usuário pode selecionar alguns filtros para obter uma lista com uma imagem e breve descrição dos anúncios.
- **Abrir Detalhes do Imóvel:** Após listar os imóveis é possível selecionar um imóvel para visualizar seus detalhes e estatísticas.
- **Comparar Imóveis:** Usuário pode selecionar filtros e tipos de comparações para analisar os imóveis.

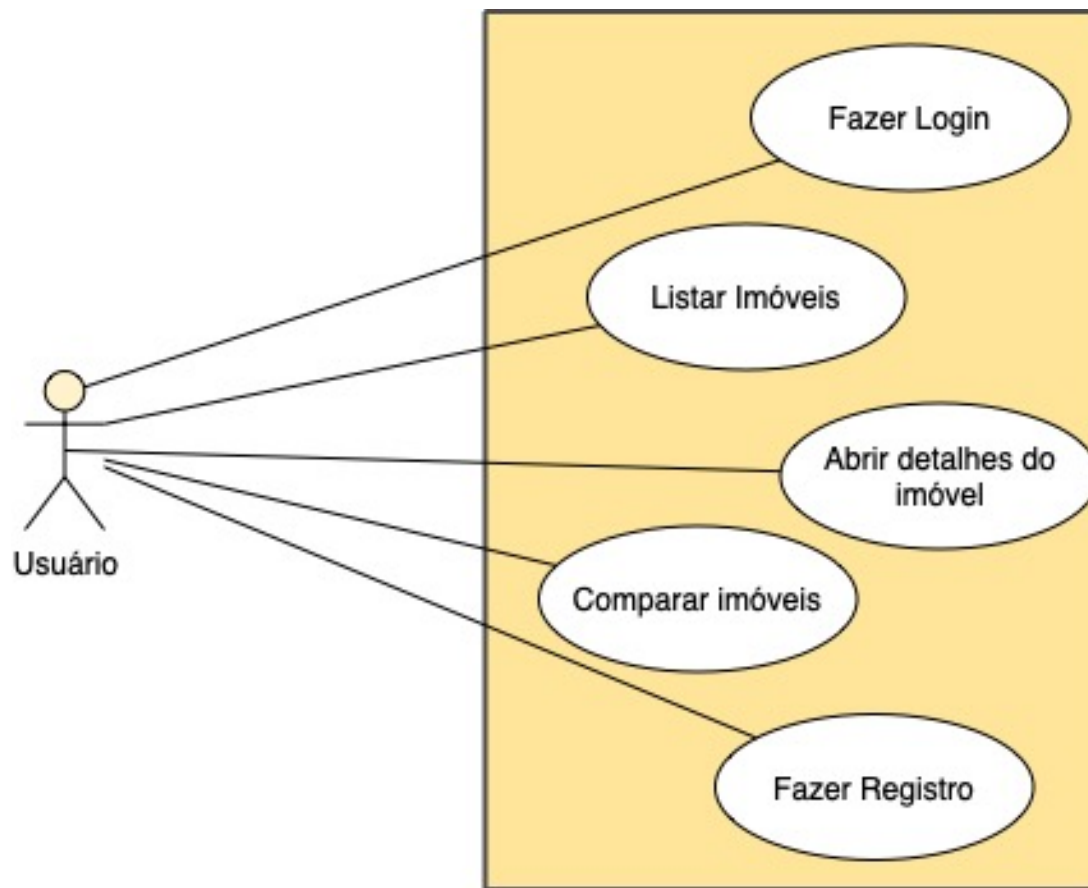


Figura 3.8: Diagrama de caso de uso.

# Capítulo 4

## Desenvolvimento e Resultados

Neste capítulo é descrito o trabalho de implementação de toda a arquitetura que compõe o aplicativo que agrega dados imobiliários da HydraDev. Serão apresentadas as principais ferramentas utilizadas no desenvolvimento, também como detalhes da arquitetura.

### 4.1 Ambiente de Trabalho

Nesta seção serão apresentadas as ferramentas utilizadas no desenvolvimento do projeto.

#### 4.1.1 Visual Studio Code

O Visual Studio Code é um editor de códigos multi plataformas e de código aberto. Esse editor é altamente customizável, através de *plugins* ou configurações adicionais do próprio editor.

Através dos *plugins* é possível adicionar funcionalidades como colorir as palavras de acordo com a regra sintática de cada linguagem de programação ou até mesmo auto completar nomes variáveis e funções. De forma que agilize o desenvolvimento e contribua na prevenção de erros sintáticos.

Além disso, é possível usar *plugins* que integrem com outros serviços como Git, melhorando a experiência do usuário que está desenvolvendo em conjunto com ambas ferramentas.

### 4.1.2 Git

O Git é um sistema de controle de versão distribuído desenvolvido por Linus Torvalds em 2005. Esse sistema mantém um histórico completo de modificações e é possível obter uma cópia de qualquer momento deste histórico. Assim, temos um sistema de versionamento altamente disponível e alta capacidade de restauração (Chacon e Straub [20]).

Todo software em desenvolvimento necessita de versões, seja ela definida em números ou nomes exclusivos. Assim o software pode oferecer um maior detalhamento de seu estado ao longo do tempo para todos que estão envolvidos em seu desenvolvimento. Por exemplo, é possível ver quando uma nova funcionalidade foi adicionada ou uma biblioteca foi removida, etc (Ingeno [21]).

### 4.1.3 GitHub

O GitHub é uma plataforma que hospeda códigos fontes, juntamente com um controle de versão chamado Git. A plataforma foi lançada em 2008 e desde então é possível hospedar projetos, colaborar em projetos abertos, rastrear problemas e vincular ferramentas de integração contínua (*About GitHub* [22]).

### 4.1.4 Trello

O Trello é uma ferramenta online de gerenciamento de projetos que está disponível desde 2011. A ferramenta utiliza o paradigma Kanban para gerenciar as tarefas do projeto. Os projetos são representados por quadros e cada quadro possui uma lista com várias tarefas. As tarefas são representadas por cartões que estão inseridos nas listas e podem ser movidos de lista conforme a tarefa passe pelo fluxo de desenvolvimento (*About Trello* [23]).

A imagem 4.1 demonstra o quadro do presente projeto e todas as fases pelas quais as tarefas tinham que passar.

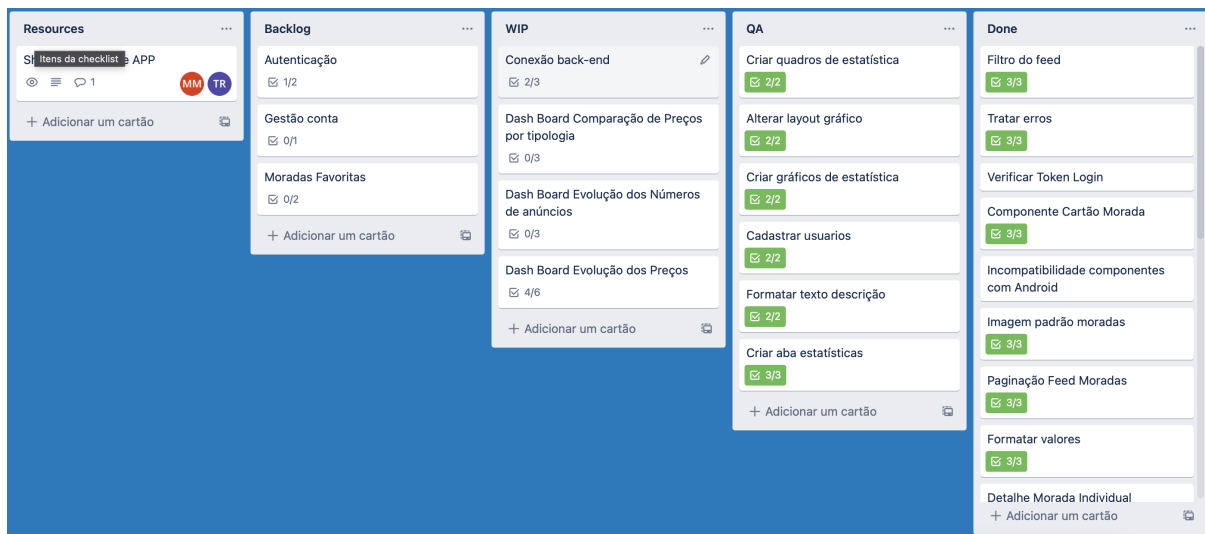


Figura 4.1: Tarefas no Trello

## 4.2 Arquitetura

Como podemos ver na imagem 4.2 o projeto desenvolvido se divide em basicamente 3 módulos, sendo eles a base de dados, servidor e o aplicativo. O acesso à base de dados foi fornecido pela empresa, a base está online em um serviço próprio do MongoDB, chamado de Mongo Atlas. Essa base possui mais de um milhão de instâncias e tem o registro diário das cinco maiores imobiliárias do distrito de Porto.

O servidor REST foi desenvolvido em Node.js com a biblioteca Express e está hospedado na plataforma Heroku. O servidor é responsável por consultar a base de dados e fornecer informações para o aplicativo, e essa comunicação é feita pelo protocolo HTTP.

Já o aplicativo, foi desenvolvido com o *framework* React Native e se comunica com o servidor para prover informações em tempo real ao usuário. O aplicativo é multi plataforma, podendo ser executado tanto no Android como no iOS.

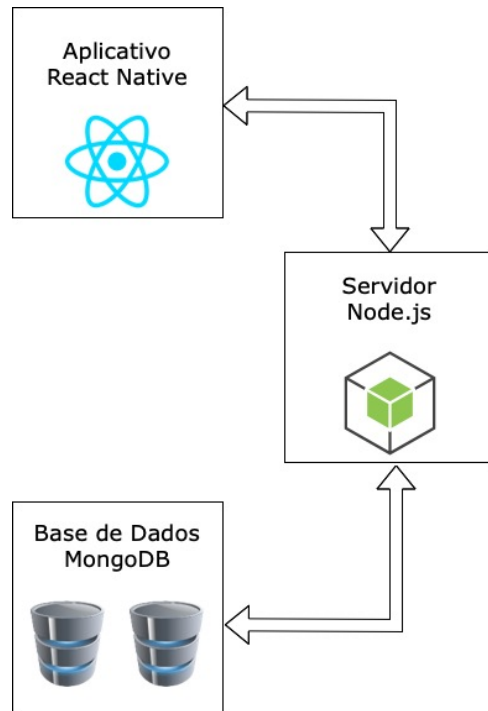


Figura 4.2: Arquitetura Aplicativo

Como foi explicado anteriormente, o projeto utiliza uma base de dados provida pela empresa. Entretanto, a imagem 4.3 demonstra toda arquitetura que envolve o processo completo de obtenção dos dados até sua visualização na aplicação.

Além da arquitetura que já foi explicada na imagem 4.2, temos a adição do Crawler que foi desenvolvido em Python e consulta diariamente os sites de imobiliárias do distrito de Porto e armazena na base que temos em comum com a arquitetura já explicada.

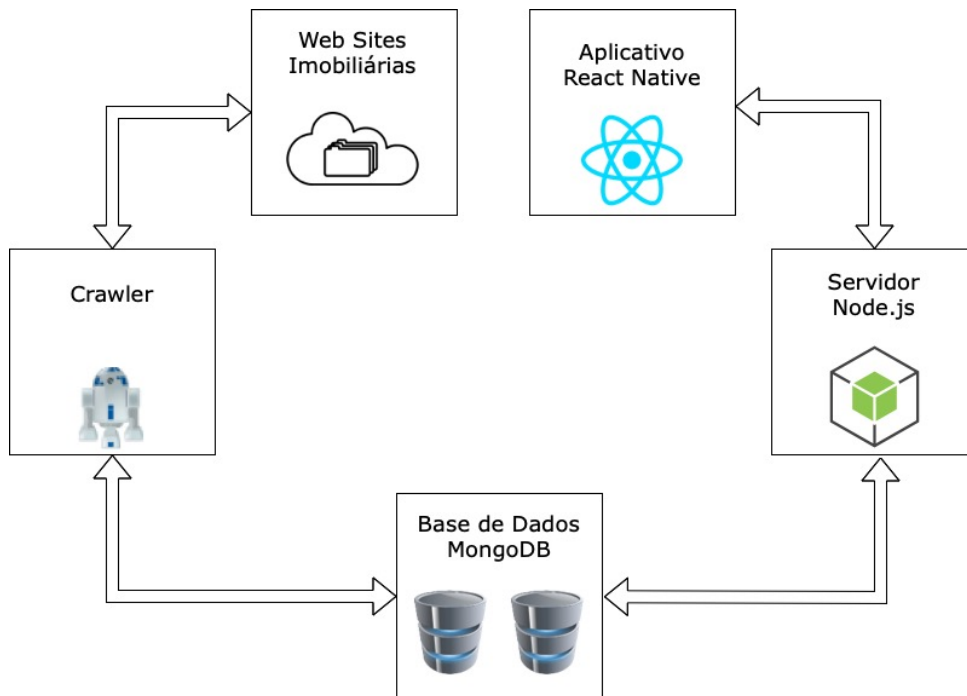


Figura 4.3: Arquitetura Completa

### 4.3 Base de Dados

A base de dados utilizada no presente trabalho foi previamente disponibilizada pela empresa, trabalho proveniente de outros colaboradores que desenvolveram um *crawler* na linguagem *Python* para percorrer os principais *websites* de imobiliárias de Portugal. Os dados obtidos nesse processo são normalizados, a fim de manter um padrão nos dados, como localização e características do imóvel.

Para armazenar todo os dados obtidos pelo *crawler*, foi escolhido o banco de dados MongoDB. A escolha desse banco de dados foi devido à grande quantidade de dados a ser armazenada e por não possuir grandes relacionamentos, fatores que aproveitam as maiores vantagens dos bancos de dados não-relacionais.

Na imagem abaixo temos um exemplo de como cada imóvel está sendo armazenado na base da dados. Nele contém todos dados e características disponível e normalizado para todos os sites de imobiliárias que foram utilizados para obter os dados.

```

{
  "_id": {
    "$oid": "5fd110d7542c2c1e9c090d2b"
  },
  "title": "Apartamento T2 para Venda",
  "photos": [
  ],
  "price": {
    "$numberInt": "85000"
  },
  "typology": "t2",
  "usefulArea": {
    "$numberInt": "93"
  },
  "terrainArea": {
    "$numberInt": "123"
  },
  "otherCharacteristics": [
    "construction year: 2011",
    "energy class: none"
  ],
  "features": null,
  "description": "imovel de banco\n\napartamento t2 em canelas, 1",
  "address": {
  },
  "platformID": "3764B",
  "views": null,
  "creationDate": {
  },
  "updateDate": {
  },
  "contact": null,
  "purpose": "buy",
  "platform": "bpiexpresso",
  "type": "apartamento",
  "lastSeen": {
  },
  "link": "https://www.bpiexpressoimobiliario.pt/apartamento/t2/pc",
  "buyPrice": {
    "$numberInt": "85000"
  },
  "rentPrice": null,
  "toBuy": true,
  "toRent": false,
  "typologyShort": "t2",
  "outliers": {
  }
}

```

Figura 4.4: Exemplo de imóvel na base de dados.

## 4.4 Bibliotecas

Durante o desenvolvimento deste projeto foram utilizadas diversas bibliotecas, de código livre e provenientes do *Node Package Manager* (NPM) que é o gerenciador de pacotes oficial do Node.js. A utilização dessas bibliotecas possibilita o desenvolvedor focar apenas no trabalho essencial e reutilizar funcionalidades prontas que auxiliam no desenvolvimento da solução.

No presente trabalho foram utilizadas diversas bibliotecas provenientes do NPM, segue abaixo as principais bibliotecas utilizadas e uma breve descrição de suas funcionalidades.

- **bcrypt**: é uma biblioteca que auxilia na criptografia de senhas, possuindo funções tanto de transformar a senha em um *hash* como comparar senhas (*Bcrypt* [24]).
- **jsonwebtoken**: também conhecido com JWT, tem a função de criar dados com assinatura criptografada, e com essa assinatura é possível verificar se um usuário que está acessando o sistema realmente está logado e tem suas devidas permissões (*Jsonwebtoken* [25]).
- **react-navigation**: é uma biblioteca que permite que o usuário consiga navegar entre diferentes telas de uma aplicativo (*React-navigation* [26]).
- **babel**: é um compilador para JavaScript, o qual possibilita a utilização de arquivos de diversos formatos sem preocupações extras (*Babel* [27]).
- **intl**: é uma biblioteca que ajuda na formatação de datas, possibilitando uma fácil utilização de diferentes formatos e diferentes fusos horários (*Intl* [28]).
- **SliderBox**: essa biblioteca oferece um componente que possibilita a exibição de várias imagens em um mesmo lugar, sendo necessário apenas o usuário passar para próxima imagem (*Sliderbox* [29]).

Quase todas essas bibliotecas citadas são padrões em projetos do gênero, entretanto, as bibliotecas jsonwebtoken e SliberBox foram decisões de projeto pensadas especificamente para esse projeto. Primeiramente a biblioteca jsonwebtoken é a implementação do

JWT, que é um padrão aberto, definido pela RFC 7519[30]. O JWT define uma maneira compactada e criptografada para transmitir um objeto JSON que será responsável por armazenar dados de autenticação do usuário. Além disso, o JWT é um padrão que já é maduro e amplamente utilizado pela comunidade de desenvolvedores, sendo assim uma ótima escolha para auxiliar na autenticação de usuários.

Por segundo, a biblioteca do SliderBox foi a solução mais próxima encontrada do que o projeto precisava para exibir as imagens dos imóveis. Foram testadas várias bibliotecas com funcionalidades semelhantes, entretanto o SliderBox é o que mais correspondeu as necessidades. Apesar disso, a biblioteca não aceitava uma configuração na forma de mostrar a quantidade de fotos a serem exibidas, então foi necessário incorporar o código da biblioteca e fazer as alterações necessárias diretamente no código fonte dessa biblioteca.

## 4.5 Entregas e Testes

A entrega contínua é um processo que aborda todo fluxo de desenvolvimento de um software, desde o código escrito até a sua implantação no ambiente de produção. Esse processo provê ciclos mais curtos de *feedback* entre desenvolvedores e *stakeholders* afim de entregar um produto com mais eficiência (Krusche e Alperowitz [31]).

A HydraDev, empresa onde o projeto foi desenvolvido, executa um processo de entrega continua adaptado as suas necessidades. São utilizadas algumas medidas internas para prover uma entrega continua e conseqüentemente receber *feedbacks* com mais frequência dos *stakeholders*, que no caso são os gestores da empresa.

As medidas adotadas no projeto para possibilitar a entrega continua foram implementadas de formas distintas no *backend* e *frontend*. Sendo que no *backend* foi utilizado o GitHub para versionar o código e conectá-lo a plataforma Heroku, a qual faz o trabalho de monitorar quando uma versão é submetida e automaticamente atualizar o seu serviço com o código submetido mais recente. Dessa forma, a cada alteração submetida o Heroku consegue prover na mesma URL previamente estabelecida a versão mais recente do projeto.

Na parte do *frontend* também foi utilizado o GitHub para versionar o código e a cada submissão par ao repositório também era publicado uma nova versão no Expo. O Expo permite que qualquer usuário que possua acesso consiga abrir a versão mais recendo do aplicativo, independentemente de estar na mesma rede que o desenvolvedor. Dessa forma é possível que todos ligados ao projeto consigam acompanhar o desenvolvimento de forma remota e simultaneamente com as últimas submissões de código.

Todo código é inicialmente submetido para uma *branch* de desenvolvimento e após passar por uma revisão de código é integrada com a *branch* principal, onde ocorre todo processo de entrega continua explicado anteriormente. Nessa revisão de código é possível prevenir erros e receber comentários de possíveis melhorias no código.

Além dessas revisões de código, acontecia quinzenalmente uma entrega oficial e apresentação das novas funcionalidades implementadas. Nessas entregas eram feitos testes no *backend* e *frontend* pelos gestores, assim como comentários de melhorias.

## 4.6 Aplicativo

A partir dos requisitos solicitados pela empresa que estão dispostos no Capítulo 3, foi desenvolvido o sistema que agrega dados imobiliários.

O sistema permite que usuários consigam listar todos imóveis agregados, dispondo de um filtro com diversas configurações para melhorar os resultados de busca. Também é possível encontrar diversas métricas nos detalhes de cada imóvel ou até mesmo comprar imóveis por suas características. Dessa maneira pode-se perceber que o sistema atende aos requisitos estabelecidos.

Foram encontrados dois principais desafios desenvolvimento desse sistema. O primeiro é em relação a grande quantidade dados que o sistema precisa lidar, dessa forma foi preciso implementar soluções que tivessem melhor performance, tanto em processamento como em memória. Foi utilizado listas através do método *lazy list*, que consiste em renderizar os itens da lista sob demanda. Além disso, foi implementado a paginação no *back-end* e conforme o usuário percorrer a lista, automaticamente o aplicativo realiza requisições

para as próximas páginas.

O segundo principal desafio do sistema foi em relação a quantidade de campos a serem filtrados e comparados. Dessa forma foi preciso padronizar vários campos e ter muita atenção com a grande quantidade de filtros.

## **Demonstração do Aplicativo**

A primeira tela do aplicativo é o *log in* demonstrado na imagem 4.5. Nessa tela é necessário preencher os campos *username* e *password* para acessar o aplicativo ou clicar em *sign up* para se registrar.

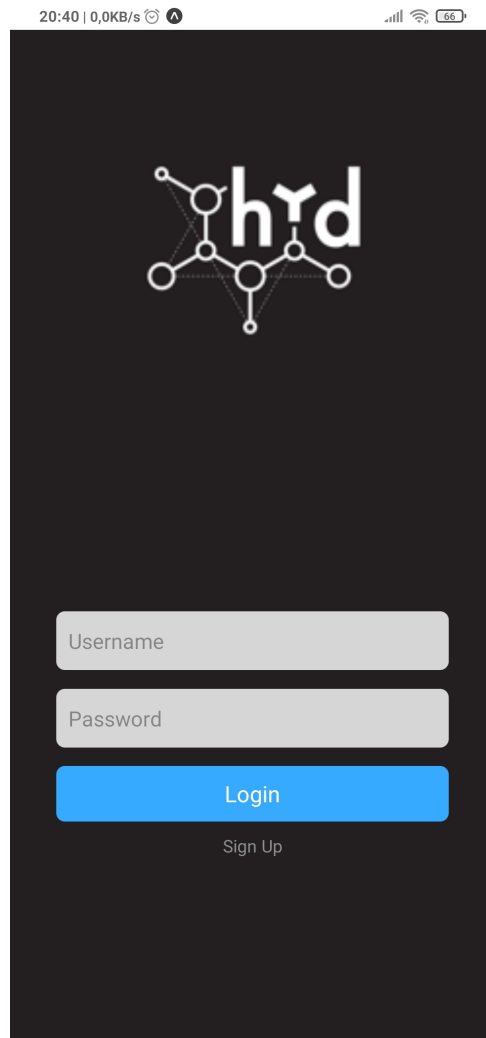


Figura 4.5: Tela de Login.

Após realizar o *log in*, o usuário será automaticamente redirecionado para a tela *home* demonstrada na imagem 4.6. Essa tela permite o acesso a todas as funcionalidades do aplicativo, sendo sempre necessário regressar a essa tela caso queira usar outra funcionalidade.

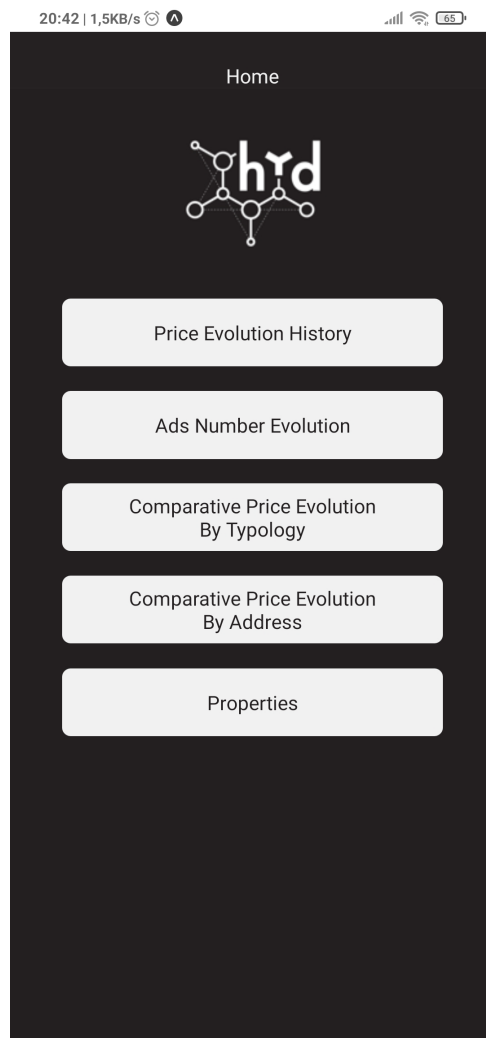


Figura 4.6: Tela *Home*

A primeira funcionalidade do aplicativo é a listagem de propriedades, como podemos ver na imagem 4.7. Nessa tela é preciso preencher o propósito do anúncio, sendo ele para vender ou arrendar. Por segundo tem-se a tipologia, a qual pode ser selecionada uma ou mais e significa a quantidade de quartos que o imóvel possui.

Por terceiro tem-se o tipo do imóvel, isso define a característica da construção do imóvel. Ou seja, se é um apartamento, casa, escritório, etc. Por último tem-se o endereço, onde pode ser selecionado o distrito ou o município.

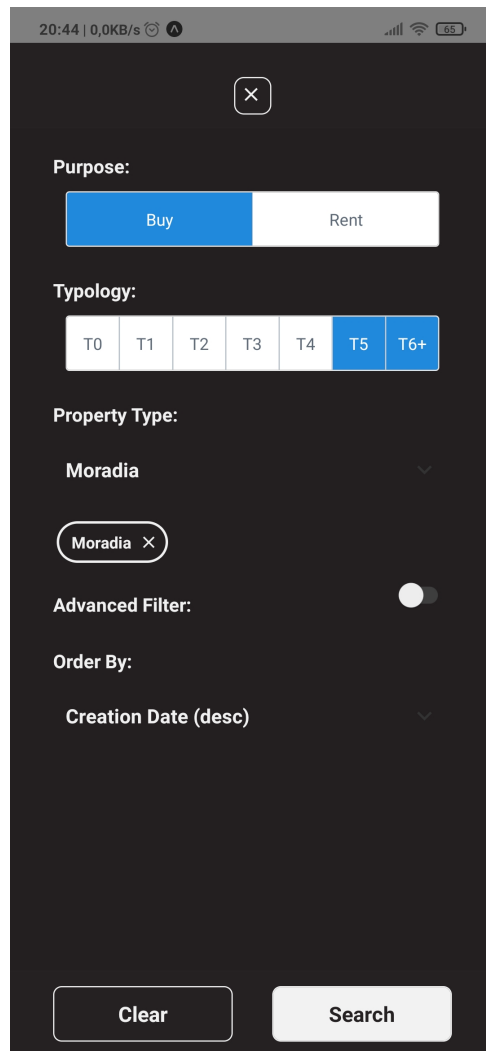


Figura 4.7: Tela filtros propriedade

Ao seleccionar um imóvel para ver seus detalhes, entra-se na tela como mostra a imagem 4.8. Aqui pode-se ver as principais características do anúncio como o título, imagens, propósito, preço, localização, etc.

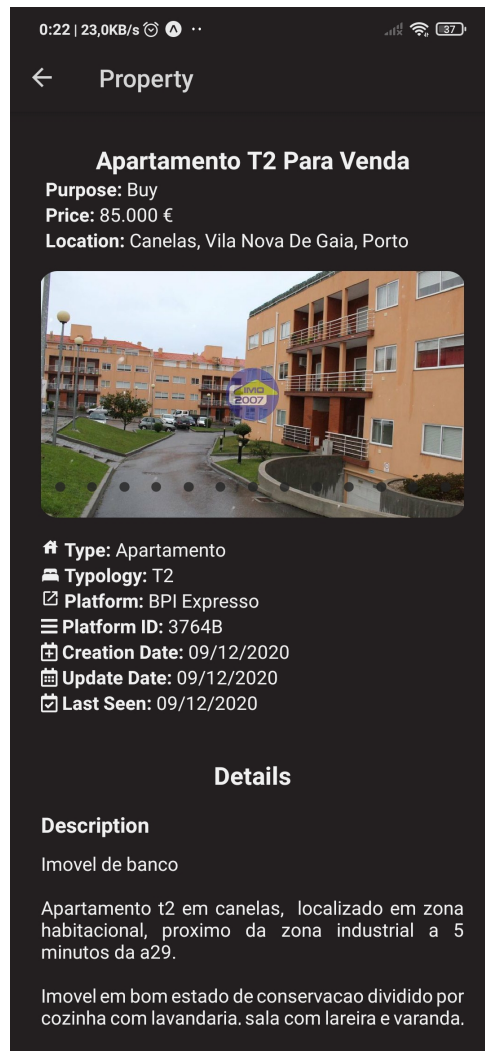


Figura 4.8: Tela Detalhes da Propriedade

Ao seleccionar um imóvel para ver seus detalhes, entra-se na tela como mostra a imagem 4.8 e ao descer essa tela tem-se as estatísticas do imóvel, como mostra a imagem 4.9. Aqui pode-se ver as estatísticas baseadas em valores médios globais da base de dados, é claro, respeitando os filtros indicados em cada estatística.

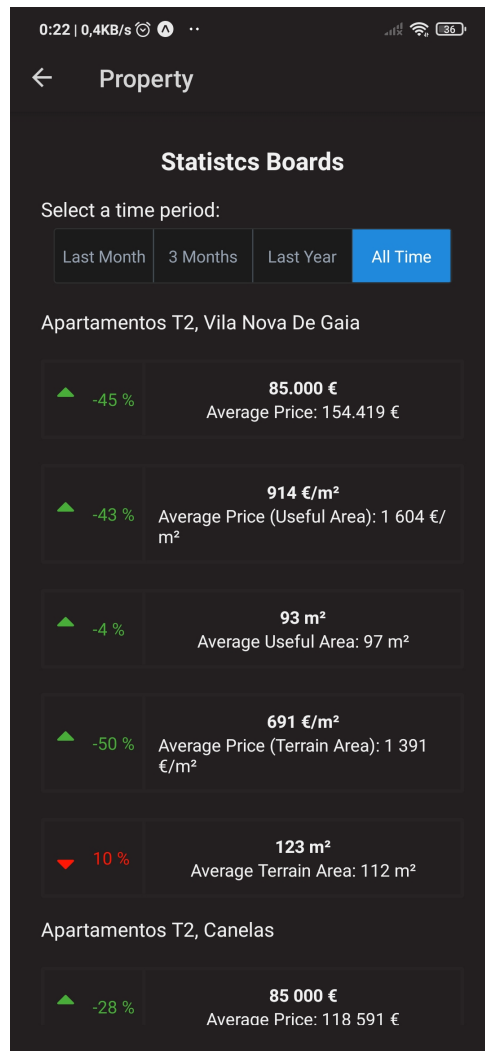


Figura 4.9: Tela Estatísticas da Propriedade

Ao selecionar um imóvel para ver seus detalhes, pode-se ver os dados já descritos anteriormente e também as estatísticas em gráficos como na imagem 4.10. Aqui pode-se ver as estatísticas ao decorrer do tempo de acordo com as características indicados em cada um, como localização.

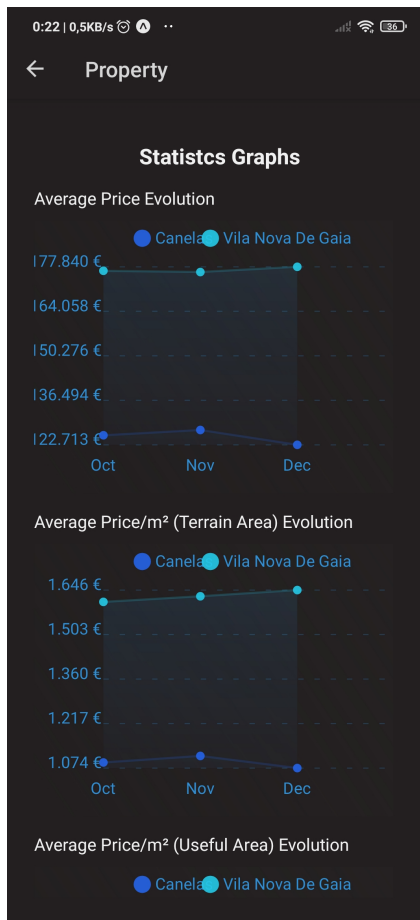


Figura 4.10: Tela Estatísticas Propriedade 2

Além de listar imóveis com vários filtros, também é possível analisar a evolução dos preços dos imóveis mês a mês. Nessa tela do aplicativo, como pode-se ver na imagem 4.11, é possível escolher filtros como propósito, tipologia, tipo e localização do imóvel. Através desses filtros temos um grupo de imóveis e a evolução do preço médio desse grupo mês a mês.

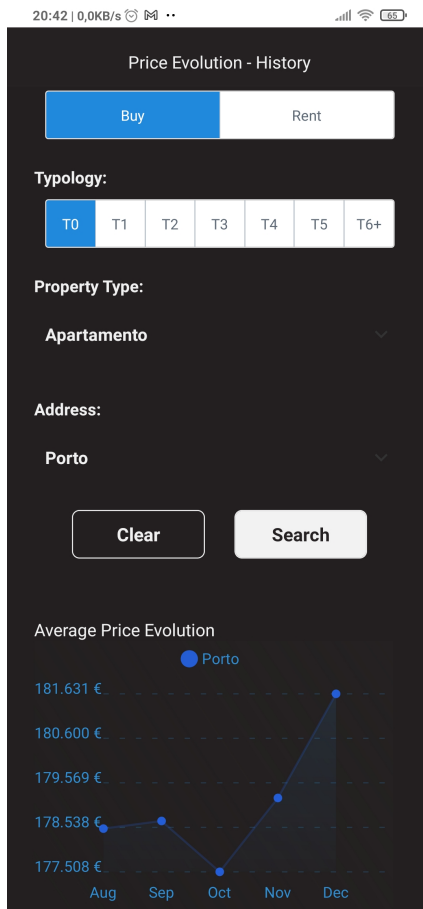


Figura 4.11: Tela de Evolução dos Preços

De uma forma muito semelhante à tela anterior, utilizando os mesmos filtros é possível analisar a quantidade de anúncios ativos mês a mês, assim como a tela na imagem 4.12 demonstra.

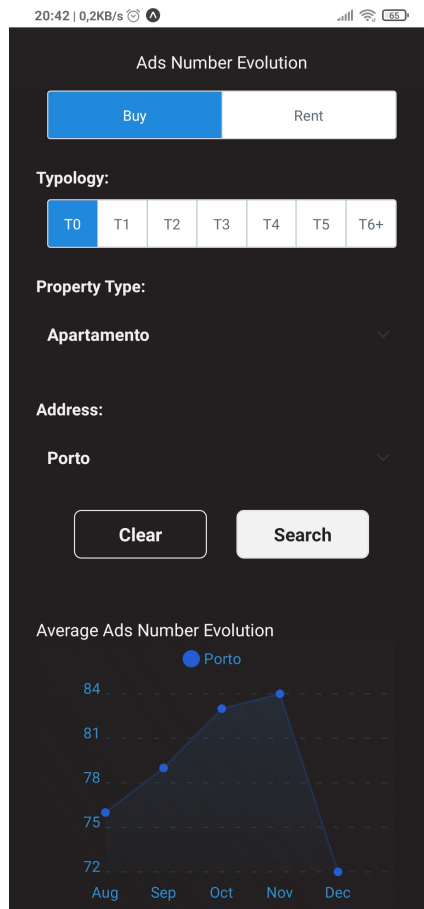


Figura 4.12: Tela de Evolução do Número de Anúncios

A partir dos dados de evolução dos preços também é possível comparar classes de imóveis. Como a tela na imagem 4.13 demonstra, é possível comparar a mesma classe de imóveis, porém com diferentes tipologias. Ou seja, é possível comparar o preço de imóveis de uma determinada região com diferentes números de quartos, e assim analisar em decorrência do tempo qual foi o comportamento desses preços.

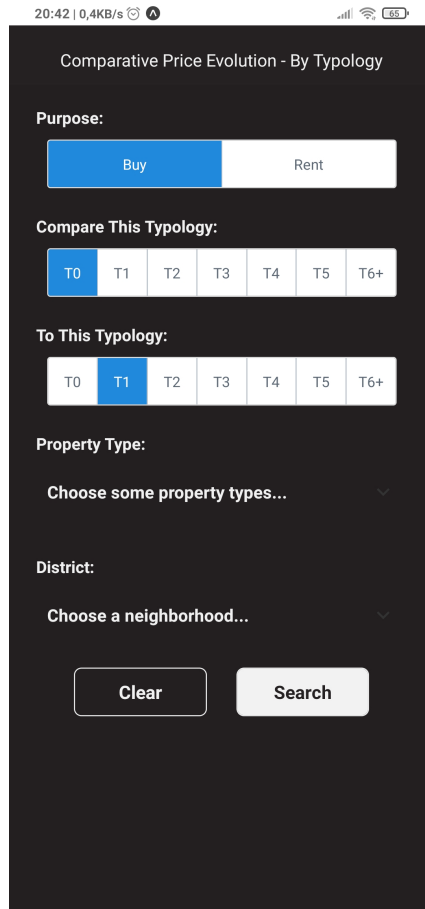


Figura 4.13: Tela Comparativo Preços 1

A tela da imagem 4.14 é a continuação da tela anterior, onde é possível visualizar o resultado de uma das buscas possíveis.

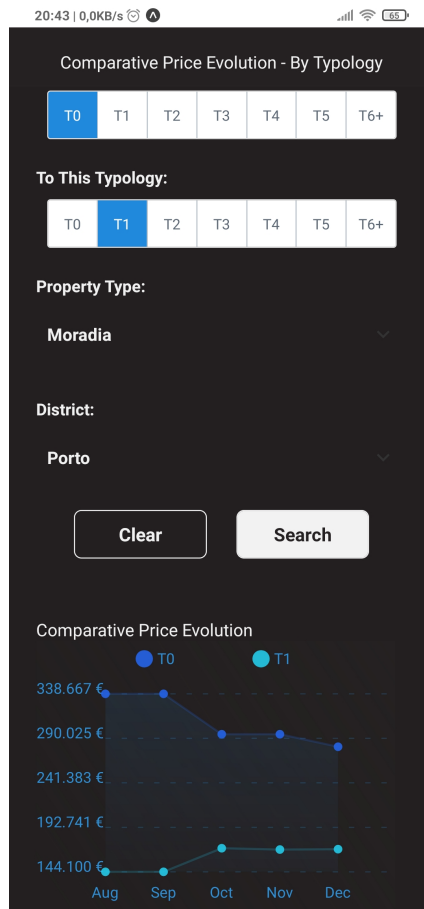


Figura 4.14: Tela Comparativo Preços 2

O aplicativo também possui outro comparativo de preços, no qual é possível comparar classes de imóveis de diferentes localizações. A tela é muito semelhante à imagem 4.13, onde apenas é alterado o filtro para comparar diferentes localizações. Essa funcionalidade permite analisar o preço do imóvel em decorrência do tempo, comparando propriedades de localizações distintas, como por exemplo, verificar que o preço médio de imóveis de um bairro está subindo tanto quanto um outro bairro nobre.

# Capítulo 5

## Conclusões e Trabalhos Futuros

A tecnologia veio para integrar com todas as áreas e proporcionar um ambiente cada vez mais eficiente, entretanto algumas áreas abstêm-se da tecnologia. Podemos ver isso claramente no mercado imobiliário que demora a incorporar tecnologias no seu dia a dia. Dessa forma, existe um mercado a ser preenchido, no qual pode-se aproveitar de meios tecnológicos para gerar novas e eficientes soluções para o mercado imobiliário.

O mercado imobiliário em Portugal vem crescendo muito nos últimos anos, agregando esse crescimento a lacuna tecnológica que essa área tem, surge a proposta que o presente trabalho visa resolver. Ou seja, proporcionar dados de qualidade e que não são fáceis de encontrar sobre o mercado para quem trabalha diretamente com esse setor.

Para entregar essa solução ao mercado foi desenvolvido uma aplicação mobile em React Native, a qual pode ser executada tanto em Android como iOS. Já a API REST foi implementada com JavaScript em um servidor Node.js que está hospedado no serviço Heroku. Por fim, utilizou-se o banco de dados MongoDB para armazenar os dados.

Como trabalhos futuros são recomendados:

- Novos comparadores de acordo com as necessidades encontradas por usuários.
- Implementação de uma versão web com React.js que possibilite novas métricas e análises mais complexas.
- Alterar o *crawler* para cobrir mais *websites* de imobiliárias.

# Bibliografia

- [1] J. M. S. d. O. Dias, «O Setor Imobiliário na era digital: o caso da Brixel,» rel. téc., jun. de 2020.
- [2] M. Silva, *JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript*. Novatec Editora, 2020, ISBN: 9788575227947. URL: <https://books.google.pt/books?id=6DfnDwAAQBAJ>.
- [3] E. Brown, *Web Development with Node and Express: Leveraging the JavaScript Stack*. O'Reilly Media, 2019, ISBN: 9781492053460. URL: <https://books.google.pt/books?id=-Dq-DwAAQBAJ>.
- [4] Node, «Node. js,» *OpenJS Foundation*, 2020. URL: [https://scholar.google.com/scholar?hl=pt-PT&as\\_sdt=0%5C%2C5&q=NODE.JS.+Sobre+Node.js.+OpenJS.+Foundation%5C%2C+2020.&btnG=](https://scholar.google.com/scholar?hl=pt-PT&as_sdt=0%5C%2C5&q=NODE.JS.+Sobre+Node.js.+OpenJS.+Foundation%5C%2C+2020.&btnG=).
- [5] E. Masiello e J. Friedmann, *Mastering React Native*. Packt Publishing, 2017, ISBN: 9781785889790. URL: <https://books.google.pt/books?id=A1QoDwAAQBAJ>.
- [6] J. A. Lundar, T.-M. Gronli e G. Ghinea, «Performance evaluation of a modern web architecture,» *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 8, n.º 1, pp. 36–50, 2013.
- [7] C. A. Alperstedt, C. R. d. Rolt e G. D. Alperstedt, «Acessibilidade e tecnologia na construção da cidade inteligente,» *Revista de Administração Contemporânea*, vol. 22, pp. 291–310, 2018.

- [8] E. Brown, *Web development with node and express: leveraging the JavaScript stack*. O'Reilly Media, 2019.
- [9] A. Horton e R. Vice, *Mastering React*. Packt Publishing, 2016.
- [10] A. Boduch, *React and React Native*. Livery Place, # 35 Livery Street, Birmingham, 2017.
- [11] V. Antani, S. Timms e D. Mantyla, *JavaScript: Functional Programming for JavaScript Developers*. Packt Publishing Ltd, 2016.
- [12] A. Paul e A. Nalwaya, «React Native for Mobile Development,» *React Native for Mobile Development*. California: Apress, Berkeley, CA. <https://doi.org/10.1007/978-1-4842-4454-8>, 2019.
- [13] K. Banker, D. Garrett, P. Bakkum e S. Verch, *MongoDB in Action: Covers MongoDB version 3.0*. Simon e Schuster, 2016.
- [14] M. A. Siby Anusha, «Android Hacking Using Msfvenom: Integrating NGROK,»
- [15] N. Middleton e R. Schneeman, *Heroku: up and running: effortless application deployment and scaling*. "O'Reilly Media, Inc.", 2013.
- [16] Aníbal, *Novo alerta de bolha especulativa imobiliária em Portugal*. *Jornal Público*. Acessado: 2021-09-25.
- [17] Sousa, *Mercado imobiliário em Portugal dá sinais de abrandamento e “futuro incerto”*. *Jornal Económico*, Acessado: 2021-09-25.
- [18] Andrade, *Porque é que os preços das casas não caem apesar da pandemia?* *Jornal Expresso*, 2020. Acessado: 2021-09-25.
- [19] L. A. d. M. Pereira, «Análise e modelagem de sistemas com a UML: com dicas e exercícios resolvidos.,» rel. téc., jan. de 2011.
- [20] S. Chacon e B. Straub, *Pro Git*. Apress, 2014, ISBN: 9781484200773. URL: <https://books.google.pt/books?id=KhW8oAEACAAJ>.

- [21] J. Ingeno, *Software Architect's Handbook: Become a successful software architect by implementing effective architecture concepts*. Packt Publishing, 2018, ISBN: 9781788627672. URL: <https://books.google.pt/books?id=6EZsDwAAQBAJ>.
- [22] *About GitHub*, <https://github.com/about>, Acessado: 2021-09-30.
- [23] *About Trello*, <https://trello.com/about>, Acessado: 2021-10-03.
- [24] *Bcrypt*, <https://www.npmjs.com/package/bcrypt>, Acessado: 2021-10-05.
- [25] *Jsonwebtoken*, <https://www.npmjs.com/package/jsonwebtoken>, Acessado: 2021-10-05.
- [26] *React-navigation*, <https://www.npmjs.com/package/react-navigation>, Acessado: 2021-10-05.
- [27] *Babel*, <https://www.npmjs.com/package/Babel>, Acessado: 2021-10-05.
- [28] *Intl*, <https://www.npmjs.com/package/intl>, Acessado: 2021-10-05.
- [29] *Sliderbox*, <https://www.npmjs.com/package/react-native-image-slider-box>, Acessado: 2021-10-05.
- [30] *jwt*, <https://datatracker.ietf.org/doc/html/rfc7519>, Acessado: 2021-10-15.
- [31] S. Krusche e L. Alperowitz, «Introduction of continuous delivery in multi-customer project courses,» em *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 335–343.

# Apêndice A

## Proposta Original do Projeto

## Mestrado em Informática

### Programa e supervisão do Estágio

1.- O Estágio a realizar é constituído pelo seguinte programa de trabalho e cronograma:

A HydraDev, Lda está a trabalhar num projecto de R&D que visa agregar dados imobiliários de várias fontes e permitir não só pesquisas mas também análises de mercado complexas de forma intuitiva para um comprador ou vendedor individual. Actualmente, a componente de angariação de dados está já em funcionamento pelo que é possível passar às fases seguintes em que se inclui o presente trabalho.

Com este trabalho pretende-se implementar uma aplicação móvel com duas vertentes principais. Uma prende-se com pesquisas avançadas dos dados imobiliários já pré-processados. A outra prende-se com a apresentação de métricas relativas ao mercado imobiliário como um todo, ou a nichos específicos. Tanto uma vertente como a outra, obrigam ao desenvolvimento duma interface de fácil utilização mas grande versatilidade, com elementos de visualização de dados, e que dado o volume de dados terá de ser eficiente.

Para acelerar o desenvolvimento duma primeira iteração da aplicação móvel, utilizar-se-á React Native, podendo assim abranger Android e iOS com uma só base de código.

Considera-se que este trabalho é de alta relevância para um aluno a concluir um mestrado em informática. Por um lado, há uma componente forte de engenharia de software dado que será preciso começar o desenvolvimento da aplicação de raiz, definindo a sua arquitectura e implementando-a tendo em conta os objectivos do projecto. Por outro, há uma componente R&D dado o contexto do projecto e dos elementos de visualização de dados que serão necessários. Note-se ainda que o desenvolvimento em React Native, uma tecnologia de ponta com grande procura neste momento, é também uma mais-valia para o aluno.