

Helbert Alves Monteiro da Silva

**Um Aplicativo para auxiliar estudantes a
encontrar e dividir aluguel na cidade de
Picos-PI**

Picos - PI
Dezembro de 2019

Helbert Alves Monteiro da Silva

Um Aplicativo para auxiliar estudantes a encontrar e dividir aluguel na cidade de Picos-PI

Monografia submetida ao curso de Bacharelado em Sistemas de Informação como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação na Universidade Federal do Piauí. Orientador: Prof. Ismael de Holanda Leal

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
Dezembro de 2019

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

S725m Silva, Helbert Alves Monteiro da.
Um aplicativo para auxiliar estudantes a encontrar e dividir aluguel na cidade de Picos-PI. / Helbert Alves Monteiro da Silva. -- Picos,PI, 2019.
65 f.
CD-ROM: 4 ¾ pol.

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação. – Universidade Federal do Piauí, Picos, 2019.
“Orientador(A): Prof. Dr. Ismael de Holanda Leal.”

1. Aplicativo Móvel – Aluguel Imobiliário. 2. Android. 3. Web Service. I. Título.

CDD 005.2

Elaborada por Rafael Gomes de Sousa CRB 3/1163

UM APLICATIVO PARA AUXILIAR ESTUDANTES A ENCONTRAR E DIVIDIR ALUGUEL
NA CIDADE DE PICOS-PI

HELBERT ALVES MONTEIRO DA SILVA

Monografia APROVADA como exigência parcial para obtenção do
grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 02 de DEZEMBRO de 2019


Prof. Ismael de Holanda Leal


Prof. Frank César Lopes Vêras


Prof. Alan Rafael Ferreira dos Santos

Agradecimentos

A Deus, pela capacidade a mim concebida e por todas as bênçãos a mim dadas, sem Ele nada seria possível.

Aos meus pais, José Nilton e Maria do Ó, pelo amor incondicional e por todo apoio e dedicação a me proporcionarem recursos e oportunidades que contribuem para o meu crescimento, seja educacional ou em qualquer área da minha vida, me incentivando e me apoiando sempre a ser uma pessoa melhor. Eu amo vocês.

Aos meus irmãos, José Mário e Vanessa e minha cunhada Raquel, que sempre estiveram ao meu lado e me ajudaram de alguma e de todas as formas, apoiando e cuidando de mim.

À minha namorada Taís, por todo o bem e felicidade que me causa, pelo amor e paciência, pela cumplicidade, pelo carinho, pelo apoio e incentivo nos meus projetos e por todos os momentos especiais juntos. Você é muito especial para mim.

À Universidade Federal do Piauí, pelo compromisso em prover estrutura e recursos que possibilitam a seus acadêmicos alcançarem uma graduação de ensino superior com bom reconhecimento e respeito.

Aos meus professores da graduação, pela dedicação em transmitir conhecimento da melhor forma, exigindo empenho e orientando a seguir os caminhos corretos da vida acadêmica e profissional. Sem vocês este trabalho não teria tamanha riqueza.

Aos meus professores do primário ao ensino médio, que cultivaram em mim a sede de conhecimento e me proporcionaram uma base educacional memorável.

Aos meus amigos, pelos momentos de alegria, pelas admirações que expõem sobre mim que de certa forma me incentivam a ser tão bom como me enxergam, admiráveis também são vocês que transmitem paz e felicidade através de uma convivência de grande prestígio.

Aos colegas de graduação, que estão sempre dispostos a ajudar e contribuir para o êxito das atividades acadêmicas.

A todos os familiares e demais que de alguma forma contribuíram para a minha formação.

“Lembre de Deus em tudo o que fizer, e Ele lhe mostrará o caminho certo”
(Provérbios 3:6)

“Confia ao Senhor as tuas obras, e os teus desígnios serão estabelecidos.”
(Provérbios 16:3)

Resumo

Procurar um imóvel para alugar, geralmente, é uma tarefa cansativa quando se deseja encontrar o imóvel ideal que atenda aos requisitos preferíveis, tais como: custo mensal, localização, número de quartos, banheiros, entre outras informações. Esta tarefa se torna, além de cansativa, mais difícil quando o interesse é em outra cidade e não se tem uma ferramenta que auxilie com eficiência e agilidade a busca destes imóveis com disponibilidade para locação, já informando características relevantes para análise de qual imóvel atende melhor as condições desejadas. Com o avanço da tecnologia, certas tarefas se tornaram mais fáceis, por exemplo, com o uso de dispositivos móveis como um *smartphone* conectado à rede mundial de computadores, a *internet*, é possível alcançar informações e executar ações de qualquer lugar do mundo em qualquer sistema que assim permita realizar tais atividades. Partindo do problema e deste meio de facilidade citados acima, nasceu a ideia de desenvolver um aplicativo móvel para a plataforma *Android* para auxiliar os estudantes que migram para a cidade de Picos, no Piauí, a encontrarem sua nova moradia. É notável também a grande quantidade de alunos que buscam dividir as despesas dos aluguéis de onde moram, espalhando anúncios em redes sociais e murais nas universidades e colégios, buscando outros inquilinos para o imóvel que já alugam. O aplicativo aqui apresentado também auxilia estes estudantes que buscam dividir o aluguel dos imóveis com outras pessoas, disponibilizando além das informações dos imóveis, as informações dos seus perfis com o intuito de que encontrem as pessoas que se encaixam melhor com o seu perfil, possibilitando, assim, mais chances de uma boa convivência.

Palavras-chaves: *Android*. Aplicativo Móvel. Aluguel. *Web Service*.

Abstract

Search for a immobile to rent it is usually a tiring task when is desired find the ideal immobile, that meets the preferables requirements, suches as: monthly cost, location, number of bedrooms, bathrooms, among other informations. This task becomes beyond besides tiring, more difficult when the interest is in another city and don't have a tool that help with efficiency and agility the search these immobiles with disponibilitate for lease, already informing relevant characteristics for analyse of which immobile cater better the desired conditions. With the advance of technology, certain tasks have become easier, for example, with the use of mobile devices like a smartphone connected to the world wide web, the internet, it is possible to reach informations and perform action from anywhere in the world on any system that permits such activities. Starting from the problem and of this means of ease mentioned above, was born the idea of develop a mobile app for the Android platform to aid the students that migrate to the city of Picos, in Piauí, the find your new home. It is remarkable too the great amount of students that seek to divide the expenses of the rents of where they live, spreading ads on social networks and murals in universities and colleges, searching for others tenants for the immobile that already rent. That app here presented also aid these students that seek to divide the rent of the immobiles with others people, availabling beyond os the informations of immobiles, the informations of the your profiles with the intent of that they find the people that best fit with your profile, thus enabling more chances of the a good coexistence.

Keywords: *Android. Mobile App. Rent. Web Service.*

Lista de ilustrações

Figura 1	– <i>Ranking dos Sistemas Operacionais mobile no Brasil</i>	17
Figura 2	– <i>Arquitetura da plataforma Android</i>	17
Figura 3	– <i>Serviços oferecidos pelo Firebase</i>	21
Figura 4	– <i>Ranking das linguagens de programação mais populares no mundo em 2019.</i>	23
Figura 5	– <i>Ciclo de requisição em um Web Service RESTful.</i>	24
Figura 6	– <i>Diagrama de Casos de Uso.</i>	33
Figura 7	– <i>Diagrama Entidade Relacionamento.</i>	36
Figura 8	– <i>Tela Login.</i>	39
Figura 9	– <i>Tela Cadastro.</i>	40
Figura 10	– <i>Tela Redefinir Senha.</i>	41
Figura 11	– <i>Tela Inicial.</i>	42
Figura 12	– <i>Tela Perfil.</i>	43
Figura 13	– <i>Tela Lista.</i>	44
Figura 14	– <i>Tela Mapa.</i>	45
Figura 15	– <i>Tela Favoritos.</i>	46
Figura 16	– <i>Tela Interesses.</i>	47
Figura 17	– <i>Tela Meus Anúncios.</i>	48
Figura 18	– <i>Tela Busca.</i>	49
Figura 19	– <i>Tela Cadastro de Imóvel.</i>	50
Figura 20	– <i>Tela Gerência de Imóvel.</i>	51
Figura 21	– <i>Tela Imóvel.</i>	52
Figura 22	– <i>Perfil de outro usuário e Contato.</i>	53
Figura 23	– <i>Resposta da questão 1 do questionário de avaliação de usabilidade.</i>	57
Figura 24	– <i>Resposta da questão 2 do questionário de avaliação de usabilidade.</i>	58
Figura 25	– <i>Resposta da questão 3 do questionário de avaliação de usabilidade.</i>	58
Figura 26	– <i>Resposta da questão 4 do questionário de avaliação de usabilidade.</i>	58
Figura 27	– <i>Resposta da questão 5 do questionário de avaliação de usabilidade.</i>	59
Figura 28	– <i>Resposta da questão 6 do questionário de avaliação de usabilidade.</i>	59
Figura 29	– <i>Resposta da questão 7 do questionário de avaliação de usabilidade.</i>	59
Figura 30	– <i>Resposta da questão 8 do questionário de avaliação de usabilidade.</i>	60
Figura 31	– <i>Resposta da questão 9 do questionário de avaliação de usabilidade.</i>	60
Figura 32	– <i>Resposta da questão 10 do questionário de avaliação de usabilidade.</i> . . .	60
Figura 33	– <i>Pontuação da avaliação dos usuários.</i>	61
Figura 34	– <i>Resultado da avaliação.</i>	62

Lista de tabelas

Tabela 1 – <i>Linguagem SQL: Agrupamentos e seus comandos</i>	26
Tabela 2 – <i>Análise comparativa entre os trabalhos relacionados e o trabalho proposto</i>	30

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
IU	<i>Interface de Usuário</i>
SMS	<i>Short Message Service</i>
app	<i>Application</i>
SO	Sistema Operacional
SDK	<i>Software Development Kit</i>
Bass	<i>Backend as a Service</i>
JDK	<i>Java Development Kit</i>
JRE	<i>Java Runtime Environment</i>
HTTP	<i>Hypertext Transfer Protocol</i>
XML	<i>Extensible Markup Language</i>
SOAP	<i>Simple Object Access Protocol</i>
IDE	<i>Integrated Development Environment</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
SQL	<i>Structured Query Language</i>
ANSI	<i>American National Standards Institute</i>
DBMS	<i>Database Management System</i>
C2C	<i>Consumer to Consumer</i>
DER	Diagrama Entidade Relacionamento
URL	<i>Uniform Resource Locator</i>
MD5	<i>Message-Digest algorithm 5</i>
JSON	<i>JavaScript Object Notation</i>
SUS	<i>System Usability Scale</i>
UFPI	Universidade Federal do Piauí

Sumário

1	Introdução	13
1.1	Objetivos	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	15
1.2	Organização do Trabalho	15
2	Revisão Bibliográfica	16
2.1	<i>Android</i>	16
2.1.1	Arquitetura da plataforma <i>Android</i>	17
2.1.1.1	<i>Kernel do Linux</i>	18
2.1.1.2	Camada de Abstração de <i>Hardware</i>	18
2.1.1.3	<i>Android Runtime</i>	18
2.1.1.4	Bibliotecas C/C++ Nativas	19
2.1.1.5	Java API <i>Framework</i>	19
2.1.1.6	Aplicativos do Sistema	19
2.1.2	Segurança do Sistema <i>Android</i>	20
2.2	<i>Android Studio</i>	20
2.2.1	<i>Android SDK</i>	21
2.3	<i>Firebase</i>	21
2.3.1	<i>Firebase Storage</i>	22
2.4	Java	22
2.5	<i>Web Service</i>	24
2.5.1	<i>Web Service RESTful</i>	24
2.6	NetBeans	25
2.7	Banco de Dados	25
2.8	SQL	26
2.9	MySQL	27
2.10	<i>Google Maps API</i>	27
2.10.1	SDK do <i>Google Maps</i> para <i>Android</i>	27
3	Trabalhos Relacionados	29
3.1	Descrição dos Trabalhos	29
3.1.1	Imóveis Estudantes - Aplicativo Facebook	29
3.1.2	Viva Real	29
3.1.3	ZAP Imóveis	29
3.1.4	Incasa Picos	30

3.2	Análise Comparativa	30
4	Modelagem e Desenvolvimento do Sistema	32
4.1	Modelagem do Sistema	32
4.1.1	Diagrama de Casos de Uso	32
4.1.2	Diagrama Entidade Relacionamento	36
4.2	Desenvolvimento do Sistema	38
4.2.1	Aplicativo	38
4.2.1.1	Tela Login	39
4.2.1.2	Tela Cadastro	40
4.2.1.3	Tela Redefinir Senha	41
4.2.1.4	Tela Inicial	42
4.2.1.5	Tela Perfil	43
4.2.1.6	Tela Lista	44
4.2.1.7	Tela Mapa	45
4.2.1.8	Tela Favoritos	46
4.2.1.9	Tela Interesses	47
4.2.1.10	Tela Meus Anúncios	48
4.2.1.11	Tela Busca	49
4.2.1.12	Tela Cadastro de Imóvel	50
4.2.1.13	Tela Gerência de Imóvel	51
4.2.1.14	Tela Imóvel	52
4.2.1.15	Perfil de Outros Usuários e Contato	53
4.2.2	Banco de Dados	54
4.2.3	Web Service	54
5	Avaliação de Usabilidade	56
5.1	Testes Realizados	56
5.2	Resultados Obtidos	57
6	Considerações Finais	63
6.1	Trabalhos Futuros	63
	Referências	64

1 Introdução

O avanço da tecnologia, a modernização e a facilidade de acesso aos meios de comunicação têm melhorado bastante a vida das pessoas, possibilitando que elas consigam realizar tarefas e resolver problemas do dia-a-dia com mais praticidade através destas tecnologias. Um dos meios mais usados são os aplicativos móveis desenvolvidos para as mais diversas áreas, executados em dispositivos pequenos e de fácil transporte, podendo estar conectados à *internet* ampliando o alcance e facilidade de resolver tais problemas de qualquer lugar com mais comodidade.

Aplicativos móveis são *softwares* desenvolvidos para dispositivos móveis, como *smartphones* e *tablets*, por exemplo. O uso dos dispositivos móveis tem crescido rapidamente em todo o mundo nos últimos anos. De acordo com uma pesquisa realizada¹ pela Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas, a FGV-EAESP², que divulga anualmente resultados de estudos e pesquisas no uso de Tecnologia da Informação nas empresas, em maio de 2019 o uso de *smartphones* pela população brasileira foi de 230 milhões, e se adicionado o uso de *notebooks* e *tablets*, foram 324 milhões de dispositivos, cerca de 1,6 dispositivos por habitante. Esse cenário também tem favorecido o crescimento do número de aplicativos móveis desenvolvidos por empresas, instituições de ensino e, também, pessoas autônomas. Esses aplicativos, quando aliados ao uso da *internet*, podem se tornar ferramentas poderosas e atender problemas nas mais diversas áreas para os mais diversos públicos.

No ano de 2016, na cidade de Picos no Piauí, segundo dados informados pela administração do Campus Senador Helvídio Nunes de Barros da UFPI, ingressaram um total de 841 novos alunos nesta instituição, sendo apenas 199 da cidade de Picos. Os outros 642 vieram de outras cidades e outros estados e a grande maioria, provavelmente, sem nenhum parente ou pessoa conhecida morando nesta cidade. Com estes fatores, os alunos que migram para Picos encontram dificuldades na busca de uma nova moradia. A tarefa de buscar um novo imóvel para morar é cansativa e muitas vezes estressante quando se deseja encontrar o imóvel ideal para suas condições financeiras e de conforto, necessitando andar pela cidade por vários lugares, efetuando ligações para adquirir diversas informações dos imóveis, agendando visitas para ver a estrutura e condições destes imóveis, entre outros fatores.

Nota-se que a grande maioria opta por alugar um apartamento ou uma casa, levando em conta que, a princípio, só permanecerão na cidade enquanto cursam o ensino superior, descartando a ideia de comprar um imóvel, já que não morarão na cidade por mais tempo que o da sua graduação. Há também aqueles que preferem morar com outras pessoas

¹ https://eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2019fgvcia_2019.pdf

² <https://eaesp.fgv.br>

a fim de dividir as despesas do aluguel, entre outras despesas do imóvel. Para encontrar pessoas que também desejam dividir o imóvel e suas despesas, os alunos espalham anúncios nos murais do campus e em redes sociais, porém, estes meios dificilmente atingem todos os interessados, visto que poucas pessoas observam os murais da universidade e nem todos tem algum vínculo nessas redes sociais, enfraquecendo estas formas de comunicação deixando os anúncios dispersos e ineficientes para o devido fim de encontrar facilmente e de forma ágil os interessados.

Com isso, o presente trabalho tem por finalidade o desenvolvimento de um aplicativo móvel para a plataforma *Android*, que é o Sistema Operacional³ presente na maioria dos dispositivos móveis em todo o mundo, buscando, assim, atender à necessidade deste público.

Com o desenvolvimento deste aplicativo, é possível auxiliar os estudantes na procura destes imóveis, tanto para alugar, como também dividir com algum locatário. Sendo um aplicativo de busca online, todos os anúncios se concentrarão em uma única ferramenta disponibilizando todas as informações relevantes para a análise, como localização, preço, os cômodos existentes e número de vagas. Nos casos de buscas para dividir, é possível ver um perfil básico do locatário anunciante, assim como também, o anunciante pode observar o perfil dos usuários que se interessarem em dividir o aluguel com ele, possibilitando assim, a análise não só das informações dos imóveis, mas também das pessoas interessadas, tornando todo o processo prático e eficiente. O aplicativo também facilita a conversação entre estes usuários através de outros canais de comunicação.

Desta forma, o aplicativo pode auxiliar não só os novos alunos que migram para a cidade, mas também aqueles que desejam mudar de imóvel, assim como também todas as outras pessoas que tem estas mesmas finalidades e interesses.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste trabalho é o desenvolvimento de um aplicativo móvel, voltado para a plataforma *Android*, como uma ferramenta para auxiliar estudantes de forma eficiente na oferta e procura de imóveis para alugar e dividir aluguel na cidade de Picos, no Piauí, intermediando o contato entre locadores e locatários e entre os locatários interessados em dividir os imóveis, com informações de contato para facilitar a comunicação entre os mesmos, além de informações de perfis, a fim de encontrarem pessoas que sejam mais compatíveis entre si para um melhor convívio.

³ Um programa ou conjunto de programas que tem a função de gerenciar o sistema do dispositivo que está em execução.

1.1.2 Objetivos Específicos

1. Estudar as tecnologias que serão utilizadas para a construção do projeto e desenvolvimento do aplicativo móvel;
2. Desenvolver o aplicativo que auxiliará estudantes nas ofertas e procuras de imóveis para alugar e dividir aluguéis;
3. Desenvolver o *web service* que fará a troca de informações do aplicativo com a base de dados pela *web*⁴;
4. Realizar testes para avaliar a usabilidade do aplicativo.

1.2 Organização do Trabalho

Este trabalho está organizado em 6 (seis) capítulos. O Primeiro apresenta uma introdução e os objetivos que o trabalho visa alcançar. No capítulo 2 (dois) é apresentada a revisão bibliográfica com um embasamento para a compreensão das tecnologias utilizadas no desenvolvimento deste trabalho. No capítulo 3 (três) são apresentados os trabalhos relacionados, fazendo uma análise comparativa entre os mesmos e esta proposta. O capítulo 4 (quarto) apresenta a modelagem e como o sistema foi desenvolvido, descrevendo seu funcionamento e os recursos utilizados. No capítulo 5 (cinco) encontram-se os testes realizados, avaliando a usabilidade do sistema e apresentando os resultados obtidos. O capítulo 6 (seis) apresenta as considerações finais e os trabalhos futuros.

⁴ Pode ter a mesma definição de *internet*, a rede mundial de computadores, capaz de interligar inúmeros sistemas.

2 Revisão Bibliográfica

O desenvolvimento de uma aplicação móvel voltada para a plataforma *Android* pode utilizar diversas tecnologias em sua execução, principalmente quando a aplicação adota o conceito de Sistemas Distribuídos¹ e necessita da *internet* para o seu funcionamento. Este capítulo expõe a fundamentação teórica das tecnologias que foram usadas para o desenvolvimento deste trabalho.

2.1 *Android*

O *Android* é o Sistema Operacional mais utilizado nos dispositivos móveis em todo o mundo, principalmente em *smartphones* e *tablets* (ZURIARRAIN, 2017). Criado em 2003 na Califórnia, desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White, atualmente é propriedade da *Google*². Sua criação teve grande avanço quando em 2007, a *Google*, juntamente com outras empresas fundaram a *Open Handset Alliance*³, com o objetivo de desenvolver uma plataforma de código aberto⁴ para *smartphones*.

O *Android*, por ser baseado no núcleo *Linux*⁵ e ser uma plataforma de código aberto, possibilita uma grande variedade de personalização, permitindo que empresas fabricantes de dispositivos que usam este sistema tenham a liberdade de modificar o design dele para suas marcas, como uma tentativa de chamar mais atenção para seus produtos e trazer uma ideia de singularidade.

De acordo com dados apresentados pela *StatCounter*⁶ em outubro de 2019, o Sistema Operacional da *Google* lidera o *ranking* dos mais usados no Brasil com 87,86%, ficando muito à frente do seu principal concorrente, o *iOS*⁷ da *Apple*⁸ que domina apenas 11,36%, além dos demais, como o *Windows*⁹ e o sistema da *Samsung*¹⁰. Na Figura 1 vê-se o gráfico detalhado desta disputa entre os principais sistemas operacionais dos dispositivos móveis mais usados no Brasil.

¹ Um conjunto de sistemas executados em computadores diferentes e independentes entre si e que passam a ideia de um único sistema.

² <https://about.google/intl/pt-BR/>

³ <https://www.openhandsetalliance.com>

⁴ Termo dado a programas de computador que são disponibilizados publicamente o código, seja para estudo, modificação e distribuição de forma gratuita

⁵ <https://www.linuxfoundation.org>

⁶ <https://gs.statcounter.com>

⁷ Sistema Operacional desenvolvido pela *Apple* para seus dispositivos móveis.

⁸ <https://www.apple.com>

⁹ <https://www.microsoft.com/pt-br/windows/phones>

¹⁰ <https://www.samsung.com/br>

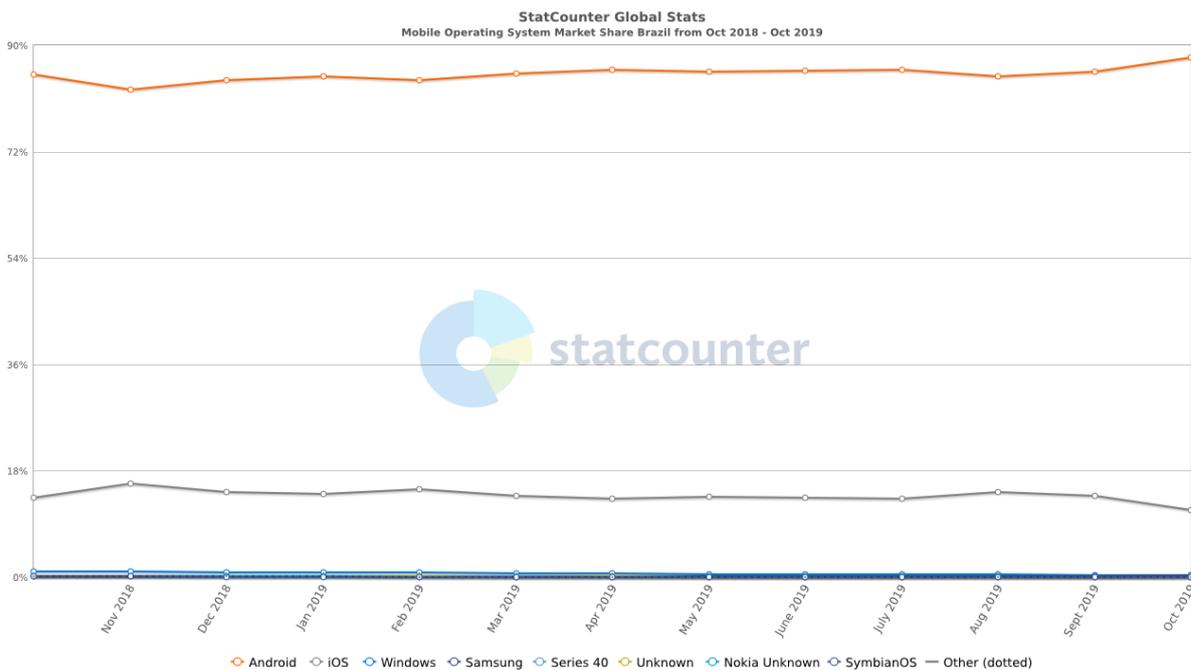


Figura 1: Ranking dos Sistemas Operacionais mobile no Brasil

Fonte: <https://gs.statcounter.com/os-market-share/mobile/brazil>

2.1.1 Arquitetura da plataforma Android

A plataforma *Android* é projetada para rodar em diversos aparelhos de configurações e tamanhos diferentes. Ela se divide em diversas camadas, como é visto na Figura 2. Nos subtópicos seguintes é apresentada uma explicação sobre cada uma dessas camadas.

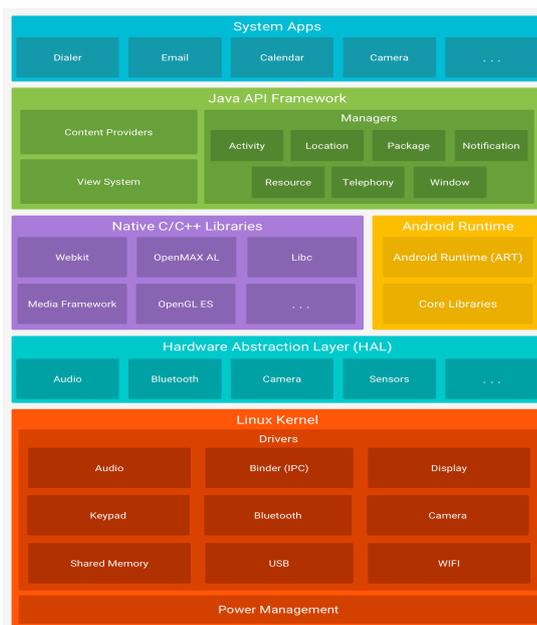


Figura 2: Arquitetura da plataforma Android

Fonte: https://developer.android.com/guide/platform/images/android-stack_2x.png

2.1.1.1 Kernel do Linux

O *Kernel*, ou seja, o núcleo, é como o “cérebro” do sistema e é essencial para o funcionamento do *Android*. É uma camada fundamental para o perfeito funcionamento e comunicação entre *hardware*¹¹ e *software*¹². Sua base sendo *Linux*, o *Android* herda seus famosos recursos de segurança e, por ser um sistema conhecido, facilita para os fabricantes de aparelhos a desenvolver os *drivers*¹³ essenciais para o funcionamento de seus dispositivos internos.

É no *Kernel* onde acontece o gerenciamento dos recursos de hardware que são usados pelo Sistema Operacional.

2.1.1.2 Camada de Abstração de Hardware

A camada de abstração de *hardware* é responsável pela comunicação do sistema com os dispositivos internos, sensores e periféricos¹⁴ que podem ser conectados ao aparelho. É nela onde acontecem as chamadas de componentes de *hardwares* através de *interfaces*. Essas *interfaces* são como programas que utilizam bibliotecas desenvolvidas para esta finalidade, auxiliando a comunicação entre um componente lógico, ou seja, algum programa que necessita de um *hardware* para seu funcionamento, e um componente físico, ou seja, o *hardware* que o programa precisa para seu funcionamento ou vice-versa, que normalmente não podem se comunicar de forma direta, tornando assim, cada *interface* sendo específica para cada tipo de *hardware*.

2.1.1.3 Android Runtime

Os aplicativos para *Android* em sua maioria são escritos na linguagem Java e, dessa forma, para serem executados eles necessitam ser compilados em uma Máquina Virtual Java¹⁵. Porém, no *Android*, é necessário recompilar o código para um *bytecode*¹⁶ específico para a plataforma *Android*.

Nos sistemas *Android*, antes da versão 5.0 era utilizado o *Dalvik*, uma Máquina Virtual desenvolvida para executar o *bytecode* gerado a partir do *bytecode* do Java. A partir da versão 5.0 houve melhorias de desempenho e recursos no *Runtime*, cada aplicativo executa o próprio processo e são criadas várias Máquinas Virtuais. Para que isto seja possível,

¹¹ Termo dado às partes físicas de um computador, dispositivo móvel e outros aparelhos que utilizam um sistema computacional.

¹² Termo dado à parte lógica de computadores, dispositivos móveis e outros aparelhos que utilizam um sistema computacional, estes são mais conhecidos também como programas de computador.

¹³ Termo dado a programas específicos responsáveis pela comunicação entre o Sistema Operacional e o *hardware*, algum dispositivo interno necessário para uma tarefa.

¹⁴ São dispositivos que quando conectados a um aparelho com recursos computacionais, são usados para enviar informações para este aparelho ou receber informações dele.

¹⁵ Um programa encarregado de converter os *bytecodes* do aplicativo Java em códigos para serem executados na máquina.

¹⁶ Um código gerado da linguagem Java para ser executado em uma Máquina Virtual Java, tornando possível ser executado em diversas plataformas diferentes.

estas Máquinas Virtuais são responsáveis por executar o *bytecode DEX*, desenvolvido especificamente para a própria plataforma *Android* ([ANDROID](#)) .

2.1.1.4 Bibliotecas C/C++ Nativas

Os recursos e serviços do sistema *Android* citados nas camadas anteriores também são implementações e estas são implementadas em código nativo que necessitam de bibliotecas nativas. Essas bibliotecas são construídas na linguagem C e C++ e também estão disponíveis para serem usadas por aplicativos através de uma *Framework*¹⁷ em Java, onde os desenvolvedores podem acessá-las por ela ([ANDROID](#)) .

2.1.1.5 Java API Framework

Assim como para o sistema se comunicar com um componente de *hardware* é necessário uma *interface* que faça essa ligação, os desenvolvedores de aplicativos também dispõem de *interfaces* para auxiliar no uso de todos os recursos e serviços que o sistema *Android* possui em seus projetos.

Através desta *Framework* de *APIs*¹⁸, os desenvolvedores podem utilizar os mesmos componentes de sistema que os aplicativos nativos do *Android* utilizam, como componentes de IU¹⁹, componentes de *layout*²⁰, gerenciador de notificações, gerenciador de tarefas e provedores de conteúdo.

2.1.1.6 Aplicativos do Sistema

O *Android* possui aplicativos nativos, aplicativos estes que exercem funções básicas do sistema como efetuar ligações, trocar mensagens *SMS*, gerenciar *e-mails*, navegar na *web*, gerenciar agenda telefônica e calendário, configurações de sistema, entre outros. Estes aplicativos vem definidos como aplicativos padrões de uso, mas nada impede que desenvolvedores possam criar aplicativos para exercerem as mesmas funções e ainda torná-los como padrões de uso. É o que muitas empresas fabricantes de dispositivos móveis que usam o sistema *Android* fazem. Elas desenvolvem aplicativos próprios para algumas dessas tarefas e lançam seus produtos com seus aplicativos como padrões.

Desenvolvedores independentes também podem criar aplicativos para substituírem os nativos do *Android* ou que os usem através de chamadas pelo sistema. Se ele quiser que seu *app* (aplicativo) em determinada situação de uso envie um *e-mail*, por exemplo, não é necessário programá-lo para isso, se este não for o foco ou não for tão necessária esta

¹⁷ Um conjunto de ferramentas desenvolvidas com a finalidade de auxiliar na resolução de problemas específicos utilizando técnicas pré-definidas.

¹⁸ Padrões criados, implementados e disponibilizados para auxiliar terceiros no uso de recursos e serviços específicos.

¹⁹ Interface gráfica pela qual o usuário pode interagir com o sistema.

²⁰ Esboço da organização dos elementos presentes em um ambiente.

implementação. Ele pode chamar um *app* gerenciador de *e-mail* já instalado no sistema para executar esta tarefa.

2.1.2 Segurança do Sistema *Android*

Por ser o SO (Sistema Operacional) mais utilizado nos dispositivos móveis, o *Android* acaba se tornando um potencial alvo para os mal intencionados que desejam se aproveitar da grande quantidade de usuários para conseguir informações pessoais e sigilosas através de técnicas de invasão e arquivos maliciosos, principalmente pelo uso de *malwares*²¹.

Como vimos anteriormente, o *Android* é um sistema de código aberto, tornando possível que usuários possam fazer inúmeras modificações nele. Essa flexibilidade prejudica a segurança do sistema da *Google*, por não ter tanta restrição quanto ao uso de *softwares* como o sistema da *Apple*, por exemplo, tornando-se mais fácil a instalação de aplicativos indesejados, e assim, facilitando também a instalação de aplicativos maliciosos que podem comprometer o uso dos aparelhos e a segurança de informações pessoais dos usuários (LEITE; MACEDO, 2017).

Para combater estes perigos, a *Google* tem trabalhado e investido nesta parte relacionada às permissões de publicações de aplicativos em sua loja oficial de *apps*. Antes era possível publicar aplicativos com mais facilidade. Agora, os novos aplicativos passam por algumas avaliações, diminuindo assim os riscos da distribuição de aplicativos maliciosos. Também é feita uma avaliação de restrição de faixa etária, a fim de que conteúdos impróprios para certas idades não alcancem usuários desqualificados para o acesso.

Antes de qualquer aplicativo ser aceito no *Google Play*²², o analisamos em busca de segurança, incluindo possíveis problemas de segurança. Também analisamos continuamente os mais de um milhão de aplicativos no *Google Play* em busca de ameaças adicionais (ANDROID) .

2.2 *Android Studio*

O *Android Studio* é o ambiente de desenvolvimento integrado oficial do *Android*. Criado especificamente para o *Android*, ele acelera o desenvolvimento e ajuda a criar aplicativos da mais alta qualidade para todos os dispositivos *Android* (ANDROID) .

O *Android Studio* já vem com todas as ferramentas *Android SDK* para desenhar, configurar e testar o projeto, além do arquivo de imagem mais recente para emular o *Android*. Possui um *layout* simples, porém intuitivo e facilita bastante a vida do desenvolvedor (BARROS, 2013).

²¹ Termo dado a programas maliciosos capazes de roubar informações e danificar aparelhos, entre outros males no meio computacional

²² Loja de aplicativos oficial da *Google*

2.2.1 Android SDK

O *Android SDK* é o *software* utilizado para desenvolver aplicações para o sistema *Android*. Ele possui um emulador similar aos celulares reais com diversos modelos e podendo instalar as diversas versões do *Android*. Possui ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias para desenvolver aplicações (LECHETA, 2015).

Desta forma, o *Android Studio* juntamente com o *Android SDK* se tornam uma ferramenta completa e poderosa disponível gratuitamente no site *Android Developers*²³, tornando o trabalho de desenvolver um aplicativo mais prático e ágil, levando em conta a quantidade de recursos e auxílios concentrados em um só ambiente de desenvolvimento.

2.3 Firebase

O *Firebase* é um *BaaS*²⁴ (*Backend as a Service*) para aplicações *web* e *mobile* do *Google*. Foi lançado em 2004 e tem crescido muito nos últimos anos, se tornando uma das mais escolhidas para alguns tipos de projetos por oferecer uma grande quantidade de serviços e ser de fácil implementação (ORLANDI, 2018).

Na Figura 3, podem ser observados os serviços que o *Firebase* oferece. Na subseção seguinte, será descrito um pouco sobre o *Firebase Storage*, que é o serviço utilizado neste trabalho.

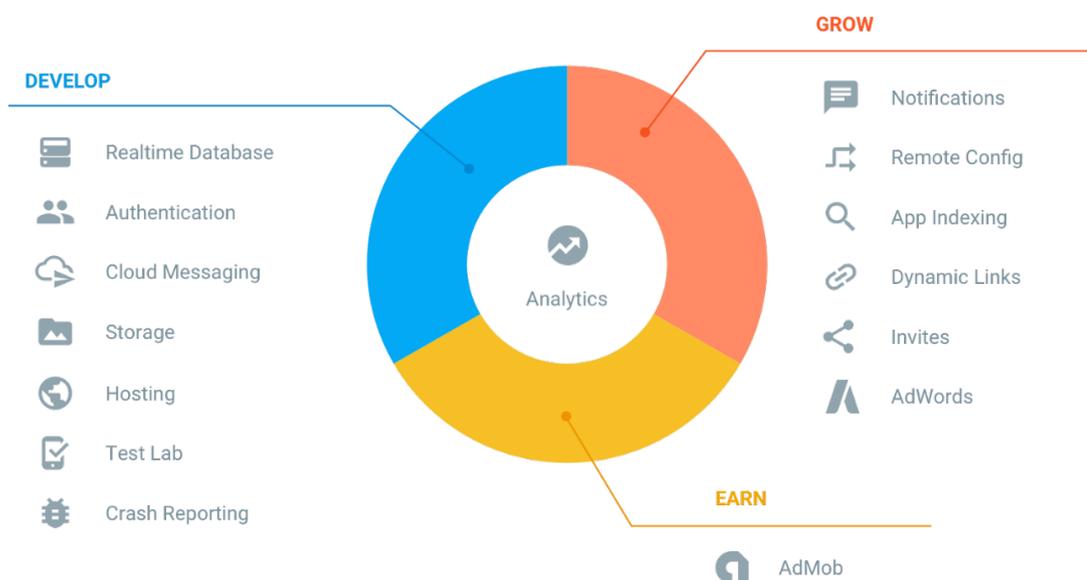


Figura 3: Serviços oferecidos pelo *Firebase*

Fonte: <https://blog.rocketseat.com.br/content/images/2018/12/firebase-services.png>

²³ <https://developer.android.com/studio/?hl=pt-br>

²⁴ Um conjunto de serviços e ferramentas disponibilizados em nuvem para serem utilizadas por outras aplicações.

2.3.1 *Firestore Storage*

O *Firestore Storage* ou *Firestore Cloud Storage* é uma ótima opção para aplicações que necessitam armazenar arquivos de mídia²⁵ na nuvem com praticidade e segurança. Pela facilidade de implementação, tem sido bastante utilizado nos últimos anos.

O *Cloud Storage* para *Firestore* é um serviço de armazenamento de objetos poderoso, simples e econômico criado para a escala do *Google*. Com os SDKs do *Firestore* para *Cloud Storage*, você usa a segurança do *Google* para fazer o *upload* e o *download* de arquivos nos aplicativos *Firestore*, independentemente da qualidade da rede ([FIREBASE](#)) .

Seu gerenciamento é bem fácil e organizado, tornando simples para os desenvolvedores o seu uso nas suas aplicações. Os arquivos são organizados em uma estrutura hierárquica dentro do *Google Cloud Storage*²⁶. O aplicativo recebe uma referência do arquivo, tornando possível o acesso diretamente a ele ou a um nó correspondente a níveis superiores na hierarquia ao qual o arquivo pertence, permitindo *upload* e *download* de dados, assim como também excluí-los ([FIREBASE](#)) .

2.4 Java

Java é uma linguagem de programação²⁷ orientada a objetos²⁸, seus números ultrapassam os 9 (nove) milhões de desenvolvedores e é usada em mais de 3 (três) bilhões de dispositivos, tornando-se uma das mais usadas em todo o mundo²⁹ ([JAVA](#)) .

Foi desenvolvida pela *Sun Microsystems* na década de 90 e, atualmente, tanto a empresa *Sun* como a linguagem Java pertencem à *Oracle*³⁰. Os programas em Java são compilados para um *bytecode* e executados por uma máquina virtual, desta forma seus programas podem ser executados em qualquer plataforma que tenha o interpretador instalado, assim, os desenvolvedores precisam escrever apenas uma versão de código ([BRITO, 2014](#)).

O Java, por muitos anos, se manteve no topo das linguagens de programação preferidas dos desenvolvedores e ainda é uma das mais populares, ficando atrás apenas da linguagem *Python* que recentemente cresceu fortemente no mercado. Na Figura 4 é apresentado um *ranking* das linguagens de programação mais populares no mundo em 2019, publicado

²⁵ São arquivos do tipo de imagem, áudio e vídeo, por exemplo.

²⁶ <https://cloud.google.com/storage/?hl=pt-BR>

²⁷ Conjunto de comandos organizados semanticamente para serem convertidos em instruções para o computador, formando programas.

²⁸ Um paradigma de programação que busca trazer elementos do mundo real como objetos para dentro do código, definindo características para estes objetos e fazendo com que interajam entre si.

²⁹ <https://www.oracle.com/br/java/>

³⁰ <https://www.oracle.com/index.html>

pela *IEEE Spectrum*³¹ que se baseia em 11 métricas e 8 fontes para chegar ao resultado apresentado³².



Rank	Language	Type	Score
1	Python	🌐 🖥️ ⚙️	100.0
2	Java	🌐 📱 🖥️	96.3
3	C	📱 🖥️ ⚙️	94.4
4	C++	📱 🖥️ ⚙️	87.5
5	R	🖥️	81.5
6	JavaScript	🌐	79.4
7	C#	🌐 📱 🖥️ ⚙️	74.5
8	Matlab	🖥️	70.6
9	Swift	📱 🖥️	69.1
10	Go	🌐 🖥️	68.0

Figura 4: Ranking das linguagens de programação mais populares no mundo em 2019.

Fonte: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2019>

Para desenvolver aplicações em Java é necessário que se tenha instalado na máquina o JDK (*Java Development Kit*). O JDK inclui o JRE (*Java Runtime Environment*)³³, um conjunto de classes de API, o compilador³⁴ JAVA, *Webstart* e arquivos adicionais necessários para criar *applets*³⁵ e aplicativos Java (**JAVA**) .

³¹ Principal revista e site da *IEEE (Institute of Electrical and Electronics Engineers)*.

³² <https://spectrum.ieee.org/static/ieee-top-programming-languages-2019-methods>

³³ Pacote de recursos necessários para executar aplicações em Java

³⁴ Um programa de computador que traduz o código escrito em uma linguagem de programação para linguagem de máquina.

³⁵ Um pequeno *software* que executa uma tarefa específica dentro de outro *software*

2.5 Web Service

Web Services nada mais são que serviços tradicionais oferecidos pela *internet*. O que torna um *Web Service* especial é que ele obedece padrões que podem ser acessados por outras aplicações que utilizem os mesmos padrões (TANENBAUM; STEEN, 2007). Em outras palavras, são serviços hospedados em um servidor que podem ser acessados por clientes através da *internet*. Esses acessos são feitos utilizando protocolos como HTTP, XML, SOAP, entre outros.

Usuários de *Web Services* não precisam conhecer sua estrutura ou sua linguagem de programação para poder fazer uso dos serviços oferecidos. Estes usuários somente precisam saber do que necessita, para realizar a solicitação do serviço desejado e obter a resposta sobre o serviço solicitado (CORREIA, 2005).

2.5.1 Web Service RESTful

Web Services RESTful são serviços que possuem endereços na *web* para acessar e transferir informações e conteúdos através do protocolo HTTP e seus métodos *GET*, *POST*, *UPDATE* e *DELETE*. Através desses métodos, o cliente pode solicitar um recurso do *web service*, se o pedido for bem sucedido, o servidor que hospeda o serviço retorna uma representação do recurso para o cliente (ALVES et al., 2019). Na Figura 5, é possível ver como funciona este ciclo de requisição entre cliente e servidor através de um *Web Service RESTful*.

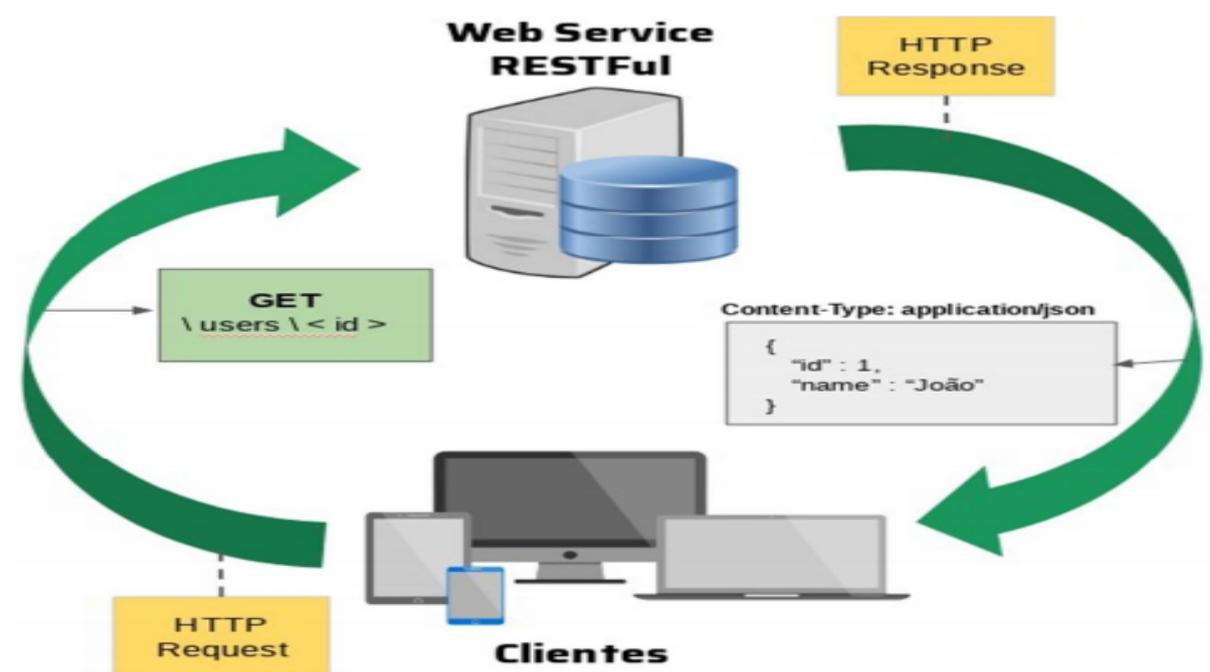


Figura 5: Ciclo de requisição em um Web Service RESTful.

2.6 NetBeans

NetBeans é um *software* ambiente de desenvolvimento utilizado principalmente para desenvolver aplicações em Java, mas pode ser usado para desenvolver programas em diversas outras linguagens de programação, seja para plataformas *desktop* ou *web*.

O *NetBeans* IDE é um ambiente de desenvolvimento - uma ferramenta para programadores, que permite escrever, compilar, depurar e instalar programas. O IDE é completamente escrito em Java, mas pode suportar qualquer linguagem de programação. Existe também um grande número de módulos para estender as funcionalidades do IDE *NetBeans*. O *NetBeans* IDE é um produto livre, sem restrições à sua forma de utilização ([NETBEANS](#)).

O seu projeto de criação começou em 1996 em uma universidade em Praga, na República Checa, por dois estudantes tchecos, quando a linguagem Java não era tão popular como hoje. Seu primeiro nome foi “*xelfi*”, em alusão ao *delphi*³⁶, pois a ideia era fazer com que o *NetBeans* possuísse funções semelhantes ao *delphi*, mas usando a linguagem Java. Em 1999, o projeto evoluiu para um IDE proprietário, com o nome de *NetBeans DeveloperX2*, nome que veio da reutilização de componentes que eram a base do Java. A *Sun* havia desistido de sua IDE Java *Workshop* e adquiriu o *NetBeans DeveloperX2* incorporando à sua linha de *softwares*. O *NetBeans* é uma das mais poderosas ferramentas para o desenvolvimento de aplicações robustas e profissionais, pela facilidade de usar e seu ambiente visual que facilita a interação com o desenvolvedor, auxiliando na criação do código ([CARDOSO, 2013](#)).

2.7 Banco de Dados

Muitas aplicações necessitam carregar bastante informações para funcionarem parcialmente ou até integralmente. Essas informações nem sempre ficam armazenadas dentro da aplicação, pois geralmente quando uma aplicação é encerrada, todos os processos relacionados a essa aplicação são encerrados e, assim, as informações acabam se perdendo. Então, para solucionar este problema, é utilizado um sistema de banco de dados.

Banco de dados é um conjunto de informações ou dados armazenados de forma estruturada em um sistema de computador. Geralmente são administrados por um Sistema Gerenciador de Banco de Dados (SGBD). Esse controle dos dados por um SGBD e outras aplicações é chamado de sistema de banco de dados ([ORACLE](#)). Quando uma aplicação utiliza um banco de dados para armazenar dados e informações, mesmo que a aplicação seja encerrada, as informações não serão perdidas e poderão ser recuperadas futuramente quando a aplicação for iniciada novamente.

³⁶ Uma IDE, compilador e também linguagem de programação.

Um modelo de banco de dados bastante usado é o modelo relacional. Neste modelo o banco de dados é organizado por tabelas e relações, onde cada tabela possui registros e atributos e cada um desses registros possui um identificador único, chamado de chave. As relações são feitas entre as tabelas através dessas chaves e são como abstrações de como a representação dessas tabelas podem se relacionar no mundo real.

2.8 SQL

SQL é uma das linguagens de programação mais utilizadas para desenvolver bancos de dados. É uma linguagem cheia de recursos e é a base para se trabalhar com banco de dados relacionais.

SQL é uma linguagem de programação usada por quase todos os bancos de dados relacionais para consultar, manipular e definir dados e fornecer controle de acesso. O SQL foi desenvolvido pela primeira vez na IBM³⁷ nos anos 1970, com a Oracle como principal contribuinte, o que levou à implementação do padrão SQL ANSI³⁸; o SQL estimulou muitas extensões de empresas como IBM, Oracle e Microsoft. Embora o SQL ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer (ORACLE).

A linguagem SQL é dividida em 4 (quatro) agrupamentos que são eles: linguagem de definição de dados, linguagem de manipulação de dados, linguagem de controle de dados e linguagem de controle de transações (OLIVEIRA et al., 2018). Na Tabela 1 são apresentados os comandos e as descrições de cada comando referente a cada agrupamento.

Tabela 1: *Linguagem SQL: Agrupamentos e seus comandos*

Linguagem SQL: Agrupamentos e seus comandos			
Agrupamento	Definição	Comandos	Descrição dos comandos
Linguagem de definição de dados (DDL)	Comandos usados para definir as estruturas dos dados	CREATE	Criar uma tabela ou outro objeto
		ALTER	Alterar qualquer objeto criado
		DROP	Apagar qualquer objeto criado
Linguagem de manipulação de dados (DML)	Comandos usados para inserir, selecionar, deletar e atualizar dados	INSERT	Adicionar dados em uma tabela
		SELECT	Realizar consultas no banco
		DELETE	Deletar dados do banco
		UPDATE	Atualizar dados no banco
Linguagem de controle de dados (DCL)	Comandos usados para definir o controle de gerenciamento do banco pelos usuários	GRANT	Conceder permissões definir regras a usuários
		REVOKE	Revogar permissões e definições de regras a usuários
Linguagem de controle de transações (DTL)	Comandos para o controle de transações do banco	BEGIN TRANSACTION	Iniciar uma transação
		COMMIT	Confirmar alterações
		ROLLBACK	Cancelar

³⁷ <https://www.ibm.com/br-pt>

³⁸ Versão padronizada do SQL estabelecida pelo Instituto Americano de Padronização ou *American National Standards Institute* (ANSI)

2.9 MySQL

O *MySQL* é um Sistema Gerenciador de Banco de Dados de código aberto que trabalha com a linguagem SQL, pode ser usado em diversos Sistemas Operacionais e é bastante usado para trabalhar com aplicações *web* e *sites*.

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto baseado em SQL. Ele foi projetado e otimizado para aplicativos da *web* e pode ser executado em qualquer plataforma. Como surgiram requisitos novos e diferentes com a *internet*, o MySQL tornou-se a plataforma preferida para desenvolvedores da *web* e aplicativos baseados na *web*. Por ter sido projetado para processar milhões de consultas e milhares de transações, o MySQL é uma escolha popular para empresas de comércio eletrônico que precisam gerenciar várias transferências de dinheiro. A flexibilidade sob demanda é o principal recurso do MySQL.

O MySQL é o DBMS (*Database Management System*) por trás de alguns dos principais sites e aplicativos baseados na *web* do mundo, incluindo Airbnb³⁹, Uber⁴⁰, LinkedIn⁴¹, Facebook⁴², Twitter⁴³ e YouTube⁴⁴ (ORACLE).

Como um banco de dados relacional, ele armazena os dados de forma organizada em tabelas separadas em vez de num mesmo local, proporcionando velocidade e flexibilidade. Projetado para trabalhar com bancos de dados extensos com mais rapidez, é usado em ambientes de produção em alta escala mesmo estando em constante desenvolvimento. Possui um grande leque de funcionalidades, sua conectividade, velocidade e segurança o tornam uma ótima opção para trabalhar via *internet* (JUNIOR, 2018).

2.10 Google Maps API

A API do *Google Maps* é um serviço disponibilizado pela *Google* tanto de forma gratuita como paga para aplicações que desejam cobrar algo de seus usuários. Através desta API é possível implementar várias funcionalidades do *Google Maps* em suas aplicações, como marcar pontos no mapa, traçar rotas, buscar endereços e muitas outras. O *Google Maps* é uma ferramenta poderosa de mapas, contendo informações geográficas, climáticas e visuais de quase todos os lugares do planeta.

2.10.1 SDK do *Google Maps* para *Android*

O SDK do *Google Maps* para *Android* permite que desenvolvedores implementem em suas aplicações um mapa personalizado a partir da ferramenta *Google Maps*.

³⁹ <https://www.airbnb.com.br>

⁴⁰ <https://www.uber.com/br/pt-br/>

⁴¹ <https://www.linkedin.com>

⁴² <https://www.facebook.com>

⁴³ <https://twitter.com>

⁴⁴ <https://www.youtube.com>

Com o SDK do *Google Maps* para *Android*, você pode adicionar mapas com base nos dados do *Google Maps* ao seu aplicativo. A API gerencia automaticamente o acesso aos servidores do *Google Maps*, o download de dados, a exibição do mapa e a resposta aos gestos do mapa. Você também pode usar chamadas de API para adicionar marcadores, polígonos e superposições a um mapa básico e alterar a exibição do usuário de uma área específica do mapa. Esses objetos fornecem informações adicionais para os locais do mapa e permitem a interação do usuário com o mapa (GOOGLE) .

Ideal para projetos que buscam oferecer melhor visibilidade de cidades, interações com pontos no mapa, busca de endereços, uso de localização do usuário em tempo real, etc.

3 Trabalhos Relacionados

Este capítulo apresenta alguns trabalhos correlatados que tem por objetivo auxiliar as pessoas na busca por imóveis para comprar ou alugar, desenvolvidos para plataforma *web* ou *mobile*, e apresenta também uma análise comparativa entre estes trabalhos e o trabalho aqui proposto.

3.1 Descrição dos Trabalhos

3.1.1 Imóveis Estudantes - Aplicativo Facebook

Desenvolvido por TADDEI, é um sistema criado para plataforma *web*, uma ferramenta integrada à rede social Facebook através de sua API. Tem por objetivo auxiliar estudantes a encontrar um novo lugar para morar na cidade de Pelotas-RS, podendo ser um pensão ou quarto em apartamento alugado por alguém disposto a dividir. Foi desenvolvido em PHP¹ e seu banco de dados com MySQL, além de usar a API do Facebook que facilita na divulgação do sistema e visualização dos perfis dos usuários, também utiliza a API do *Google Maps* para facilitar a visualização da localização dos imóveis divulgados no sistema (TADDEI, 2012).

3.1.2 Viva Real

Viva Real² é um portal Imobiliário iniciado como uma *startup*³ em 2007, nos Estados Unidos, e que veio para o Brasil no ano de 2009. Espalhando-se pelo país, fazendo parcerias com imobiliárias e corretoras, a empresa cresceu bastante consolidando-se no mercado nacional. Desenvolvido tanto para plataforma *web* como *mobile*, possui 3 (três) milhões de anúncios de mais de 17 (dezesete) mil corretores e imobiliárias e 13 milhões de visitantes mensais. Oferece imóveis para comprar e alugar (MIRANDA, 2015).

3.1.3 ZAP Imóveis

ZAP Imóveis⁴ é um portal brasileiro desenvolvido para plataforma *web* e *mobile*, pertence ao Grupo Globo⁵ e tem como objetivo a oferta de imóveis para venda e aluguel. Inicialmente, apenas como plataforma *web* nos anos 2000, cujo nome era Planeta Imóvel e, em 2005, era líder no setor. Em 2007 recebeu o nome de Zap Imóveis. Com fortes

¹ Uma linguagem interpretativa livre usada principalmente no desenvolvimento de sistemas *web*

² <https://www.vivareal.com.br>

³ Uma empresa recém-criada que adota conceitos inovadores para desenvolver ideias.

⁴ <https://www.zapimoveis.com.br>

⁵ <https://grupoglobo.globo.com>

investimentos em *marketing*⁶ em todos os tipos de mídias, o portal possui mais de 11 (onze) milhões de acessos mensais e mais de 700 (setecentos) mil requisições enviadas aos anunciantes pelos usuários interessados em seus imóveis (MIRANDA, 2015).

3.1.4 Incasa Picos

Desenvolvido por NETO, é um portal *web* com objetivo de auxiliar seus usuários a encontrar e anunciar imóveis na região da cidade de Picos. Definido como um *e-commerce* C2C⁷, propõe a interação direta entre os usuários, que podem anunciar de forma prática e supervisionada pelo sistema e pelo administrador do sistema, a fim de evitar a omissão de informações relevantes para a compreensão do anúncio, e também visualizar outros anúncios com informações adequadas de localização e moradia. Desenvolvido em PHP, utiliza um banco de dados MySQL (NETO, 2016).

3.2 Análise Comparativa

É notável a semelhança entre os trabalhos apresentados na sessão anterior, mas é possível notar também algumas particularidades em alguns deles. Na Tabela 2 são expostas algumas comparações entre estes trabalhos e o trabalho proposto.

Tabela 2: Análise comparativa entre os trabalhos relacionados e o trabalho proposto

Análise comparativa entre os trabalhos relacionados e o trabalho proposto							
Aplicação	Plataforma WEB	Plataforma Mobile	Anúncios de Venda	Anúncios de Aluguel	Anúncios de Dividir Aluguel	Intermédio de comunicação entre usuários	Informações para análise rápida de perfis de usuários dentro da aplicação
Trabalho Proposto	não	sim	não	sim	sim	sim	sim
Imóveis Estudantes	sim	não	não	sim	sim	sim	não
Viva Real	sim	sim	sim	sim	não	sim	não
ZAP Imóveis	sim	sim	sim	sim	não	sim	não
Incasa Picos	sim	não	sim	sim	não	sim	não

Na Tabela 2 pode-se observar comparações entre as aplicações com características que são relevantes para o trabalho proposto. A seguir será detalhado um pouco sobre essas características.

1. **Plataforma WEB** - Necessita de um navegador *web* para acessar o sistema, descartando a necessidade de instalação de um *software* específico para o uso. Todas as aplicações possuem plataforma *web*, exceto o trabalho proposto que foca em uma aplicação *mobile*.
2. **Plataforma Mobile** - Necessita de um dispositivo móvel para acessar o sistema. Algumas aplicações não possuem plataforma *mobile*. O trabalho proposto foca nesta

⁶ Técnica utilizada para agregar valor a um produto ou serviço e promover sua divulgação a fim de atrair mais público

⁷ Comércio eletrônico que realiza as transações de consumidor para consumidor

plataforma pela praticidade de acesso de qualquer lugar com cobertura de acesso à *internet*, descartando a necessidade de um *software* terceiro para acessar o sistema, como um navegador, por exemplo.

3. **Anúncios de Venda** - Anúncios para usuários que desejam comprar um imóvel. Mais voltado para aqueles que desejam morar definitivamente no local ou fazer algum tipo de investimento com ele. Nem todas as aplicações fazem este tipo de anúncio.
4. **Anúncios de Aluguel** - Anúncios para usuários interessados em alugar um imóvel. Uma alternativa para aqueles que não podem ou preferem não comprar um imóvel. Boa parte da população mora em imóvel alugado e, ocasionalmente, procuram um novo imóvel para alugar com custos e condições mais viáveis. É opção também para aqueles que se mudam de forma passageira para uma cidade, como é o caso de estudantes, por exemplo. Todas as aplicações possuem anúncios de aluguel e o trabalho proposto foca nesta categoria justamente por ser uma procura mais constante e para auxiliar estes estudantes que migram de cidade por um período passageiro.
5. **Anúncios de Dividir Alguel** - Anúncios para usuários interessados em dividir as despesas de aluguel com outros inquilinos, pensando principalmente nos estudantes, já que adotam bastante esta prática, mas que não é exclusividade deles. Boa parte das pessoas que migram para outra cidade buscam esta alternativa para diminuir as despesas na nova moradia. Apenas duas aplicações possuem este tipo de anúncio e é um dos principais focos do trabalho proposto.
6. **Intermédio de comunicação entre usuários** - Possui formas de contato entre os usuários para que possam combinar visitas aos imóveis, combinar detalhes relativos a contrato e, no exemplo do trabalho proposto, para que possam combinar detalhes sobre como dividir o aluguel de um imóvel. Todas as aplicações possuem intermédio de comunicação entre os usuários.
7. **Informações para análise rápida de perfis de usuários dentro da aplicação** - Para usuários interessados em dividir alguel com outros inquilinos, algumas informações de perfil são interessantes para estes usuários, como o sexo, principalmente. Ter estas informações já disponíveis na aplicação evita o trabalho de pesquisar sobre a pessoa interessada em outras ferramentas e aplicações. O trabalho proposto possui essa característica, pensado na questão de que é visto neste tipo de anúncio em outros meios de divulgação algumas exigências para ser aceito para dividir a moradia. Afinal, nem todas as pessoas e personalidades diversas conseguem conviver juntas.

4 Modelagem e Desenvolvimento do Sistema

A proposta principal do trabalho é o desenvolvimento de um aplicativo móvel, porém, neste trabalho o aplicativo é parte de um sistema distribuído que funciona no modelo cliente-servidor¹. Desta forma, o trabalho apresentará o desenvolvimento do aplicativo como o cliente do sistema e também apresentará o lado do servidor, composto por um *web service* e o banco de dados.

Neste capítulo é apresentada a modelagem e o desenvolvimento do sistema em sua totalidade, composto pelo aplicativo, o banco de dados e o *web service*.

4.1 Modelagem do Sistema

4.1.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso é uma representação gráfica das funcionalidades do sistema através de atores² e casos de uso³, com suas relações⁴ específicas. Na Figura 6, é possível observar o Diagrama de Casos de Uso do sistema apresentado neste trabalho criado através da ferramenta Astah⁵.

¹ Arquitetura que distribui a carga de processamento de um sistema entre cliente e servidor que se comunicam através de uma rede.

² Uma representação de entidade que participa do sistema, podendo ser um tipo de usuário ou outro sistema

³ Representa uma funcionalidade ou ação observável dentro do sistema

⁴ Representa as interações entre atores e casos de uso ou de casos de uso entre si

⁵ <http://astah.net>

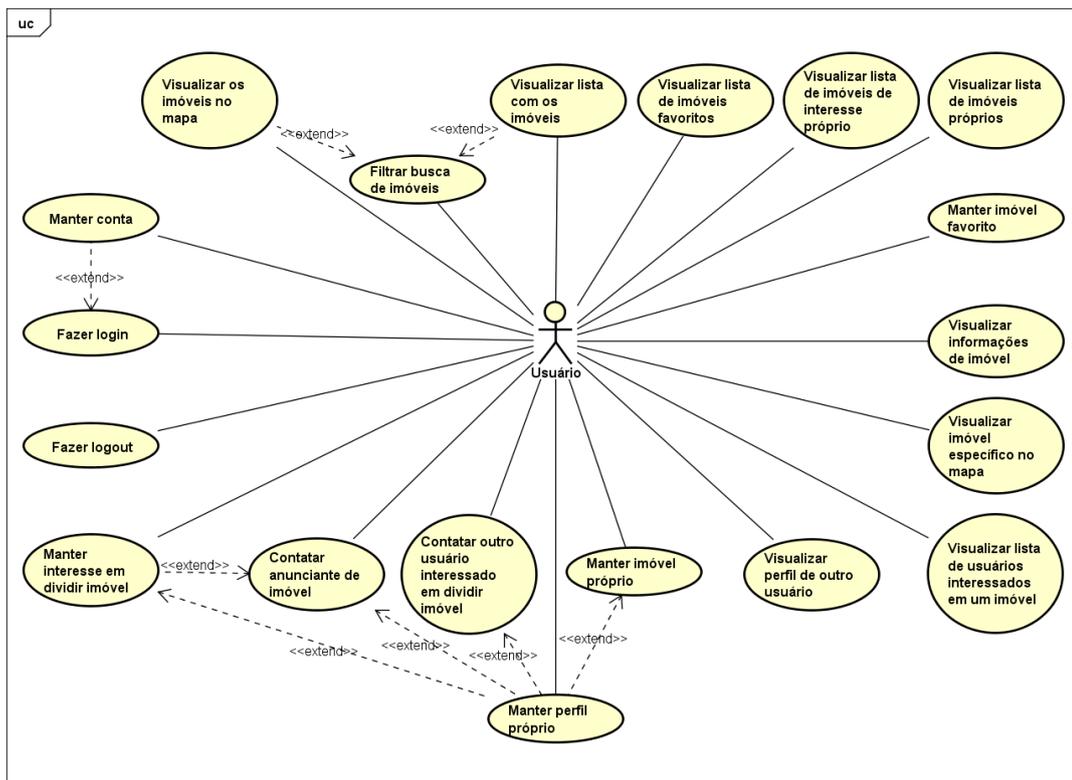


Figura 6: Diagrama de Casos de Uso.

Fonte: Autor

Através do Diagrama de Casos de Uso é possível observar como o sistema se comporta e, para ajudar a compreender, segue a descrição de cada caso de uso:

- **Fazer *login***

Ator: Usuário

Descrição: O usuário deve fazer autenticação no sistema com seu *e-mail* e senha criada no ato do cadastro da conta no sistema. Caso o usuário não possua uma conta, ele deve cadastrar uma antes de fazer *login*.

- **Manter conta**

Ator: Usuário

Descrição: O usuário realiza o cadastro da conta informando seu *e-mail*, criando uma senha e confirmando a senha. Ele pode também redefinir a senha posteriormente.

- **Fazer *logout***

Ator: Usuário

Descrição: O usuário pode fazer *logout* e sair do sistema.

- **Visualizar os imóveis no mapa**

Ator: Usuário

Descrição: O usuário pode visualizar os imóveis no mapa.

- **Visualizar lista com os imóveis**
Ator: Usuário
Descrição: O usuário pode visualizar os imóveis numa lista.
- **Filtrar busca de imóveis**
Ator: Usuário
Descrição: O usuário pode realizar uma busca de imóveis filtrada com suas preferências e o resultado desta busca pode ser exibido ou no mapa ou numa lista.
- **Visualizar lista de imóveis favoritos**
Ator: Usuário
Descrição: O usuário pode visualizar a lista dos imóveis que ele marcou como favorito.
- **Visualizar lista de imóveis de interesse próprio**
Ator: Usuário
Descrição: O usuário pode visualizar a lista dos imóveis que ele demonstrou interesse, tanto para alugar como para dividir aluguel com outras pessoas.
- **Visualizar lista de imóveis próprios**
Ator: Usuário
Descrição: O usuário pode visualizar em uma lista os imóveis que ele cadastrou no sistema.
- **Manter imóvel favorito**
Ator: Usuário
Descrição: O usuário pode marcar imóveis como favorito para encontrar tais imóveis com mais facilidade futuramente. Pode também desmarcar estes imóveis de favorito.
- **Visualizar informações de imóvel**
Ator: Usuário
Descrição: O usuário pode abrir um imóvel desejado para visualizar suas informações mais detalhadas.
- **Visualizar imóvel específico no mapa**
Ator: Usuário
Descrição: O usuário pode visualizar a localização de um imóvel desejado no mapa.
- **Visualizar lista de usuários interessados em um imóvel**
Ator: Usuário
Descrição: O usuário pode visualizar uma lista que contém os usuários que demonstraram interesse em um determinado imóvel.

- **Visualizar perfil de outro usuário**

Ator: Usuário

Descrição: O usuário pode visualizar e analisar o perfil de outro usuário que demonstrou interesse em um determinado imóvel.

- **Manter perfil próprio**

Ator: Usuário

Descrição: O usuário pode criar um perfil próprio com suas informações de idade e sexo, uma foto e algumas formas de contato para que outros usuários possam analisar o perfil e entrar em contato se desejar. Ele também pode editar essas informações e escolher se deseja manter algumas informações pessoais como idade e números de telefone como visíveis ou não para outros usuários.

- **Manter imóvel próprio**

Ator: Usuário

Descrição: O usuário pode cadastrar imóveis no sistema, especificando detalhes do imóvel e endereço, podendo também adicionar fotos. Também poderá editar as informações do mesmo e excluí-lo quando desejar. Caso o usuário ainda não tenha criado um perfil próprio, para cadastrar um imóvel será necessário que se crie um perfil próprio antes.

- **Contatar outro usuário interessado em dividir imóvel**

Ator: Usuário

Descrição: O usuário pode contatar algum usuário que demonstrou interesse em dividir o aluguel de um determinado imóvel. Caso o usuário ainda não tenha criado um perfil próprio, para contatar outro usuário será necessário que se crie um perfil próprio antes.

- **Contatar anunciante de imóvel**

Ator: Usuário

Descrição: O usuário pode contatar o anunciante do imóvel que ele deseja alugar ou dividir aluguel. Caso o imóvel seja da categoria de dividir aluguel, ao contatar o anunciante, automaticamente ele marca o imóvel como interesse em dividir. Caso o usuário ainda não tenha criado um perfil próprio, para contatar um anunciante será necessário que se crie um perfil próprio antes.

- **Manter interesse em dividir imóvel**

Ator: Usuário

Descrição: O usuário pode marcar e demonstrar interesse em dividir um determinado imóvel para que outros usuários que também desejam dividir este imóvel possam vê-lo e entrar em contato. Ele também pode desmarcar e deixar de demonstrar interesse. Caso o usuário ainda não tenha criado um perfil próprio, para

demonstrar interesse em um imóvel será necessário que se crie um perfil próprio antes.

4.1.2 Diagrama Entidade Relacionamento

O DER é uma representação gráfica que possibilita analisar a estrutura de um banco de dados relacional. Ele expõe as tabelas que compõem o banco com seus atributos e características e os tipos de relacionamentos entre as tabelas. Na Figura 7, pode-se observar o DER criado para o banco de dados do sistema apresentado neste trabalho através da ferramenta *MySQL Workbench*⁶.

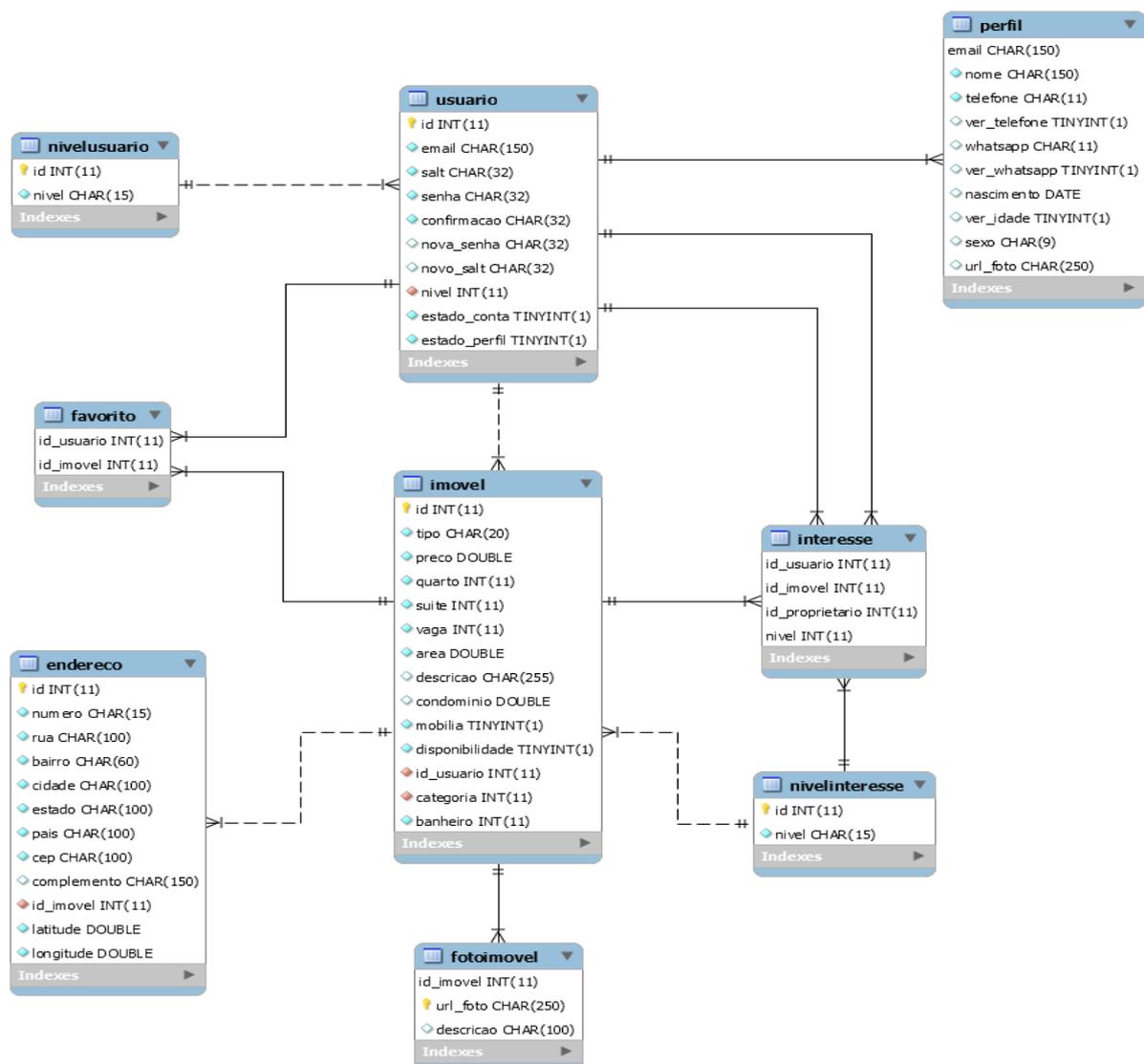


Figura 7: Diagrama Entidade Relacionamento.

Fonte: Autor

⁶ <https://www.mysql.com/products/workbench/>

Segue uma descrição das tabelas e relacionamentos que compõem este diagrama:

- **NivelUsuario**

Chave primária: id

Descrição: Esta tabela possui por padrão dois registros (comum e anunciante) que servem de referência para definir o nível de usuário cadastrado no sistema.

- **NivelInteresse**

Chave primária: id

Descrição: Essa tabela possui por padrão dois registros (alugar e dividir) que servem de referência para definir a categoria do imóvel cadastrado no sistema e para definir o tipo de interesse de um usuário em um imóvel.

- **Usuário**

Chave primária: id

Chave estrangeira: nivel - faz referência com chave primária da tabela NivelUsuario que identifica o nível do usuario no sistema.

Descrição: Esta tabela guarda os registros de usuários cadastrados no sistema.

- **Perfil**

Chave primária: email

Chave estrangeira: email - faz referência com chave primária da tabela Usuário que identifica a quem o perfil pertence.

Descrição: Esta tabela guarda os registros dos perfis dos usuários cadastrados no sistema.

- **Imovel**

Chave primária: id

Chave estrangeira: id_usuario - faz referência com chave primária da tabela Usuário que identifica a quem o imóvel pertence; categoria - faz referência com chave primária da tabela NivelInteresse que identifica a categoria (alugar ou dividir) do imóvel.

Descrição: Esta tabela guarda os registros dos imóveis cadastrados no sistema.

- **Endereco**

Chave primária: id

Chave estrangeira: id_imovel - faz referência com chave primária da tabela Imovel que identifica a qual imóvel o endereço corresponde.

Descrição: Esta tabela guarda os registros dos endereços dos imóveis cadastrados no sistema.

- **Favorito**

Chave primária: id_usuario; id_imovel

Chave estrangeira: `id_usuario` - faz referência com chave primária da tabela `Usuario` que identifica a qual usuário o registro de favorito pertence; `id_imovel` - faz referência com chave primária da tabela `Imovel` que identifica qual imóvel o registro de favorito corresponde.

Descrição: Esta tabela guarda os registros dos imóveis salvos como favoritos pelos usuários.

- **Interesse**

Chave primária: `id_usuario`; `id_imovel`; `id_proprietario`; `nivel`

Chave estrangeira: `id_usuario` - faz referência com chave primária da tabela `Usuario` que identifica a quem o registro de interesse pertence; `id_imovel` - faz referência com chave primária da tabela `Imovel` que identifica qual imóvel o registro de interesse corresponde; `id_proprietario` - faz referência com chave primária da tabela `Usuario` que identifica o proprietário do imóvel que o registro de interesse corresponde e serve para filtrar a busca dos usuários interessados nos imóveis do respectivo proprietário; `nivel` - faz referência com chave primária da tabela `NivelInteresse` que identifica o nível que o registro do interesse corresponde (alugar ou dividir).

Descrição: Esta tabela guarda os registros de todos os interesses de alugar e dividir aluguel demonstrados pelos usuários no sistema.

- **FotoImovel**

Chave primária: `id_imovel`; `url_foto`

Chave estrangeira: `id_imovel` - faz referência com chave primária da tabela `Imovel` que define a qual imóvel o endereço da foto registrada pertence.

Descrição: As fotos são salvas no *Firestore Storage* e esta tabela guarda as urls que são geradas pelo próprio *Firestore* a fim de guardar o endereço das fotos para serem carregadas no sistema.

4.2 Desenvolvimento do Sistema

4.2.1 Aplicativo

O aplicativo foi desenvolvido para plataforma *Android* considerando ser a mais utilizada pelos usuários de dispositivos móveis no mundo. Desta forma é possível alcançar com mais eficácia o público-alvo deste trabalho. Com o objetivo de auxiliar estudantes a encontrar um novo imóvel para alugar ou dividir com outras pessoas, o projeto propõe praticidade e eficiência nas buscas.

Para alcançar a finalidade desejada, o aplicativo apresenta telas simples e organizadas, buscando seguir algumas dicas do *Google Material Design*⁷, uma metodologia de design

⁷ <https://material.io/design/>

desenvolvida pela própria empresa dona do sistema *Android*, buscando trazer mais fluidez e elegância para os aplicativos da plataforma.

Para desenvolver o aplicativo foi utilizado a IDE *Android Studio*, criada também pela *Google* especificamente para este fim e utilizada a linguagem de programação *Java*.

O aplicativo é o cliente do sistema e funciona como um terminal, onde os dados exibidos são respostas às requisições ao *web service*. Quando o usuário acessa uma tela, as informações carregadas nela são feitas através de um pedido ao servidor que faz uma seleção no banco de dados e retorna para o aplicativo. Da mesma forma, ao executar uma função no *app*, as alterações provocadas por esta ação são enviadas ao *web service* para que se possa atualizar o banco. Cada requisição aberta ao sistema *web* é feita através de *thread*⁸, que só encerra quando o serviço *web* responde ou quando excede o tempo limite de conexão.

Para acompanhar e entender melhor como foi desenvolvido o *app*, são apresentadas a seguir as telas do *software* explicando sobre o desenvolvimento e o funcionamento das mesmas. As imagens das telas são *prints* tirados do aplicativo em funcionamento em um celular *Android*.

4.2.1.1 Tela Login

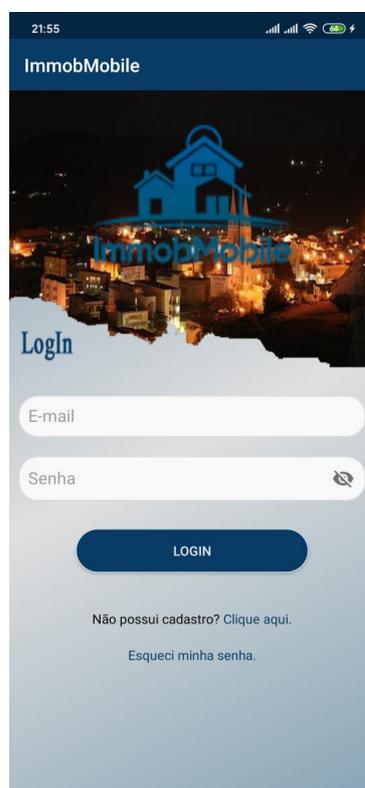


Figura 8: Tela Login.

Fonte: Autor

⁸ Uma tarefa executada em paralelo com o processo principal do sistema

Na Figura 8 é apresentada a tela de *login* do aplicativo.

Pensando na segurança do usuário, o aplicativo possui um sistema de autenticação com *e-mail* e senha. A senha na sua forma final é composta pelo conjunto de *caracteres* escolhidos pelo usuário criptografados com a tecnologia *MD5* e com a criptografia *MD5* de outro conjunto de *caracteres* gerados de forma aleatória chamado de *salt*, que juntos são criptografados novamente com o *MD5*, formando assim a senha final que será verificada pelo sistema. O *MD5* é um algoritmo unidirecional, uma técnica que após criptografar uma informação ela não pode mais retornar à sua forma original (CGRBRASIL) .

Todo este processo serve para evitar que a informação ao ser trafegada na rede, se interceptada, seja decifrada por alguém mal intencionado.

Caso o usuário ainda não tenha feito o cadastro no sistema, na tela de *login* possui um botão em forma de texto que direciona o usuário para a tela de cadastro, assim como também possui um botão em forma de texto que direciona para a tela de redefinir senha caso o usuário deseje redefinir sua senha.

4.2.1.2 Tela Cadastro



Figura 9: Tela Cadastro.

Fonte: Autor

Na Figura 9 é apresentada a tela de cadastro.

Caso o usuário não possua cadastro para fazer *login* no sistema, nesta tela é possível fazer o cadastro informando um endereço de *e-mail*, uma senha de no mínimo 6 caracteres e confirmando a senha. A senha é criptografada da mesma forma descrita na sessão que apresenta a tela de *login*.

Após efetuar o cadastro, o usuário receberá um *link* de ativação da conta no endereço de *e-mail* cadastrado. Confirmando o cadastro através deste *link*, o usuário poderá fazer *login* no sistema.

4.2.1.3 Tela Redefinir Senha



Figura 10: Tela Redefinir Senha.

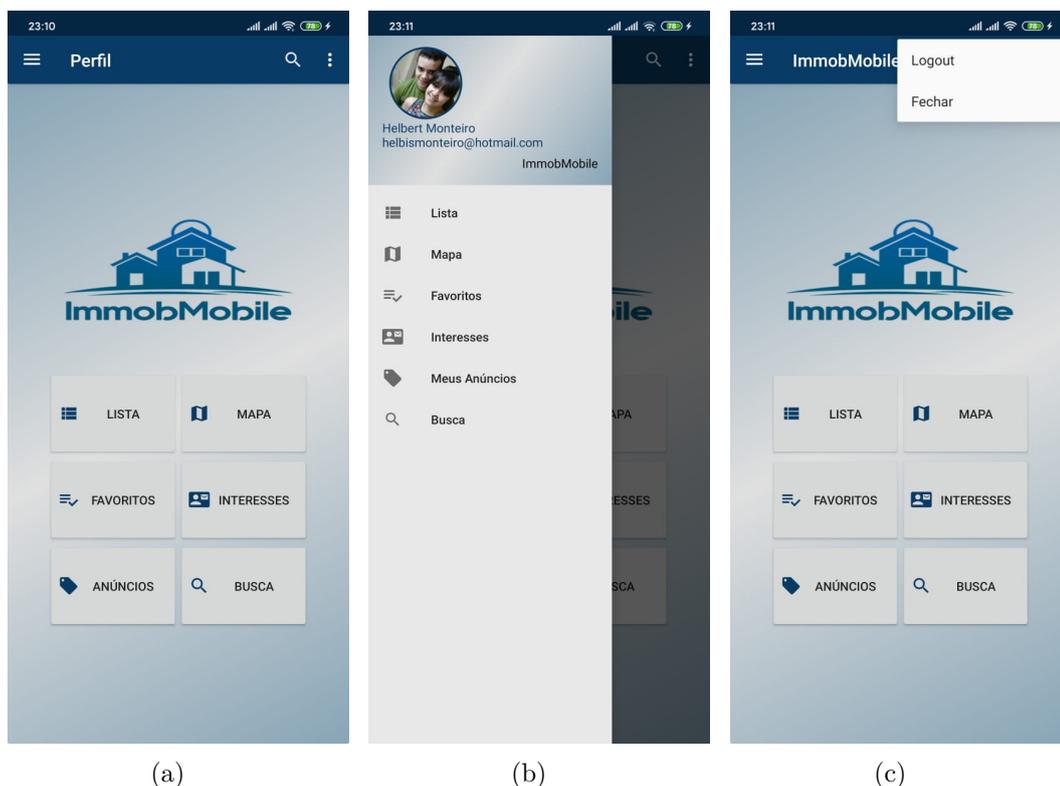
Fonte: Autor

Na Figura 10 é apresentada a tela de redefinir senha.

Semelhante à tela de cadastro, na tela de redefinir a senha o usuário pode informar o seu *e-mail* que foi cadastrado e a nova senha desejada seguida pela confirmação da nova

senha. Após confirmar no botão da tela, o usuário receberá um *link* no *e-mail* cadastrado para confirmar a redefinição de senha. Aqui também é utilizada a tecnologia *MD5* da mesma forma que na tela de *login* e de cadastro para proteger a senha do usuário.

4.2.1.4 Tela Inicial



(a)

(b)

(c)

Figura 11: Tela Inicial.

Fonte: Autor

Na Figura 11 é apresentada a tela inicial com seus menus.

A tela inicial é composta por um menu que leva a outras telas de visualização dos anúncios de imóveis e a tela de busca. Além do menu principal (Figura 11-a), também possui um menu lateral com os mesmos itens e acesso para a tela de perfil (Figura 11-b) e um menu na barra superior da tela com os itens de busca, *logout* e fechar o *app* (Figura 11-c).

A tela inicial é a *Activity*⁹ principal do aplicativo e as que são acessadas pelo menu desta tela e a tela de perfil são *Fragments*¹⁰ carregados na tela principal.

⁹ Uma tela carregada de forma independente no *Android*

¹⁰ Uma tela carregada dentro de uma *Activity*

4.2.1.5 Tela Perfil

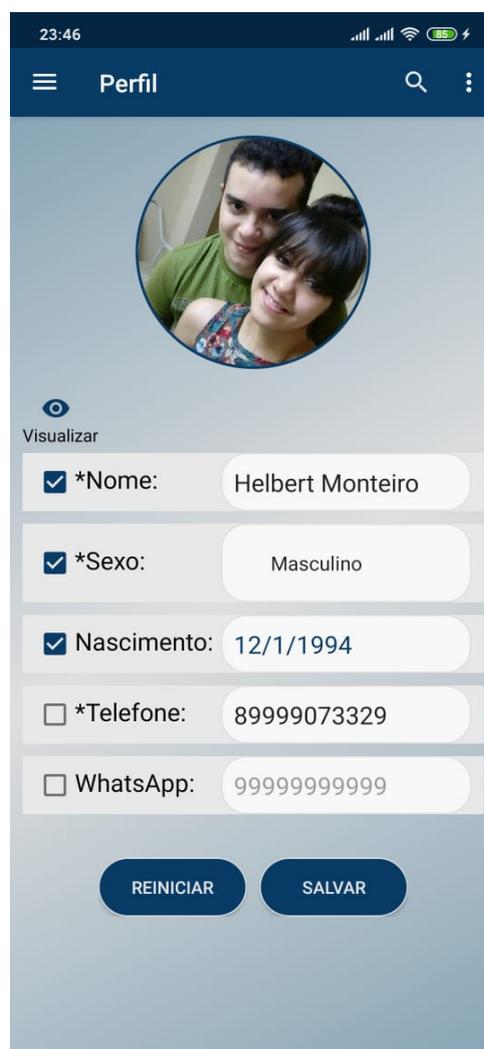


Figura 12: Tela Perfil.

Fonte: Autor

A tela de perfil é um *Fragment* que é carregado na *Activity* principal. Na Figura 12 é apresentada a tela de perfil.

Para que outros usuários tenham o mínimo de informações interessantes para analisar se determinado usuário é uma pessoa qualificada por suas preferências para conviver dividindo um imóvel, cada usuário tem um perfil com informações básicas que podem ser gerenciadas através da tela de perfil.

Informações como número de telefone, *whatsapp*¹¹ e idade não são registros obrigatórios para dar a escolha ao usuário de manter tais informações pessoais em privacidade. Porém, nome e sexo são obrigatórios, pois é o mínimo para que outros usuários tenham como analisar o perfil.

¹¹ <https://www.whatsapp.com>

O usuário também pode cadastrar seus números de contato e idade, mas se desejar, pode desmarcá-los nos *checkbox* à esquerda individualmente para torná-los invisíveis para outros usuários caso deseje manter a privacidade destas informações. A ideia de cadastrar estas informações de contato mesmo que não queira que sejam vistas publicamente é para que o usuário possa usá-las quando for entrar em contato com outro usuário ou um anunciante. Nome e sexo assim como são registros obrigatórios, também são obrigatoriamente visíveis para outros usuários.

O usuário também pode cadastrar uma foto de perfil recuperada de sua galeria de fotos.

4.2.1.6 Tela Lista



Figura 13: Tela Lista.

Fonte: Autor

Na Figura 13 é apresentada a tela lista.

A tela lista também é um *Fragment* da *Activity* principal. Nesta tela é carregada todos os imóveis cadastrados no sistema que estão registrados como disponível e apresentados

em uma lista. Esta tela também pode ser carregada com os imóveis filtrados em uma busca feita pelo usuário na tela de busca.

A lista é exibida em um *RecyclerView*¹² e foi escolhido este componente pois é mais flexível à personalização de exibição do conteúdo de um item que o componente *ListView*¹³

O usuário pode clicar em qualquer imóvel exibido na lista, se feito, ele será direcionado para a tela do imóvel.

4.2.1.7 Tela Mapa

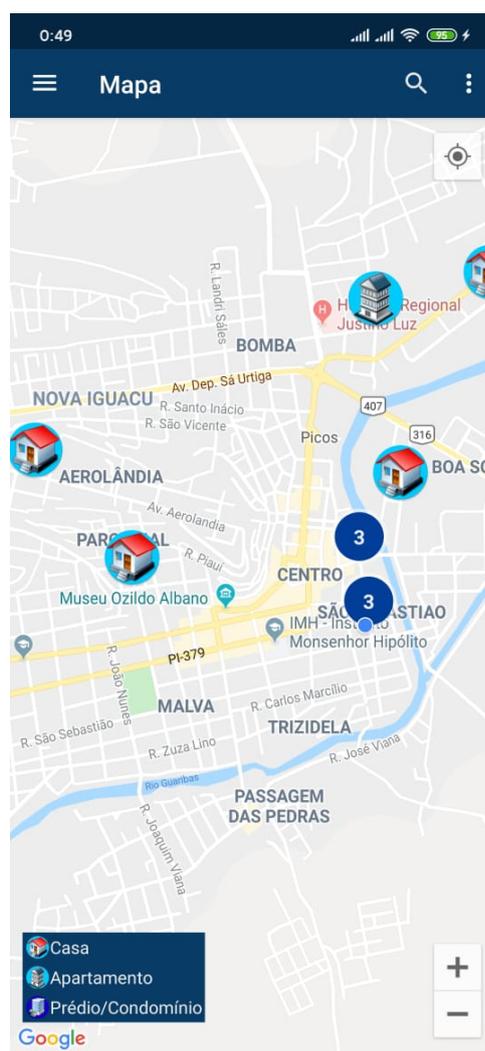


Figura 14: Tela Mapa.

Fonte: Autor

Na Figura 14 é exibida a tela do mapa.

A tela do mapa também é um *Fragment* da *Activity* principal. Nesta tela é carregada todos os imóveis cadastrados no sistema que estão registrados como disponível e apresen-

¹² Componente *Android* de fácil personalização para exibir conteúdos na tela.

¹³ Componente *Android* usado para exibir informações em forma de lista.

tados como marcadores no mapa. Esta tela também pode ser carregada com os imóveis filtrados em uma busca feita pelo usuário na tela de busca.

O mapa é um *Fragment* carregado através da API do *Google Maps*. Para poder usar o *Google Maps* em uma aplicação é necessário obter uma chave. A API permite personalizar o mapa e para tornar mais temático à ideia da proposta com imóveis, foi alterado os ícones dos marcadores exibidos no mapa. Foi utilizado um ícone de casa para representar as casas e um ícone de prédio para representar os apartamentos.

Para melhorar a visualização dos marcadores quando muitos são carregados em uma localização próxima, foi implementado um gerenciador de clusterização dos marcadores também disponível na API do *Google Maps*, que agrupa os marcadores em níveis menores de zoom evitando que eles se sobreponham e os separa em níveis maiores de zoom. Desta forma, os marcadores estão sempre visíveis separadamente tornando a visualização do mapa melhor e mais organizada.

Os imóveis são localizados no mapa através de suas coordenadas obtidas no cadastro do endereço pela API *Geocoding* do Google.

O usuário pode clicar em qualquer imóvel exibido na mapa, se feito, ele será direcionado para a tela do imóvel.

4.2.1.8 Tela Favoritos



Figura 15: Tela Favoritos.

Fonte: Autor

Na Figura 15 é apresentada a tela favoritos.

A tela favoritos é um *Fragment* da *Activity* principal. Nela é carregada os imóveis marcados como favoritos pelo usuário e exibidos em lista. O usuário também pode selecionar um imóvel da lista para visualizá-lo.

4.2.1.9 Tela Interesses

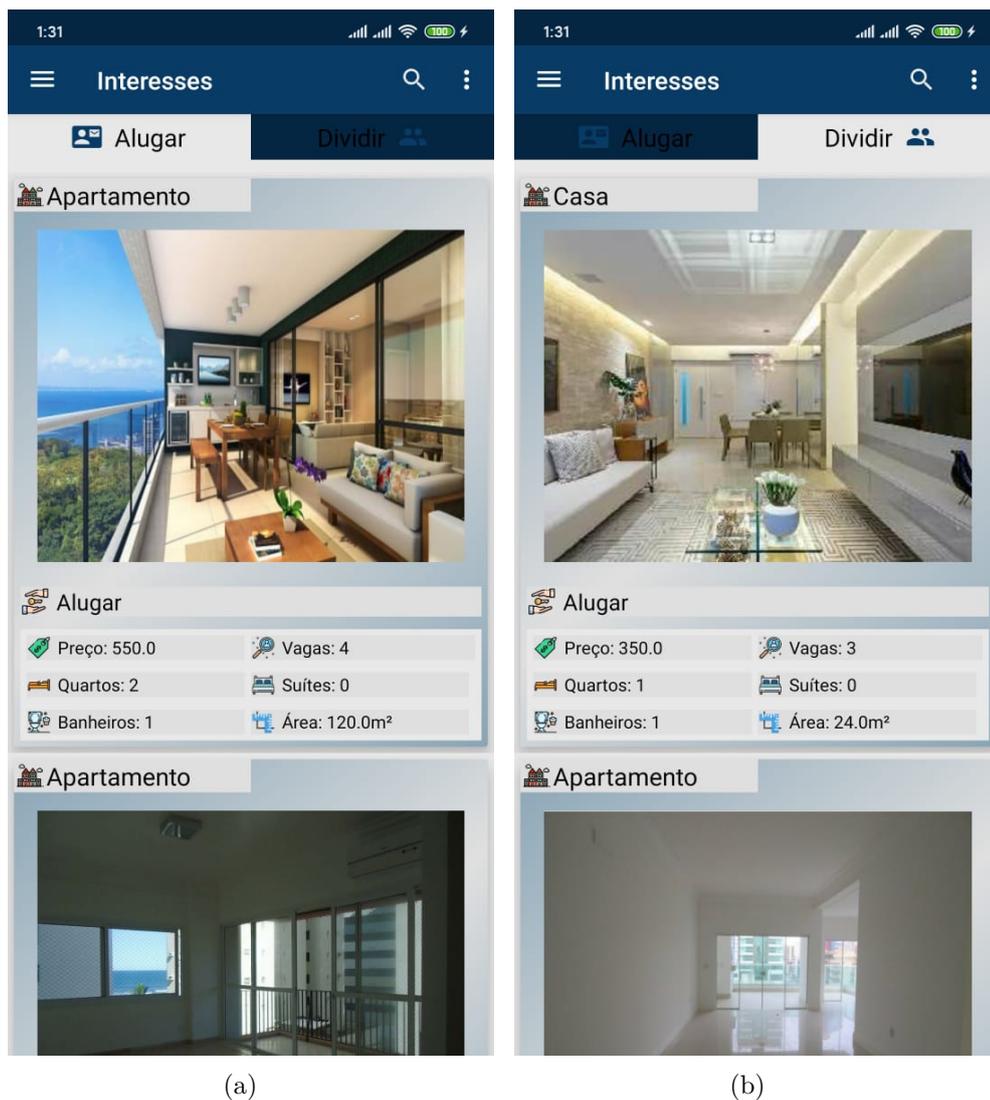


Figura 16: *Tela Interesses*.

Fonte: Autor

Na Figura 16 é exibida a tela interesses.

A tela interesses é um *Fragment* da *Activity* principal. Nela é exibida os imóveis que o usuário demonstrou interesse, seja para alugar (Figura 16-a) quando contactou um anunciante de imóvel para alugar ou interesse para dividir (Figura 16-b) quando contactou um anunciante de imóvel para dividir ou quando marcou a opção de demonstrar interesse em dividir para que outros usuários possam vê-lo na lista de interessados em dividir o imóvel.

4.2.1.10 Tela Meus Anúncios

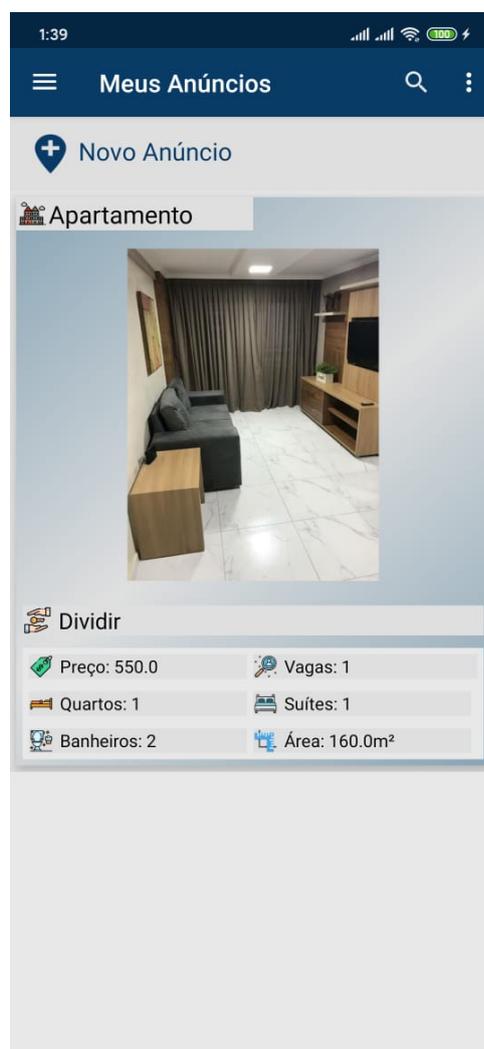


Figura 17: Tela Meus Anúncios.

Fonte: Autor

Na Figura 17 é exibida a tela meus anúncios.

A tela meus anúncios é um *Fragment* da *Activity* principal. Nela é exibida a lista dos imóveis que o usuário cadastrou no sistema. O usuário pode clicar no botão Novo Anúncio para ser direcionado à tela de cadastro de um novo imóvel.

Clicando em algum imóvel da lista, o usuário será direcionado para a tela de gerência do imóvel, que possibilita editar as informações do imóvel ou excluí-lo.

4.2.1.11 Tela Busca

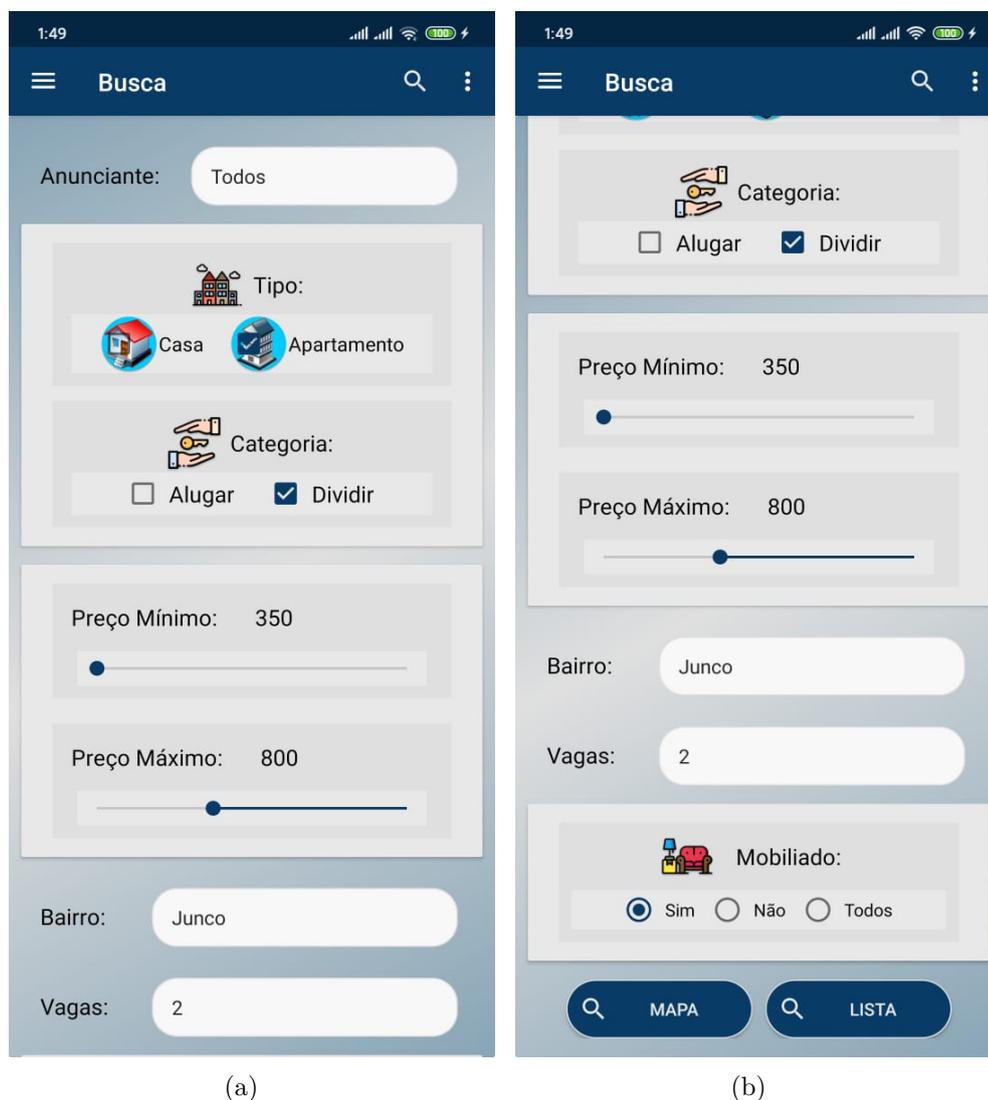


Figura 18: Tela Busca.

Fonte: Autor

Na Figura 18 é exibida a tela de busca.

A tela de busca é um *Fragment* da *Activity* principal. Por ela o usuário pode fazer uma busca personalizada usando as informações desejadas para encontrar o imóvel ideal. Os campos da tela de busca são carregados a partir dos dados dos imóveis cadastrados no sistema, dessa forma aumenta as chances de sucesso de encontrar um ou mais imóveis relativos às informações selecionadas, já que todos os dados dos campos são existentes no banco de dados.

O usuário pode exibir o resultado da busca tanto na tela de mapa como na tela de lista através dos botões MAPA e LISTA que são vistos na Figura 18-b.

4.2.1.12 Tela Cadastro de Imóvel

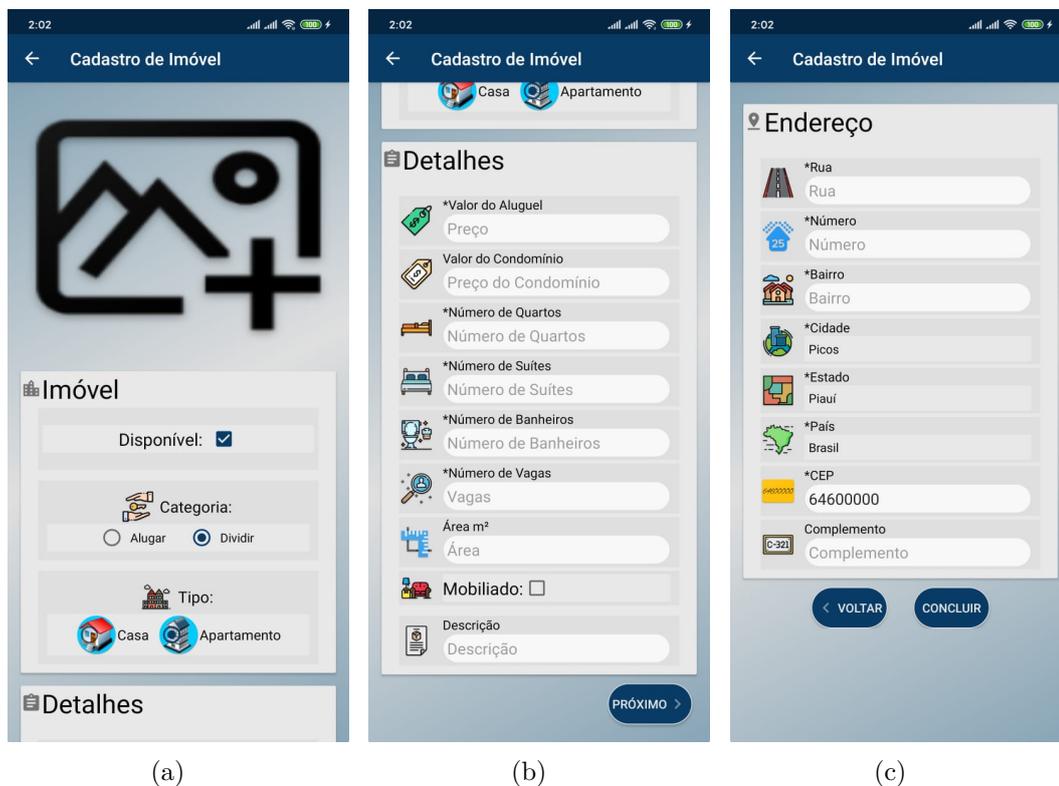


Figura 19: Tela Cadastro de Imóvel.

Fonte: Autor

Na Figura 19 é exibida a tela de cadastro de imóvel.

Na tela cadastro de imóvel o usuário pode cadastrar um imóvel para alugar ou para dividir aluguel preenchendo todas as informações relevantes para que outros usuários possam analisar e se interessar.

Além das informações características do imóvel (Figuras 19-a e 19-b), o usuário também deve cadastrar o endereço (Figura 19-c), tanto para os outros usuários saberem a localização como também para que o imóvel possa ser exibido no mapa. Quando o usuário confirma o cadastro do imóvel, as coordenadas globais da localização do imóvel são carregadas através da API *Geocoding* do google e as coordenadas são salvas junto com as informações de endereço para serem carregadas posteriormente para exibição do mapa.

O usuário pode também cadastrar fotos do imóvel para melhorar ainda mais a visualização e análise pelos outros usuários e atrair mais interessados para o anúncio.

Por padrão os campos de cidade, estado e país no cadastro de endereço já são preenchidos para a cidade de Picos e não podem ser editados, já que o aplicativo, apesar de poder ser usado de qualquer lugar do mundo, é de funcionamento exclusivo para esta cidade.

4.2.1.13 Tela Gerência de Imóvel

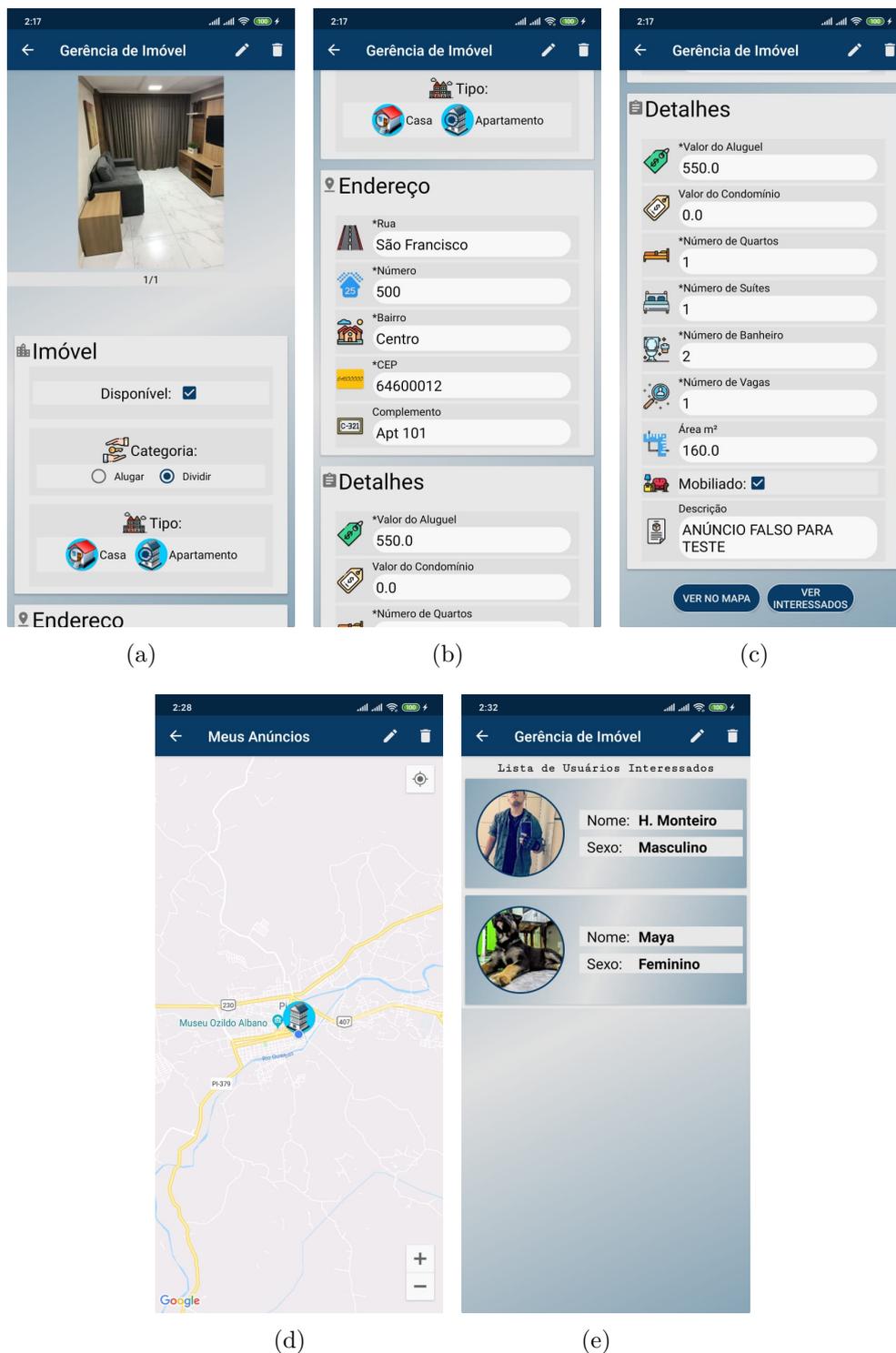


Figura 20: Tela Gerência de Imóvel.

Fonte: Autor

Na Figura 20 é apresentada a tela de gerência de imóvel.

Na tela gerência de imóvel o usuário pode gerenciar um imóvel cadastrado, editando as informações ou o excluindo (Figuras 20-a, 20-b e 20-c).

Da mesma forma como no cadastro, quando o usuário edita o endereço, também é feita uma chamada da API *Geocoding* para buscar as coordenadas e também serem atualizadas no banco de dados.

Aqui o usuário pode ver a localização do seu imóvel isolado no mapa (Figura 20-d). Ele também pode visualizar a lista dos usuários que demonstraram interesse em seu imóvel (Figura 20-e), seja para alugar ou dividir, dependendo da categoria que o imóvel pertence. Selecionando um usuário desta lista de interessados, ele pode visualizar e analisar o perfil deste usuário interessado e o contatar através das informações de contato cadastradas.

O usuário também pode excluir ou adicionar fotos do imóvel.

4.2.1.14 Tela Imóvel

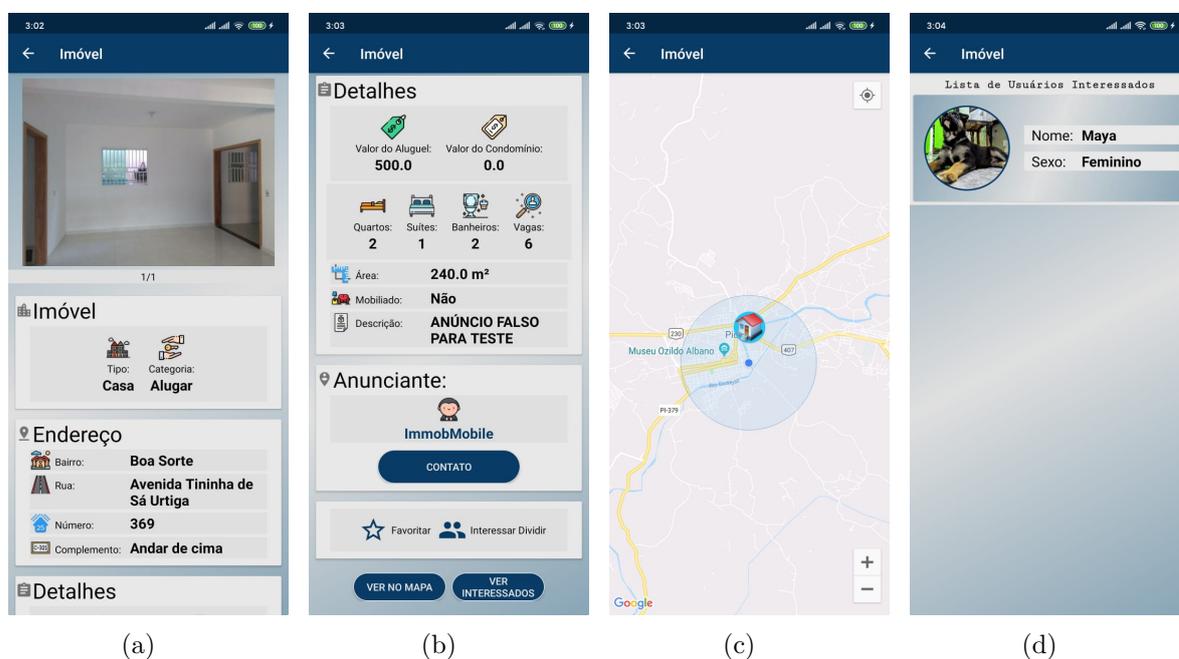


Figura 21: Tela Imóvel.

Fonte: Autor

Na Figura 21 é apresentada a tela imóvel.

Na tela imóvel é possível visualizar as informações e fotos de um imóvel (Figuras 21-a e 21-b). É possível também visualizar a localização isolada do imóvel no mapa (Figura 21-c).

Após analisar o imóvel, caso o usuário se interesse em alugar ou dividir, dependendo da categoria do imóvel, ele pode entrar em contato com o anunciante clicando no botão Contato ou clicando no nome do anunciante para observar seu perfil antes.

Caso o usuário tenha gostado das informações do imóvel mas ainda não decidiu entrar em contato com o anunciante, ele pode marcar a opção Favoritar para encontrar este imóvel com mais facilidade futuramente pela tela favoritos.

Se o usuário não se interessa em alugar o imóvel, mas deseja dividir com outro usuário, ele pode marcar a opção Interessar em Dividir para que outros usuários que visualizarem este imóvel possam encontrá-lo na lista de usuários interessados.

Ele pode verificar também na lista de usuários interessados se tem alguém interessado em dividir o imóvel (Figura 21-d). Poderá visualizar os perfis dos usuários da lista e entrar em contato com algum deles para combinar de dividir o aluguel.

4.2.1.15 Perfil de Outros Usuários e Contato

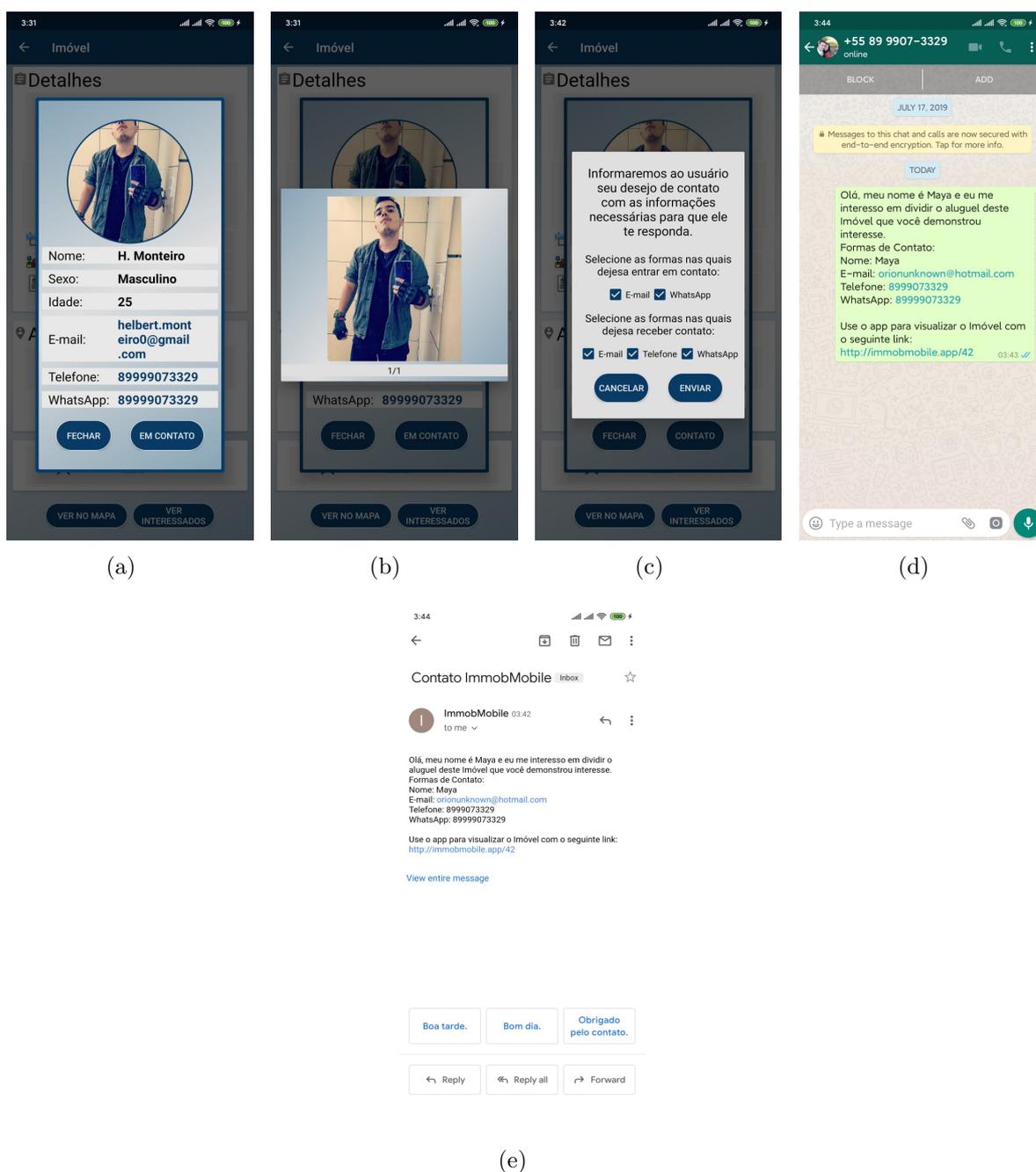


Figura 22: Perfil de outro usuário e Contato.

Fonte: Autor

Na Figura 22 são apresentadas as telas de perfil de outro usuário e de contato, assim como as mensagens entregues nos meios de contato cadastrados pelo usuário.

Esta tela é um *AlertDialog* criado dentro da *Activity* de imóvel. Ela é exibida quando o usuário toca no nome do anunciante ou seleciona um usuário da lista de usuários interessados.

Através desta tela (Figura 22-a), o usuário pode analisar o perfil e a foto de perfil (Figura 22-b) e entrar em contato tocando no botão Contato, onde será exibida uma tela (Figura 22-c) para selecionar as formas de como deseja contatar e como deseja receber contato de volta. Ele pode também tocar nos números de contato ou no *e-mail* e automaticamente será direcionado para o aplicativo de telefone para fazer uma chamada, ou para o *WhatsApp* para enviar uma mensagem direta por lá ou para o aplicativo de *e-mail*.

Quando enviada a mensagem, o outro usuário receberá a mensagem com as informações de contato para retornar e com um *link* (Figuras 22-d e 22-e), que ao tocar nele, será aberto pelo aplicativo a tela do imóvel correspondente ao contato da mensagem.

4.2.2 Banco de Dados

O banco de dados feito para este projeto foi desenvolvido no MySQL. Trata-se de um banco de dados relacional e possui 9 tabelas já detalhadas e representadas pelo DER apresentado na sessão 4.1.2 deste trabalho.

Neste projeto o banco de dados se encarrega de armazenar todos os dados do sistema, sendo nenhum dado armazenado no aplicativo.

Para que o aplicativo insira, exclua, atualize e recupere dados do banco, é feita uma requisição ao *web service* que realiza consultas e comandos pela classe de conexão com o banco de dados.

4.2.3 Web Service

O *web service* deste trabalho foi desenvolvido na linguagem Java com a IDE NetBeans.

O *NetBeans* é uma IDE com bastante recursos e o projeto base deste *web service* foi gerado pela própria IDE.

Adotando o conceito de *RESTful*, foram implementados todos os métodos com seus endereços *web* necessários para a comunicação do aplicativo com o banco de dados.

O aplicativo acessa o endereço do método *HTTP* desejado, o *web service* recupera ou atualiza dados do banco e retorna para o aplicativo a resposta correspondente ao método acionado pela requisição.

A troca de informações entre o aplicativo e o *web service* é feita com *JSON*¹⁴, onde os dois lados utilizam a biblioteca *Gson*¹⁵ para converter o texto em *JSON* para um objeto

¹⁴ Um modelo de texto usado para transmitir pequenos e grandes volumes de informação pela *web*

¹⁵ <https://github.com/google/gson>

de uma classe modelo.

É o *web service* que também se encarrega de enviar os *e-mails* do sistema, como por exemplo, os *e-mails* de confirmação de senha, confirmação de redefinição de senha e também os *e-mails* de contato entre os usuários. Todos os *e-mails* são enviados pelo *web service* para não precisar deixar informações sigilosas do *e-mail* do sistema no código fonte do aplicativo.

5 Avaliação de Usabilidade

Este capítulo apresentará o teste de usabilidade realizado com usuários que se encaixam no público-alvo deste projeto. O teste foi feito através de um questionário realizado na ferramenta Google Forms¹, respondido pelos usuários após serem instruídos a realizar 6 tarefas dentro do aplicativo. Esse questionário e sua avaliação final baseia-se e utiliza o método *SUS* (*System Usability Scale*), uma metodologia bastante usada para avaliar a usabilidade de *softwares* e outros produtos e serviços.

5.1 Testes Realizados

A usabilidade é a medida em que usuários específicos no uso de um produto ou serviço realizam objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado (ISO9241-11, 2018). Seguindo esta definição, foram selecionadas 18 pessoas entre estudantes da UFPI - Campus Senador Helvídeo Nunes de Barros, estudantes que moram em outras cidades e outras pessoas da cidade de Picos.

Estas pessoas foram instruídas a realizar as seguintes tarefas dentro do aplicativo:

1. Encontrar um imóvel de sua preferência para alugar;
2. Entrar em contato com o Proprietário do imóvel que se interessa em alugar;
3. Encontrar um imóvel de sua preferência para dividir aluguel com outras pessoas;
4. Demonstrar interesse em dividir aluguel do imóvel de sua preferência;
5. Encontrar um imóvel que já existem pessoas interessadas em dividir o aluguel e entrar em contato com alguma das pessoas da lista;
6. Criar um anúncio (para teste) de um imóvel em Picos para dividir aluguel para que outras pessoas possam encontrar seu anúncio;

Após realizarem as tarefas, elas foram instruídas a responder o seguinte questionário de 10 questões com respostas na escala *Likert*, que são alternativas que vão de 1 a 5, onde 1 representa “Discordo Completamente” e 5 representa “Concordo Completamente”:

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.

¹ <https://www.google.com/forms/about/>

4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Estas são as 10 questões padrões do método *SUS* e através das respostas dos usuários é possível analisar a eficácia, eficiência e satisfação no uso do aplicativo.

5.2 Resultados Obtidos

O resultado final proposto pelo método *SUS* é um cálculo realizado pelos valores das respostas dos usuários. A média final da avaliação pode variar de 0 a 100 pontos, sendo 68 a média de aprovação do teste. Essa média de aprovação foi estabelecida pela comunidade de pesquisa, que, pela maturidade do método *SUS*, pôde estipular este valor (BARBOZA, 2019).

As figuras a seguir apresentam as tabelas que mostram os resultados das respostas de cada pergunta.

Eu acho que gostaria de usar esse sistema com frequência.

18 respostas

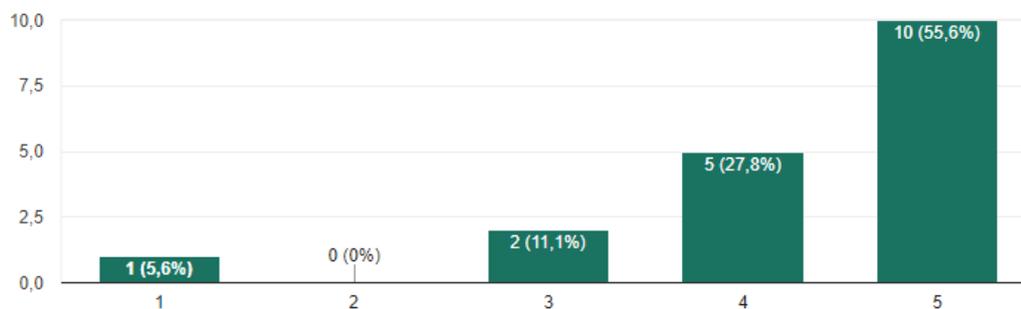


Figura 23: Resposta da questão 1 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu acho o sistema desnecessariamente complexo.

18 respostas

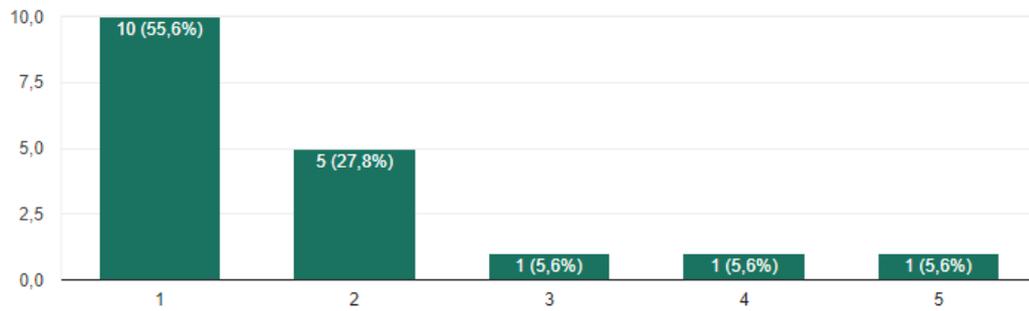


Figura 24: Resposta da questão 2 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu achei o sistema fácil de usar.

18 respostas

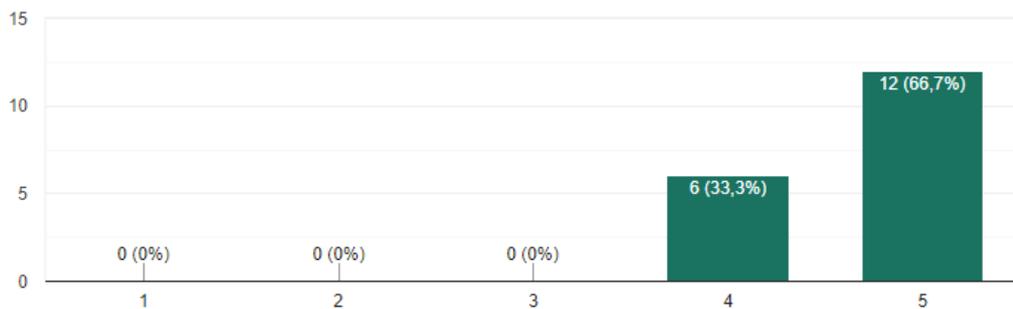


Figura 25: Resposta da questão 3 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

18 respostas

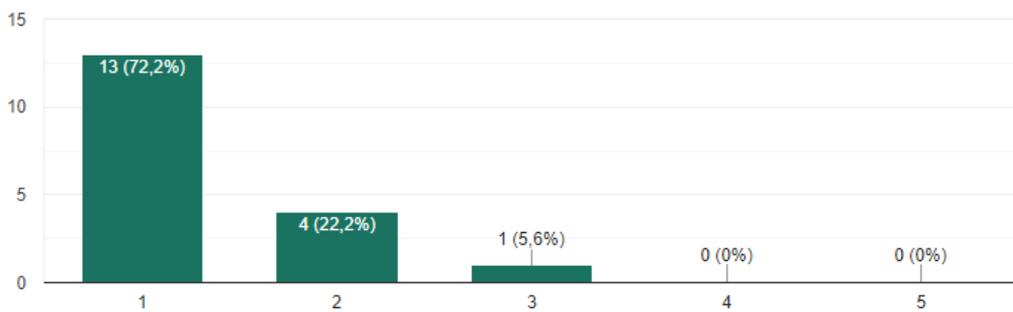


Figura 26: Resposta da questão 4 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu acho que as várias funções do sistema estão muito bem integradas.

18 respostas

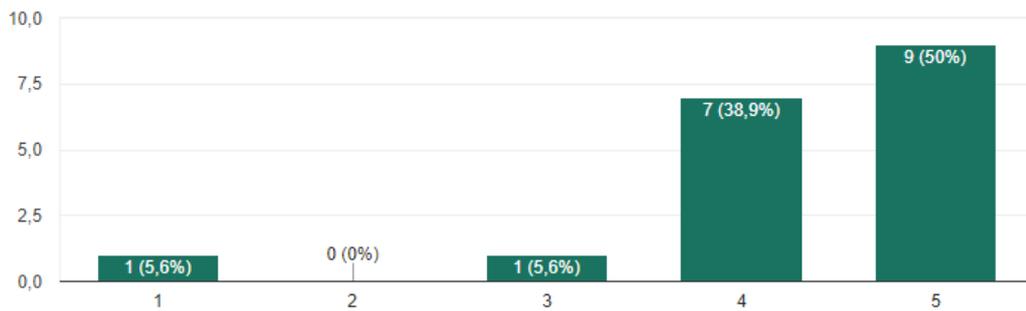


Figura 27: Resposta da questão 5 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu acho que o sistema apresenta muita inconsistência.

18 respostas

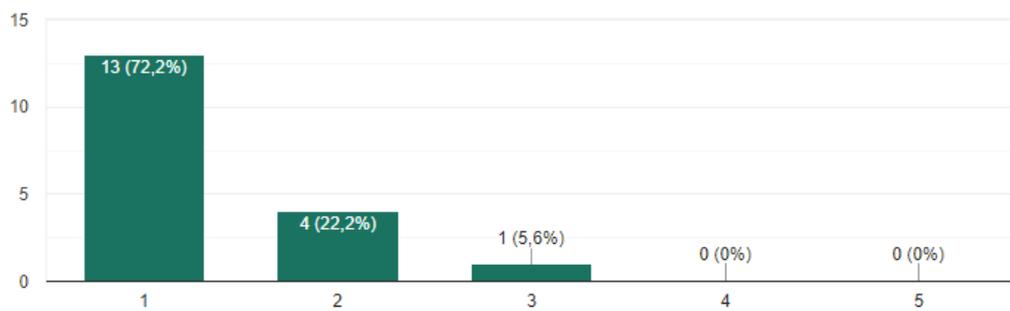


Figura 28: Resposta da questão 6 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

18 respostas

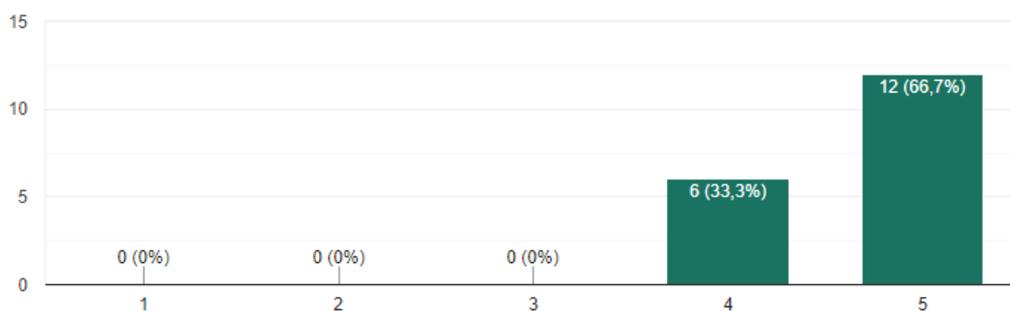


Figura 29: Resposta da questão 7 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu achei o sistema atrapalhado de usar.

18 respostas

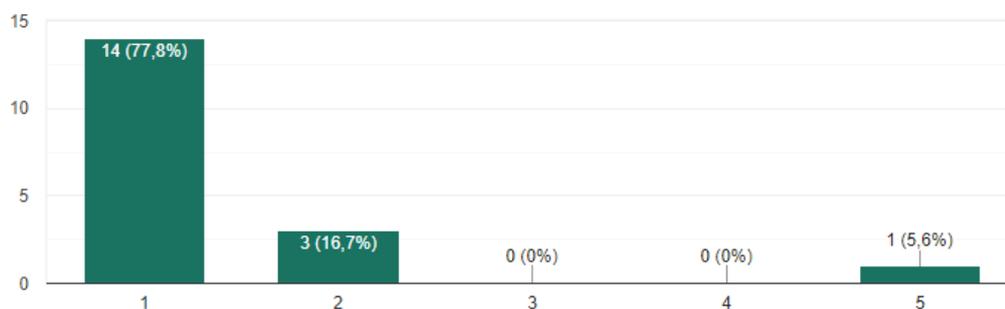


Figura 30: Resposta da questão 8 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu me senti confiante ao usar o sistema.

18 respostas

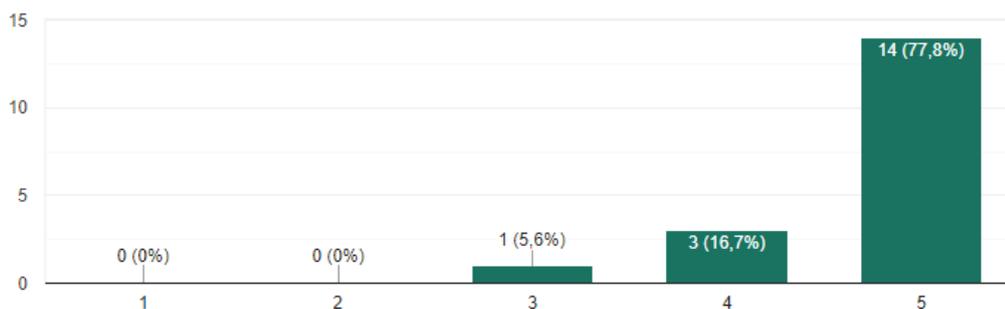


Figura 31: Resposta da questão 9 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

18 respostas

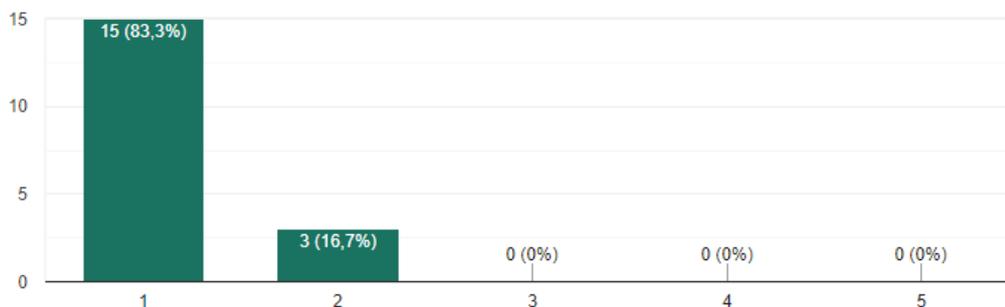


Figura 32: Resposta da questão 10 do questionário de avaliação de usabilidade.

Fonte: Gerado pelo Google Forms

Para obter o resultado, para cada resposta dos participantes são efetuados os seguintes cálculos de forma individual:

- **Para as questões ímpares:** subtrai-se 1 da pontuação que o usuário respondeu.
- **Para as questões pares:** subtrai-se a resposta de 5. Ou seja, se o usuário respondeu 2, contabiliza-se 3. Se o usuário respondeu 4, contabiliza-se 1.
- **Resultado:** Todos os valores das dez respostas são somados e multiplicados por 2,5.

Extraídas as respostas do questionário, os resultados individuais dos usuário após efetuados os cálculos apresentados acima foram os seguintes:

- | | |
|---------------------------------|----------------------------------|
| • Usuário 1: 95 pontos | • Usuário 10: 70 pontos |
| • Usuário 2: 97,5 pontos | • Usuário 11: 97,5 pontos |
| • Usuário 3: 80 pontos | • Usuário 12: 100 pontos |
| • Usuário 4: 85 pontos | • Usuário 13: 80 pontos |
| • Usuário 5: 100 pontos | • Usuário 14: 75 pontos |
| • Usuário 6: 100 pontos | • Usuário 15: 82,5 pontos |
| • Usuário 7: 90 pontos | • Usuário 16: 87,5 pontos |
| • Usuário 8: 100 pontos | • Usuário 17: 85 pontos |
| • Usuário 9: 90 pontos | • Usuário 18: 77,5 pontos |

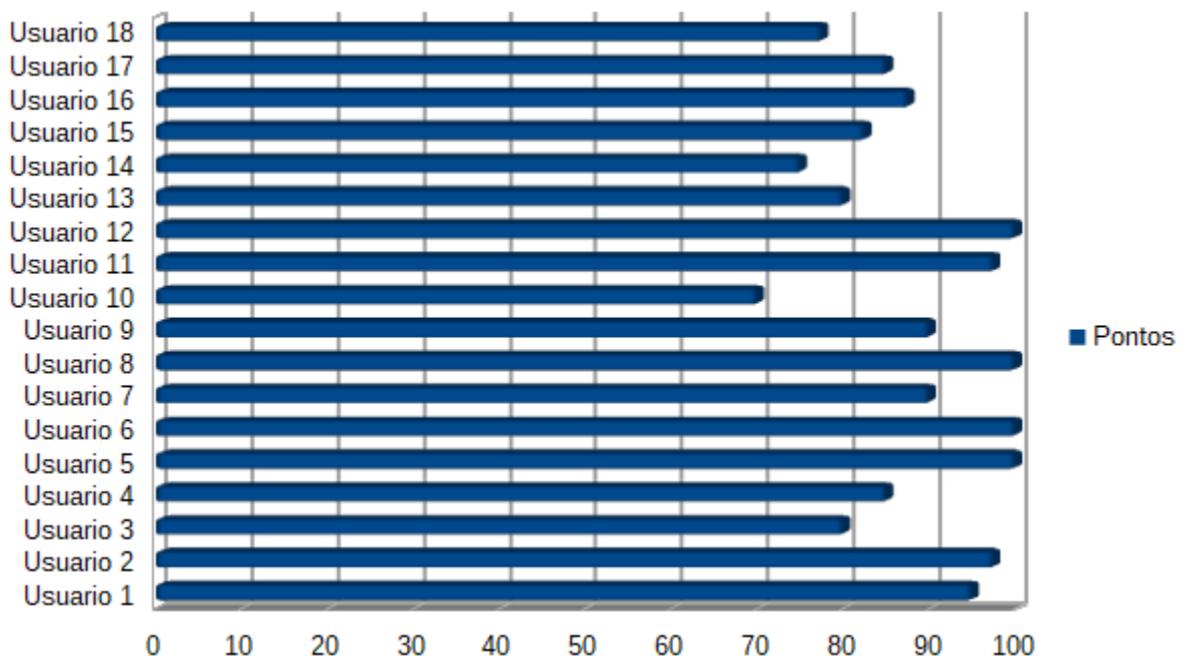


Figura 33: Pontuação da avaliação dos usuários.

Fonte: Autor

Calculando a média dos resultados obtidos, o resultado final para o teste é de 88,47 pontos, superando a média de aprovação de 68 pontos. Com isto, a avaliação de usabilidade deste trabalho apresenta um resultado satisfatório e favorável.

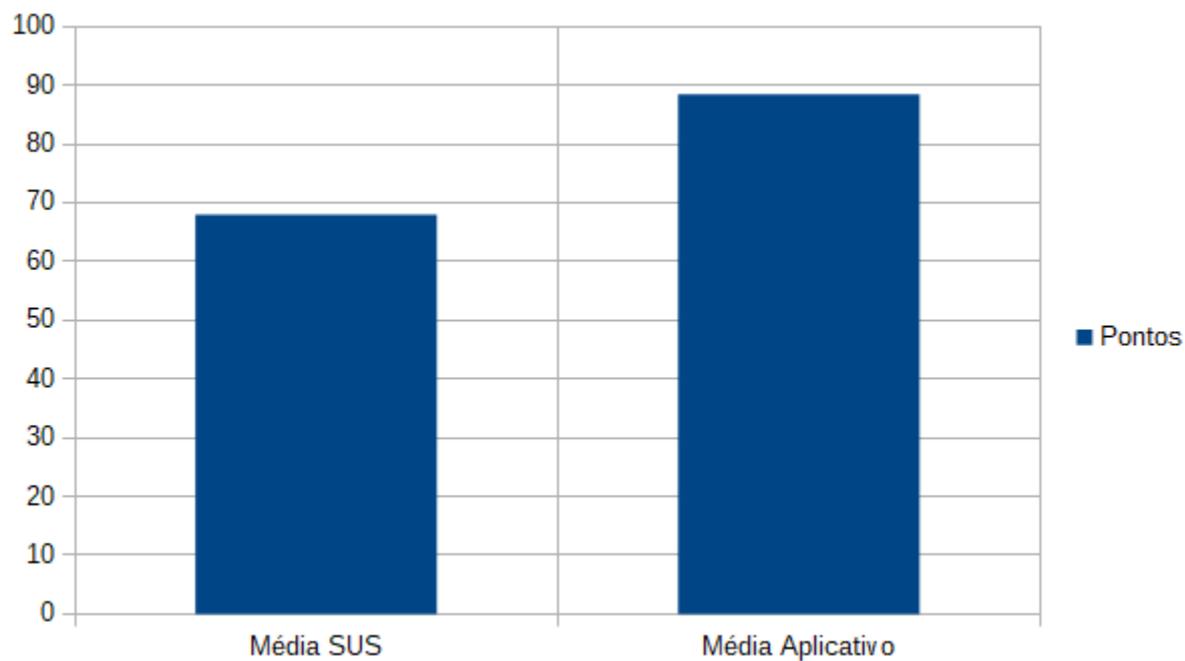


Figura 34: Resultado da avaliação.

Fonte: Autor

6 Considerações Finais

A proposta apresentada neste trabalho, com objetivo de auxiliar estudantes na busca de imóveis para alugar e dividir aluguel na cidade de Picos, através de um aplicativo *mobile* desenvolvido para plataforma *Android*, obteve resultados satisfatórios que puderam ser analisados pelos testes realizados com estudantes que moram na cidade de Picos e em outras cidades. Foi possível observar também, através dos testes realizados que tiveram participação de pessoas que não são estudantes, que o aplicativo favorece não só à classe estudantil, mas à população em geral, especificamente àqueles que se enquadram no grupo de pessoas que moram ou pretendem morar em imóveis alugados.

O desenvolvimento deste aplicativo juntamente com o *web service* e o banco de dados que compõem o sistema na sua totalidade, contribuíram de forma abundante para o aprendizado e conhecimento de tecnologias relacionadas ao campo de estudo de sistemas distribuídos, bancos de dados, redes de computadores, entre outras áreas.

Foram realizadas inúmeras pesquisas e estudos para superar as dificuldades encontradas durante o desenvolvimento deste sistema, além da necessidade de montar um servidor caseiro, um servidor *web* implantado em um *notebook* com *Windows Server 2016* em casa com o objetivo de hospedar o *web service*, junto com o banco de dados para poder realizar os testes *online*.

Na sua integridade, o trabalho apresentado demonstra alcançar êxito nos seus objetivos e ensejo de expansão e ampliação de funcionalidades e público-alvo.

6.1 Trabalhos Futuros

Os trabalhos futuros desejados para este projeto são:

- Expandir o sistema para funcionamento no contexto global, não somente na cidade de Picos;
- Reverter os pontos negativos observados na avaliação de usabilidade;
- Melhorar e ampliar os filtros de busca para proporcionar mais eficiência da funcionalidade;
- Integrar as APIs de autenticação do *Google* e *Facebook*;
- Utilizar um serviço de hospedagem adequado para um funcionamento eficiente e seguro;
- Lançar o aplicativo no mercado

Referências

- ALVES, F. H. et al. Criação de um mecanismo para encapsulamento e disponibilização de funções para manipulação de imagens digitais usando um web service restful. Universidade Federal Rural do Semi-Árido, 2019. Citado na página 24.
- ANDROID. *App security improvement program | Android Developers*. Disponível em: <<https://developer.android.com/google/play/asi>>. Acesso em: 13 nov. 2019. Nenhuma citação no texto.
- ANDROID. *Arquitetura da Plataforma | Android Developers*. Disponível em: <<https://developer.android.com/guide/platform?hl=pt-br>>. Acesso em: 07 nov. 2019. Nenhuma citação no texto.
- ANDROID. *Tudo de que você precisa para criar aplicativos no Android | Android Developers*. Disponível em: <<https://developer.android.com/studio/features?hl=pt-br>>. Acesso em: 13 nov. 2019. Nenhuma citação no texto.
- BARBOZA, A. *Medindo a usabilidade do seu produto com System Usability Scale (SUS)*. 2019. Disponível em: <<https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>>. Citado na página 57.
- BARROS, T. Android studio é o programa do google para desenvolver apps para android. 2013. Citado na página 20.
- BRITO, E. *Java: Entenda para que serve o software e os problemas da sua ausência*. 2014. Disponível em: <<https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2014/11/java-entenda-para-que-serve-o-software-e-os-problemas-da-sua-ausencia.html>>. Acesso em: 16 nov. 2019. Citado na página 22.
- CARDOSO, B. *Desenvolvimento Desktop: NetBeans IDE*. 2013. Disponível em: <<http://infoeprog.blogspot.com/2013/03/desenvolvimento-desktop-netbeans-ide.html>>. Acesso em: 17 nov. 2019. Citado na página 25.
- CGRBRASIL. *Criptografia de Dados e Senhas em MD5*. Disponível em: <<https://cgrbrasil.com.br/artigos/criptografia-de-dados-e-senhas-em-md5/>>. Acesso em: 20 nov. 2019. Nenhuma citação no texto.
- CORREIA, K. d. S. *Evolução arquitetural de um web service: Transformação de código e avaliação da arquitetura*. 2005. Citado na página 24.
- FIREBASE. *Cloud Storage | Firebase*. Disponível em: <<https://firebase.google.com/docs/storage/?hl=pt-br>>. Acesso em: 14 nov. 2019. Nenhuma citação no texto.
- FIREBASE. *Criar uma referência do Storage no Android | Firebase*. Disponível em: <<https://firebase.google.com/docs/storage/android/create-reference?hl=pt-BR>>. Acesso em: 14 nov. 2019. Nenhuma citação no texto.

- GOOGLE. *Maps SDK for Android Get started*. Disponível em: <<https://developers.google.com/maps/documentation/android-sdk/intro?hl=pt-br>>. Acesso em: 18 nov. 2019. Nenhuma citação no texto.
- ISO9241-11. *ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. [S.l.: s.n.], 2018. 29 p. (2ª Edição). Citado na página 56.
- JAVA. *Obtenha Informações sobre a Tecnologia Java*. Disponível em: <https://www.java.com/pt_BR/about/>. Acesso em: 16 nov. 2019. Nenhuma citação no texto.
- JAVA. *Onde obter informações técnicas sobre o Java*. Disponível em: <https://www.java.com/pt_BR/download/faq/techinfo.xml>. Acesso em: 16 nov. 2019. Nenhuma citação no texto.
- JUNIOR, E. F. Um estudo comparativo entre mysql e mongodb para armazenamento de dados provenientes de sensores. 2018. Citado na página 27.
- LECHETA, R. R. *Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK*. [S.l.]: Novatec, 2015. 1072 p. (5ª Edição). ISBN 9788575224687. Citado na página 21.
- LEITE, A. C.; MACEDO, H. Comparativo entre sistemas operacionais móveis—android x ios. 2017. Citado na página 20.
- MIRANDA, B. A. d. Usabilidade dos portais imobiliários: estudo comparativo entre imóvel web, viva real e zap imóveis. 2015. Citado 2 vezes nas páginas 29 e 30.
- NETBEANS. *Bem-Vindo ao NetBeans e ao site www.netbeans.org*. Disponível em: <https://netbeans.org/index_pt_PT.html>. Acesso em: 17 nov. 2019. Nenhuma citação no texto.
- NETO, J. M. d. S. Plataforma de e-commerce web c2c imobiliário. 2016. Citado na página 30.
- OLIVEIRA, M. de S. et al. Banco de dados no-sql x banco de dados sql. *South American Development Society Journal*, v. 4, n. 11, p. 298, 2018. Citado na página 26.
- ORACLE. *O que é um Banco de Dados*. Disponível em: <<https://www.oracle.com/br/database/what-is-database.html>>. Acesso em: 17 nov. 2019. Nenhuma citação no texto.
- ORLANDI, C. *Firestore: serviços, vantagens, quando utilizar e integrações*. 2018. Disponível em: <<https://blog.rocketseat.com.br/firebase/>>. Acesso em: 13 nov. 2019. Citado na página 21.
- TADDEI, A. A. Imóveis estudantes—aplicativo facebook. 2012. Citado na página 29.
- TANENBAUM, A. S.; STEEN, M. V. *Sistemas Distribuídos: Princípios e Paradigmas*. [S.l.]: Pearson Universidades, 2007. 416 p. (2ª Edição). ISBN 8576051427. Citado na página 24.
- ZURIARRAIN, J. M. Android já é o sistema operacional mais usado do mundo. *El País*, 2017. Citado na página 16.



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
- () Dissertação
- (**x**) Monografia
- () Artigo

Eu, **Helbert Alves Monteiro da Silva**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Um aplicativo para auxiliar estudantes a encontrar e dividir aluguel na cidade de Picos-PI** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 05 de fevereiro de 2020.

Helbert Alves Monteiro da Silva

Assinatura